National University of Singapore
School of Computing
CS3243 Introduction to AI

Tutorial 7: Logical Agents I

SOLUTIONS

Refer to **Appendix A** for notes on Knowledge Bases, and **Appendix B** for Propositional Logic Laws.

1. Verify the following logical equivalences. Cite the equivalence law used with each step of your working (refer to Appendix B for a list of these laws).

   (a) $\neg(p \lor \neg q) \lor (\neg p \land \neg q) \equiv \neg p$.

   (b) $(p \land \neg(\neg p \lor q)) \lor (p \land q) \equiv p$.

   Solution:

   (a)
   $$\begin{aligned} \neg(p \lor \neg q) \lor (\neg p \land \neg q) &\equiv (\neg p \land q) \lor (\neg p \land \neg q) && \text{(de Morgan's law)} \\ &\equiv \neg p \land (q \lor \neg q) && \text{(distributive law)} \\ &\equiv \neg p \land \textbf{true} && \text{(complement law)} \\ &\equiv \neg p && \text{(identity law)} \end{aligned}$$

   (b)
   $$\begin{aligned} (p \land \neg(\neg p \lor q)) \lor (p \land q) &\equiv (p \land (p \land \neg q)) \lor (p \land q) && \text{(de Morgan's law)} \\ &\equiv ((p \land p) \land \neg q)) \lor (p \land q) && \text{(associative law)} \\ &\equiv (p \land \neg q)) \lor (p \land q) && \text{(idempotent law)} \\ &\equiv p \land (\neg q \lor q) && \text{(distributive law)} \\ &\equiv p \land \textbf{true} && \text{(complement law)} \\ &\equiv p && \text{(identity law)} \end{aligned}$$

2. Victor would like to invite three friends, Alice, Ben, and Cindy to a party, but must satisfy the following constraints:

   C1. Cindy attends only if Alice does not attend.
   C2. Alice attends if either Ben or Cindy (or both) attend.
   C3. Cindy attends if Ben does not attend.

   Victor would like to know who will come to the party, and who will not. Help Victor by expressing each of the above three constraints in propositional logic, and then, using these constraints, determine who will attend his party.

   Solution: Let x denote Cindy attending, y denote Alice attending, and z denote Ben attending. Consequently, the constraints can be represented as follows.

   C1. $x \Rightarrow \neg y$
   C2. $(z \lor x) \Rightarrow y \equiv (z \Rightarrow y) \land (x \Rightarrow y)$. Hence, we have:
      i. $z \Rightarrow y$
      ii. $x \Rightarrow y$
   C3. $\neg z \Rightarrow x$

From C1 and C2ii, we have:

$$\begin{aligned}(x \Rightarrow \neg y) \land (x \Rightarrow y) &\equiv (\neg x \lor \neg y) \land (\neg x \lor y) && \text{(implication law)}\\ &\equiv \neg x \lor (y \land \neg y) && \text{(distributive law)}\\ &\equiv \neg x \lor \textbf{false} && \text{(negation law)}\\ &\equiv \neg x && \text{(identity law)}\end{aligned}$$

Since we know $\neg x$, and by the contrapositive of C3, we have $\neg x \Rightarrow z$, we therefore have $\neg x \land (\neg x \Rightarrow z)$, which implies z. Finally, from C2i, we also know $z \land (z \Rightarrow y)$, which implies y. Hence, we deduce that Alice and Ben will attend Victor's party. However, Cindy will not.

3. Consider the following knowledge base (KB).

R1. All Fire Trucks are red.
R2. All Fire Trucks are vehicles.
R3. All vehicles have four wheels.

(a) Assume that an inference algorithm, $\boldsymbol{A}_1$, that takes the query sentence "a Ferrari is a red vehicle", infers the statement "a Ferrari is a Fire Truck". Determine which of the following properties **does not** apply to $\boldsymbol{A}_1$.

Option 1: Complete.
Option 2: Sound.
Option 3: Both above.

(b) Assume that an inference algorithm, $\boldsymbol{A}_2$, is given the query sentence "a Ferrari is a red vehicle". Determine which of the following properties **would guarantee** that $\boldsymbol{A}_2$ would infer the sentence "a Ferrari has four wheels".

Option 1: Completeness.
Option 2: Soundness.
Option 3: Both above are required.

(c) Determine if the following statement is True or False. *Justify your answer*.

"Two agents with the same knowledge base but different inference engines, both of which are complete and sound, will always behave in the same way."

Solution:

(a) **Option 2**. As the inferred statement is not entailed by the KB, the algorithm cannot be sound. It may still be complete as the question does not state if the algorithm inferred anything else.

(b) **Option 1**. Completeness ensures that everything that is entailed by the knowledge base will be inferred. As the statement is entailed by the KB completeness guarantees its inference.

(c) **False**. The behaviour of an agent describes its interaction with the environment. Different complete and sound algorithms will (given the same KB) still result with the same inferences. A difference in how these inferences are derived that does not imply a difference in behaviour. However, the behaviour of an agent is not only dependent on its knowledge. Two different agents might have completely different objectives and hence act differently even with the same knowledge.

