WU, Jialin (jw1816)

Department of Computing
Academic Year **2020-2021**
Page created Tue Nov 3 23:15:11 GMT 2020

70051 rac101 2 t5 jw1816 v1 0012000013843	Electronic submission **Sat - 31 Oct 2020 09:11:42** jw1816

Exercise Information

Module:	70051 Introduction to Symbolic Artificial Intelligence (MSc AI)	**Issued:**	Tue - 20 Oct 2020
Exercise:	2 (CW)	**Due:**	Tue - 03 Nov 2020
Title:	Logic	**Assessment:**	Individual
FAO:	Craven, Robert (rac101)	**Submission:**	Electronic

Student Declaration - Version 1

- I declare that this final submitted version is my unaided work.

Signed: (electronic signature) Date: 2020-10-29 07:58:14

For Markers only: (circle appropriate grade)

WU, Jialin (jw1816)	01197870	t5	2020-10-29 07:58:14	**A* A B C D E F**

Introduction to Symbolic AI – Coursework 1

WU, JIALIN CID:01197870

Question 1

(i) **If Michel isn't either fulfilled or rich, he won't live another five years.**
p = Michel is fulfilled
q = Michel is rich
r = Michel will live another five years

$(\neg(p \vee q)) \longrightarrow (\neg r)$

(ii) **Unless the snowstorm doesn't arrive, Raheen will wear his boots; but I'm sure it will arrive.**
p = Storm arrives
q = Raheen will wear his boots
r = I'm sure the snowstorm will arrive

$((\neg p) \vee q) \wedge r$

(iii) **If Akira and Toshire are on set, then filming will begin if and only if the caterers have cleared out.**
p = Akira is on set
q = Toshire is on set
r = Filming will begin
s = The caterers have cleared out

$(p \wedge q) \longrightarrow (r \leftrightarrow s)$

(iv) **Either Irad arrived, or Sarah didn't: but not both!**
p = Irad arrived
q = Sarah arrived

$(p \vee (\neg q)) \wedge (\neg(p \wedge q))$
My understanding to this sentence is that either Irad arrived or Sarah did not arrive but not the case that both Irad and Sarah arrived.

(v) **It's not the case both that Herbert heard the performance and Anne-Sophie did, if the latter didn't answer her phone calls.**
p = Herbert heard the performance
q = Anne-Sophie heard the performance
r = Anne-Sophie answered her phone calls

$(\neg(p \wedge q)) \vee r$

Question 2

(i) **Definition of satisfiability**
A propositional formula A is satisfiable if there is some v such that $h_v(A) = t$, where v is an atomic evaluation function and $h_v(A)$ is the propositional evaluation function of A based on v .

(ii) **Definition of logical equivalence**

Two propositional formulas A,B are logically equivalent if, for every v, $h_v(A) = h_v(B)$, where v is an atomic evaluation function and $h_v(A)$ and $h_v(B)$ are propositional evaluation functions based on v for A and B respectively.

(iii) Prove that a propositional formula $\neg A$ is satisfiable iff $\neg\neg A \not\equiv \top$.
Assume $\neg\neg A \not\equiv \top$, then not for every v that $h_v(\neg\neg A) = t$. This implies there is at least one v such that $h_v(\neg\neg A) = f$. According to Definition 1.5, $h_v(\neg\neg A) = f$ iff $h_v(\neg A) = t$. Hence, there is at least one v such that $h_v(\neg A) = t$ if $\neg\neg A \not\equiv \top$. Thus, $\neg A$ is satisfiable if $\neg\neg A \not\equiv \top$ is proved.

Now assume $\neg A$ is satisfiable. There is some v such that $h_v(\neg A) = t$ and, therefore, $h_v(\neg\neg A) = f$ according to Definition 1.5. This implies $h_v(\neg\neg A) = t$ is not valid for every v, i.e. $\neg\neg A \not\equiv \top$. Thus, $\neg\neg A \not\equiv \top$ if $\neg A$ is satisfiable is proved.

Since $\neg A$ is satisfiable if $\neg\neg A \not\equiv \top$ and $\neg\neg A \not\equiv \top$ if $\neg A$ is satisfiable, it is proved that $\neg A$ is satisfiable iff $\neg\neg A \not\equiv \top$.

