FALEZAN, Thomas (tf317)

Imperial College London

Department of Computing

Academic Year **2020-2021**

Page created Tue Nov 3 23:15:05 GMT 2020

CATE

70051 rac101 2

t5 tf317 v1

0 0 1 2 0 0 0 1 5 2 2 2

Electronic submission

Tue - 03 Nov 2020 11:52:35

tf317

Exercise Information

Module: 70051 Introduction to Symbolic Artificial Intelligence (MSc AI)

Exercise: 2 (CW)

Title: Logic

FAO: Craven, Robert (rac101)

Issued: Tue - 20 Oct 2020

Due: Tue - 03 Nov 2020

Assessment: Individual

Submission: Electronic

Student Declaration - Version 1

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

Signed: (electronic signature) Date: 2020-11-03 10:52:38

For Markers only: (circle appropriate grade)

FALEZAN, Thomas (tf317)	01340793	t5	2020-11-03 10:52:38	**A* A B C D E F**

Symbolic AI - Coursework 1: Logic

1

(i) Define the following propositional atoms:

p: Michael is fulfilled

q: Michael is rich

r: He will live another 5 years

Then, we can formalize (i) in propositional logic by writing:

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

(ii) Define:

p: The snowstorm arrives

q: Raheem will wear his boots

r: I am sure the snowstorm will arrive

Then we can formalize (ii) in propositional logic by writing:

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

(iii) Define:

p: Akira is on set

q: Toshiro is on set

r: filming will begin

s: the caters have cleared out

Then we can formalize (iii) in propositional logic by writing:

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

(iv) Define:

p: Irad arrived

q: Sarah arrived

Then we can formalize (iv) in propositional logic by writing:

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

(v) Define:

p: Herbert heard the performance

q: Anne-Sophie heard the performance

r: Anne-Sophie answered her phone calls.

Then we can formalize (v) in propositional logic by writing:

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

2 (i) A propositional formula A is satisfiable if there exists an atomic evaluation function v s.t. the propositional evaluation function based on v, h_v, satisfies: $h_v(A) = t$

(ii) Let A and B be two propositional formulas. We say A and B are logically equivalent if for any atomic evaluation function v:
$h_v(A) = h_v(B)$. (h_v being the corresponding prop. evaluation function)

(iii) Let A be a propositional formula.

Claim: $\neg A$ is satisfiable iff $\neg\neg A \not\equiv \top$

Proof:

($\Rightarrow$) Suppose $\neg A$ is satisfiable. Then $\exists$ an atomic evaluation function v s.t. $h_v(\neg A) = t$.

But then $h_v(\neg\neg A) = f \neq t = h_v(\top)$

So $\neg\neg A$ and $\top$ are not logically equivalent, ie $\neg\neg A \not\equiv \top$

($\Leftarrow$) Suppose $\neg\neg A \not\equiv \top$.

Then $\exists$ an atomic evaluation function v s.t. $h_v(\neg\neg A) \neq h_v(\top) = t$

Hence $h_v(\neg\neg A) = f$.

So $h_v(\neg A) = t$

So $\neg A$ is satisfiable.

//

(3) We use a truth-table to show whether the following proposition is valid or not:

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

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

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

~~As for any atomic evaluation function, the formula is true, the formula must be valid.~~

Note that there exists an atomic evaluation function v s.t. the formula evaluates to false w.r.t. v so the formula is not valid.

④ (i)

a. $p \wedge (\neg q \vee r)$ is in CNF but not in DNF

clauses

b. $\neg p$ is both in CNF and DNF

c. $p \wedge \underbrace{(q \vee (p \wedge r))}_{\text{not a clause}}$ is neither in CNF nor in DNF

d. $\top$ is both in CNF and in DNF

e. $(p \wedge q) \vee (p \wedge q)$ is in DNF but not in CNF

f. $\underbrace{\neg\neg p}_{\text{not a clause}} \wedge (q \vee p)$ is neither in CNF nor in DNF

g. $p \wedge q$ is both in CNF and in DNF.

h. $p \vee q$ is both in CNF and in DNF.