4. In each of the cases given below, a knowledge base (**KB**) and query (α) are specified. For each case, use **Truth Table Enumeration** to determine if **KB** $\models \alpha$. In other words, do the following.

   i. Write down all possible true/false assignments for all the variables present in associated **KB**.

   ii. For each such assignment, determine (1) if the **KB** is true/false, and (2) if α is true/false.

   iii. Finally, determine if $M(\mathbf{KB}) \subseteq M(\alpha)$, i.e., determine if the models of the KB are a subset of the model of α.

   (a) **KB**: $(x_1 \lor x_2) \land (x_1 \Rightarrow x_3) \land \neg x_2$
   α: $x_3 \lor x_2$

   (b) **KB**: $(x_1 \lor x_3) \land (x_1 \Rightarrow \neg x_2)$
   α: $\neg x_2$

Solution:

(a)

x_1	x_2	x_3	**KB**	α
True	*True*	*True*	*False*	*True*
True	*True*	*False*	*False*	*True*
True	*False*	*True*	*True*	*True*
True	*False*	*False*	*False*	*False*
False	*True*	*True*	*False*	*True*
False	*True*	*False*	*False*	*True*
False	*False*	*True*	*False*	*True*
False	*False*	*False*	*False*	*False*

Note that $\alpha = x_3 \lor x_2$ is *False* iff both x_2 and x_3 are *False* (i.e., rows 4 and 8 in the truth table – note the shaded rows). For these cases, note that the **KB** also evaluates to *False*, which shows that whenever α is *False*, the **KB** is also *False* (and thus that whenever the **KB** is *True*, α is also *True*). This proves $M(\mathbf{KB}) \subseteq M(\alpha)$, and therefore that **KB** $\models \alpha$.

(b)

x_1	x_2	x_3	**KB**	α
True	*True*	*True*	*False*	*False*
True	*True*	*False*	*False*	*False*
True	*False*	*True*	*True*	*True*
True	*False*	*False*	*True*	*True*
False	*True*	*True*	*True*	*False*
False	*True*	*False*	*False*	*False*
False	*False*	*True*	*True*	*True*
False	*False*	*False*	*False*	*True*

In order to show that **KB** $\models \alpha$, we need to show that whenever the **KB** is *True*, so too is α.

However, this is not the case. Note that in row 5 (shaded), we have a case where the **KB** is *True*, but where α is *False*.

Therefore **KB** $\not\models \alpha$.

Appendix A: Notes on Knowledge Bases

A knowledge base (KB) corresponds to a set of logical rules that models what the agent knows. These rules are written using a certain language (or *syntax*) and uses a certain truth model (or *semantics*) to determine when a certain statement is *True* or *False*. In propositional logic sentences are defined as follows.

1. Atomic Boolean variables are sentences.
2. If S is a sentence, then so is $\neg S$.
3. If S_1 and S_2 are sentences, then so are:
    a. $S_1 \wedge S_2$, i.e., "S_1 and S_2".
    b. $S_1 \vee S_2$, i.e., "S_1 or S_2".
    c. $S_1 \Rightarrow S_2$, i.e., "S_1 implies S_2".
    d. $S_1 \Leftrightarrow S_2$, i.e., "S_1 holds if and only if S_2 holds".

We say that a logical statement α models β ($\alpha \vDash \beta$) if β holds whenever α holds. In other words, if $M(\alpha)$ is the set of all value assignments to variables in α for which α holds *True*, then $M(\alpha) \subseteq M(\beta)$.

An inference algorithm ***A*** is one that takes as input a knowledge base (**KB**) and a query α and decides whether α is derived from **KB**, written as $\mathbf{KB} \vdash_{\boldsymbol{A}} \alpha$. ***A*** is *sound* if $\mathbf{KB} \vdash_{\boldsymbol{A}} \alpha$ implies that $\mathbf{KB} \vDash \alpha$; ***A*** is *complete* if $\mathbf{KB} \vDash \alpha$ implies that $\mathbf{KB} \vdash_{\boldsymbol{A}} \alpha$.

Appendix B: Propositional Logic Laws

De Morgan's Laws	$\neg(p \vee q) \equiv \neg p \wedge \neg q$	$\neg(p \wedge q) \equiv \neg p \vee \neg q$
Idempotent laws	$p \vee p \equiv p$	$p \wedge p \equiv p$
Associative laws	$(p \vee q) \vee r \equiv p \vee (q \vee r)$	$(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$
Commutative laws	$p \vee q \equiv q \vee p$	$p \wedge q \equiv q \wedge p$
Distributive laws	$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$	$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$
Identity laws	$p \vee False \equiv p$	$p \wedge True \equiv p$
Domination laws	$p \vee True \equiv True$	$p \wedge False \equiv False$
Double negation law	$\neg(\neg p) \equiv p$	
Complement laws	$p \vee \neg p \equiv True \vee \neg False \equiv True$	$p \wedge \neg p \equiv False \wedge \neg True \equiv False$
Absorption laws	$p \vee (p \wedge q) \equiv p$	$p \wedge (p \vee q) \equiv p$
Conditional identities	$p \Rightarrow q \equiv \neg p \vee q$	$p \Leftrightarrow q \equiv (p \Rightarrow q) \wedge (q \Rightarrow p)$