Question 3

p	q	r	$(p$	$\wedge$	$\neg$	q	$\leftrightarrow$	$\neg$	$(\neg$	r	$\vee$	$\neg$	$p))$	$\rightarrow$	$(\neg$	$\neg$	p	$\rightarrow$	$r)$
t	t	t	t	f	f	t	f	t	f	t	f	f	t	t	t	f	t	f	t
t	t	f	t	f	f	t	t	f	t	f	t	f	t	f	t	f	t	f	f

Since we have already found an instance v such that the formulation is false, we can conclude that the formula is not valid and stop the truth table derivation. For example, when $v(p) = t, v(q) = t \text{ and } v(r) = f, h_v\Big((p \wedge \neg q \leftrightarrow (\neg r \vee \neg p)) \rightarrow (\neg\neg p \rightarrow r)\Big) = f$

Question 4

(i) **Conjunctive normal form (CNF):** a, b, f, g
Disjunctive normal form (DNF): b, e, h

(ii) Refutation-soundness and -completeness (Doets, 1994)
Let S be in CNF. $S \vdash_{res(pL)} \emptyset$ is defined as the refutation of S.

Soundness of refutation gives, if $S \vdash_{res(pL)} \emptyset$ then $S \vDash \emptyset$. Since $\emptyset$ is unsatisfiable, it follows that $S \vDash \perp$. That is:

$$\text{if } S \vdash_{res(pL)} \emptyset \text{ then } S \vDash \perp$$

Completeness of refutation defines that if S is unsatisfiable, then $\emptyset$ is result of resolution derivation of S. That is:

$$\text{If } S \vDash \perp \text{ then } S \vdash_{res(pL)} \emptyset$$

Together, it is proved logically that S $\vdash_{res(pL)} \emptyset$ iff $S \vDash \perp$. The refutation-soundness ensures that if a set of premises leads to $\perp$ after any number of resolution steps, the set must be unsatisfiable. Closeness of refutation then ensures that a conclusion can be drawn from propositional resolution. If there is no proof of a ⊥ in a finite set of premises in CNF, then the set is satisfiable. S is satisfiable iff $S \nvdash_{res(pL)} \emptyset$.

(iii) Unit propogation and pure rule

(a) {{p, s}, {q, r}, {¬s, q}, {¬p, ¬r, ¬s}}
⇒{{p, s},{¬p, ¬r, ¬s}} [q was pure]
⇒ {{p, s}} [¬r was pure]
⇒ {}
⇒ SATISFIABLE [since no condition for further application of rules apply]

(b) {{¬p, q, r}, {¬q}, {p, r, q}, {¬r, q}}
⇒ {{¬p, r}, {p, r}, {¬r}} [unit propogation by unit clause {¬q}]
⇒ {{¬p}, {p}}[unit propogation by unit clause {¬r}]
⇒ {{}}[unit propogation by unit clause {¬p}]
⇒ UNSATISFIABLE [since ∅ is in the set]

Question 5

If we define: p = I am going, q = You are going and r = Tara is going, then the sentences can be formalised as:

Premises

If I'm going, then you are going.	$p \longrightarrow \neg q$	≡	$\neg p \vee \neg q$
If you are not going, then neither is Tara	$\neg q \longrightarrow \neg r$	≡	$q \vee \neg r$
Either Tara's going or I'm not	$r \vee \neg p$	≡	$r \vee \neg p$
Tara's going unless I am.	$r \vee p$	≡	$r \vee p$

Conclusion

You are going q

In general, A1, . . . , An |= B iff A1 ∧ · · · ∧ An ∧ ¬B is unsatisfiable. So to prove the validity of argument, we can check the satisfiability of the following.

$$(\neg p \vee \neg q) \wedge (q \vee \neg r) \wedge (r \vee \neg p) \wedge (r \vee p) \wedge (\neg q)$$

Converting it to clausal form CNF: $\{\{\neg p, \neg q\}, \{q, \neg r\}, \{r, \neg p\}, \{r, p\}, \{\neg q\}\}$

Then applying DP:
⇒ $\{\{\neg r\}, \{r, \neg p\}, \{r, p\}\}$[unit propogation by unit clause $\{\neg q\}$]
⇒ $\{\{\neg p\}, \{p\}\}$[unit propogation by unit clause $\{\neg r\}$]
⇒ $\{\{\}\}$[unit propogation by unit clause $\{p\}$]
⇒ UNSATISFIABLE [since ∅ is in the set]

Since the CNF is unsatisfiable, the original argument is propositionally valid.

Question 6

(i) All of Andrea's aunts' aunts gave a cupcake to someone other than Andrea.

$\forall X(X = aunt(aunt(Andrea)) \rightarrow \exists Y(\neg(Y = Andrea) \wedge cupcake(X, Y)))$

$\mathcal{C} = \{andrea\}$

$\mathcal{F}_1 = \{aunt\}$ (aunt(X) means X's aunt)
$\mathcal{P}_2 = \{cupcake\}$ (cupcake(X,Y) means X gave a cupcake to y)
$\mathcal{L} = Tuple(\mathcal{C}, \mathcal{F}_1, \mathcal{P}_2)$

(ii) There's a computer connected to every computer which isn't connected to itself.

$\exists X \forall Y (computer(X) \land computer(Y) \land \neg connect(Y,Y) \land connect(X,Y))$
$\mathcal{P}_1 = \{computer\}$ (computer(X) means X is a computer)
$\mathcal{P}_2 = \{connect\}$ (connect(X,Y) means X is connected to Y
$\mathcal{L} = Tuple(\mathcal{P}_1, \mathcal{P}_2)$

(iii) Any painting by Paul Klee in a British gallery hangs in a room where all Kandinsky paintings in that gallery hang.

$$\forall G \exists Z (gallery(G) \land room(Z) \land in(Z,G) \longrightarrow \forall X \forall Y (X = painting(paul) \land Y = painting(kandinsky) \land in(X,Z) \land in(Y,Z)))$$

$\mathcal{F}_1\{painting\}$
painting(X) means X's painting
$\mathcal{P}_1\{gallery, room\}$
gallery(X) means X is a British gallery
room(X) mean X is a room
$\mathcal{P}_2\{in\}$
in(X,Y) X is in Y
$\mathcal{C} = \{paul, kandingsky\}$
$\mathcal{L} = Tuple(\mathcal{F}_1, \mathcal{P}_1, \mathcal{P}_2, \mathcal{C})$

(iv) If there's somebody who loves nobody, then it's false that everybody loves somebody

$$\exists X(\neg(\exists Y(love(X,Y))) \rightarrow \neg(\forall Z \exists P(love(Z,P)))$$

$\mathcal{P}_2\{love\}$ (love(X,Y) means X loves Y)
$\mathcal{L} = Tuple(\mathcal{P}_2)$

Question 7

Let $M = (D, \varphi)$ be the following $\mathcal{L}$-structure

(i) ∀X(a(k, X) → ¬(X = j))
False

- We need to show that $M \vDash \forall X(a(k,X) \rightarrow \neg(X = j))$, i.e. show that for all X $a(k,X) \rightarrow \neg(X = j)$ is true.
- So for any X, assume $(\varphi_M(k), \varphi_M(X)) \in \varphi_M(a)$, then we can show that X = $\varphi_M(j)$.
- Since X = $\varphi_M(j)$, it is false that $\neg(X = j)$.
- Hence, $M \nvDash \forall X\big(a(k,X) \rightarrow \neg(X = j)\big)$ and the original argument is false.

(ii) c(l) → ∃X(b(X) ∧ c(X) ∧ a(l, X))
True

- Let σ_1 be the M assignment such that X = j and $\sigma_1 =_X \sigma$.

- The antecedent $c(l)$ is true because $\varphi_M(l) \in \varphi_M(c)$.
- b(X) is true since constant j refers to a black object.
- c(X) is true since constant j also refers to a circular object.
- a(l, X) is true since $\varphi_{M,\sigma_1}(l), \varphi_{M,\sigma_1}(j) \in \varphi_M(A)$.
- Therefore, $M, \sigma_1 \vDash b(X) \wedge c(X) \wedge a(l, x)$.
- Hence ∃X(b(X) ∧ c(X) ∧ a(l, X)) is true.
- The original argument is therefore, true.

(iii) **∃X¬∃Y (¬(X = Y) ∧ a(X, Y))**

True

- Let σ_1 be the M assignment such that X = ■ and $\sigma_1 =_X \sigma$.
- We then need to prove that under σ_1, ¬∃Y (¬(X = Y) ∧ a(X, Y))
- Under M, σ_1, assume ∃Y (¬(X = Y) ∧ a(X, Y)) is true. Then there exists a Y such that (¬(X = Y) ∧ a(X, Y).
- Further assume that ¬(X = Y) is true, i.e.¬(Y = ■). All possible Ys are: $\varphi_{M,\sigma_1}(j)$, $\varphi_{M,\sigma_1}(k)$, $\varphi_{M,\sigma_1}(l)$ and □.
- For all these possible Ys, a(X, Y) is false.
- Thus ∃Y (¬(X = Y) ∧ a(X, Y)) is false and $\neg\exists Y(\neg(X = Y) \wedge a(X, Y))$ is true under M, σ_1.
- M,$\sigma_1 \vDash \neg\exists Y(\neg(X = Y) \wedge a(X, Y))$
- Hence, M⊨∃X¬∃Y (¬(X = Y) ∧ a(X, Y))

(iv) **∀X(¬s(X) → ∃Y (c(Y) ∧ b(Y) ∧ a(X, Y)))**

False

- For any X, assume ¬s(X) is true. Possible Xs are ■ and □.
- We must then show that ∃Y (c(Y) ∧ b(Y) ∧ a(X, Y)). There exist a Y such that c(Y) ∧ b(Y) ∧ a(X, Y)
- Further assume that c(Y) ∧ b(Y) is true. Possible Ys are $\varphi_M(j)$, $\varphi_M(k)$.
- For X = ■, a(X, $\varphi_M(j)$) and a(X, $\varphi_M(k)$) are both false.
- For X = ■, ∃Y (c(Y) ∧ b(Y) ∧ a(X, Y)) is therefore, false.
- For X = □, a(X, $\varphi_M(j)$) and a(X, $\varphi_M(k)$) are both false.
- For X = □, ∃Y (c(Y) ∧ b(Y) ∧ a(X, Y)) is therefore, false.
- Thus, for any X, if ¬s(X) then ∃Y (c(Y) ∧ b(Y) ∧ a(X, Y)) is not satisfiable.
- Hence, the original argument is false.

(v) **∀X(∃Y (¬(X = Y) ∧ a(X, Y)) → ∃Y (a(X, Y) ∧ a(Y, X)))**

False

- Let σ_1 be an X-variant of σ. σ_1 assigns $X = \varphi_M(k)$
- We then must show that M, σ_1 ⊨∃Y (¬(X = Y) ∧ a(X, Y)) → ∃Y (a(X, Y) ∧ a(Y, X))
- M, σ_1 ⊨∃Y (¬(X = Y) ∧ a(X, Y)) is true for $Y = \varphi_{M,\sigma_1}(j)$
- We then must show that M, σ_1 ⊨∃ Y (a(X, Y) ∧ a(Y, X)).
- Assume a(X, Y) is true, then $Y = \varphi_{M,\sigma_1}(j)$. a(Y, X) is therefore false.
- Thus M, σ_1 ⊭∃ Y (a(X, Y) ∧ a(Y, X)).
- Thus M, σ_1 ⊭∃Y (¬(X = Y) ∧ a(X, Y)) → ∃Y (a(X, Y) ∧ a(Y, X))
- The original argument is false.

(vi) **∀X∀Y (a(X, j) ∧ a(Y, j) → (a(X, Y) ∨ a(Y, X)))**

False

- Let σ_1 be the M assignment such that X = $\varphi_M(k)$ and Y = $\varphi_M(k)$.
- $M, \sigma_1 \vDash$ a(X, j) ∧ a(Y, j)
- $M, \sigma_1 \nvDash$ a(X, Y) and $M, \sigma_1 \nvDash$ a(Y, X). Thus $M, \sigma_1 \nvDash$ a(X, Y) ∨ a(Y, X)
- Therefore, $M, \sigma_1 \nvDash$ a(X, j) ∧ a(Y, j) → (a(X, Y) ∨ a(Y, X))
- Therefore, the original argument is false.

Reference

Doets, Kees. From Logic to Logic Programming. Cambridge: MIT, 1994. Foundations of Computing. Web.