(ii) Refutation-soundness and -completeness Theorem:

Let A be a logical formula in CNF.
Then, there exists a derivation by propositional resolution of the clause $\emptyset$ from A (ie $S \vdash_{Res(PL)} \emptyset$) iff $A \models \bot$

This theorem is important because if we can show that for a propositional formula A, $A \vdash_{Res(PL)} \emptyset$, then we have that A is unsatisfiable.

(iii) a. $\{\{p, s\}, \{q, r\}, \{\neg s, q\}, \{\neg p, \neg r, \neg s\}\}$

$\Rightarrow \{\{p, s\}, \{\neg p, \neg r, \neg s\}\}$ [applying the pure rule on q]

$\Rightarrow \{\{p, s\}\}$ [applying the pure rule on $\neg r$]

$\Rightarrow \{ \}$ [applying the pure rule on p]

b. $\{\{\neg p, q, r\}, \{\neg q\}, \{p, r, q\}, \{\neg r, q\}\}$

$\Rightarrow \{\{\neg p, r\}, \{p, r\}, \{\neg r\}\}$ [applying unit propagation on $\{\neg q\}$]

$\Rightarrow \{\{\neg p\}, \{p\}\}$ [applying unit propagation on $\{\neg r\}$]

$\Rightarrow \{\emptyset\}$ [applying unit propagation on $\{p\}$]

5. We formalize the argument in propositional logic:

Define:

p: I'm going

q: You're going

r: Tara is going

The premises are then: $\underbrace{p \to \neg q}_{A_1}, \underbrace{\neg q \to \neg r}_{A_2}, \underbrace{r \vee \neg p}_{A_3}, \underbrace{r \vee p}_{A_4}$

and the conclusion : $\underbrace{q}_{B}$

The argument is valid iff $(A_1 \wedge A_2 \wedge A_3 \wedge A_4) \wedge \neg B$ is unsatisfiable.

Hence, we convert the above formula in CNF and apply DP algorithm to check for satisfiability.

Remembering that $p \to q \equiv q \vee \neg p$, the formula becomes:

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

which we represent as the set of sets:

$\{\{\neg q, \neg p\}, \{\neg r, q\}, \{\neg p, r\}, \{r, p\}, \{\neg q\}\}$

Applying DP algorithm, we get:

$\Rightarrow \{\{\neg r, q\}, \{r\}\}$ [applying unit propagation on $\{\neg\ \}$]

$\Rightarrow \{\{q\}\}$ [applying unit propagation on $\{r\}$]

$\Rightarrow \{ \}$ [applying unit propagation on $\{q\}$]

~~Hence, the formula in CNF is satisfiable and thus, the argument is not valid.~~

Applying DP algorithm, we get:

$\Rightarrow \{ \{\neg r\}, \{\neg p, r\}, \{r, p\} \}$ [applying unit propagation on $\{\neg q\}$]

$\Rightarrow \{ \{\neg p\}, \{p\} \}$ [applying unit propagation on $\{\neg r\}$]

$\Rightarrow \{ \emptyset \}$ [applying unit propagation on $\{p\}$]

So the above formula is unsatisfiable and hence, the argument is valid.

⑥ (i) Let $\mathcal{L}$ be the First Order Logic signature consisting of the following sets:

$C = \{ \text{Andrea} \}$

$P_1 = \{ \text{cupcake} \}$, $P_2 = \{ \text{aunt} \}$, $P_3 = \{ \text{gave} \}$, $P_i = \emptyset$ for $i \geq 4$

$F_i = \emptyset$ for $i \geq 1$

where: cupcake(X) means 'X is a cupcake'

aunt(X, Y) means 'X is an aunt of Y'

gave(X, Y, Z) means 'X gave Z to Y'

The sentence then translates to:

$$\forall X \ (\exists Y \ (\text{aunt}(Y, \text{Andrea}) \wedge \text{aunt}(X, Y)) \rightarrow \exists \Theta \ \exists Z \ (\text{gave}(X, Z, \Theta) \wedge \text{cupcake}(\Theta) \wedge Z \neq \text{Andrea}))$$

$\neg(Z = \text{Andrea})$

(ii) Let $\mathcal{L}$ be the First Order Logic signature consisting of the following sets:

$C = \emptyset$

$\mathcal{P}_1 = \{\text{computer}\}$, $\mathcal{P}_2 = \{\text{connected}\}$, $\mathcal{P}_i = \emptyset$ for $i \geq 3$

$\mathcal{F}_i = \emptyset$ for $i \geq 1$

where: computer(X) means 'X is a computer'
connected(X,Y) means 'X is connected to Y'

The sentence then translates to:

~~$\exists X\,(\forall Y(\neg\text{connected}(Y,Y) \rightarrow \text{connected}(X,Y))$~~

$\exists X\,(\text{computer}(X) \wedge \forall Y\,(\text{computer}(Y) \wedge \neg\text{connected}(Y,Y) \rightarrow \text{connected}(X,Y)))$

(iii) Let $\mathcal{L}$ be the First Order Logic signature consisting of the following sets:

$C = \{\text{Paul Klee}, \text{Kandinsky}\}$

$\mathcal{P}_1 = \{\text{British\_gallery}, \text{room}\}$ $\mathcal{P}_2 = \{\text{painting}, \text{hangs}, \text{in}\}$ $\mathcal{P}_i = \emptyset$ for $i \geq 3$

$\mathcal{F}_i = \emptyset$ for $i \geq 1$.

where: British_gallery(X) means 'X is a British gallery'
room(X) means 'X is a room'
painting(X,Y) means 'X is a painting by Y'
hangs(X,Y) means 'X hangs in Y'
in(X,Y) means 'X is in Y'

According to $\mathcal{L}$, the sentence then translates to:

$$\forall X\,\forall Y\,(\text{painting}(X, \text{Paul Klee}) \wedge \text{in}(X,Y) \wedge \text{British\_gallery}(Y)$$
$$\rightarrow \exists Z\,(\text{room}(Z) \wedge \text{in}(Z,Y) \wedge \text{hangs}(X,Z)$$
$$\wedge\ \forall P\,(\text{painting}(P, \text{Kandinsky}) \wedge \text{in}(P,Y) \rightarrow \text{hangs}(P,Z))))$$

(iv) Let $\mathcal{L}$ be the First Order Logic signature consisting of the following sets:

$C = \emptyset$

$P_2 = \{ love \}$

where: $love(X,Y)$ means 'X loves Y'

According to $\mathcal{L}$, the sentence then translates to:

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

(7) Let M be the $\mathcal{L}$-structure (D, φ)

(i) $\forall X (a(k,X) \rightarrow \neg(X=j))$

Claim: $M \not\models \forall X (a(k,X) \rightarrow \neg(X=j))$

Proof:

Let σ be any variable assignment.

Consider σ', an X-variant of σ s.t. $\sigma'(X) = \varphi(j)$.

Then $M, \sigma' \models a(k,X)$ as there is a directed arrow from $\varphi(k)$ to $\varphi(j)$

but $M, \sigma' \not\models \neg(X=j)$ as $\sigma'(X) = \varphi(j)$

So it is not the case that for any X-variant σ' of σ we have:

$M, \sigma' \models a(k,X) \rightarrow X \neq j$

So $M \not\models \forall X (a(k,X) \rightarrow X \neq j)$ //

(ii) Claim: $M \models c(l) \rightarrow \exists X (b(X) \wedge c(X) \wedge a(l,X))$

Proof:

Note that $M \models c(l)$ so to show the claim, it is enough to show

$M \models \exists X (b(X) \wedge c(X) \wedge a(l,X))$

Let σ be any variable assignment and consider σ' an X-variant of σ,

s.t. $\sigma'(X) = \varphi(k)$

Then $M, \sigma' \models b(X) \wedge c(X) \wedge a(l,X)$

So $M, \sigma \models \exists X (b(X) \wedge c(X) \wedge a(l,X))$

so $M \models c(\ell) \to \exists X(b(X) \wedge c(X) \wedge a(\ell, X))$

//

(iii) Claim: $M \models \exists X \neg \exists Y(X \neq Y \wedge a(X,Y))$

Proof:

Let σ be any variable assignment and consider σ', an X-variant of σ, s.t.
~~$\sigma'(X) = \varphi(b) \wedge \varphi(s)$~~ $\sigma'(X)$ equals the black square object.
Then, it is clear from the diagram that there is no object other than $\sigma'(X)$ s.t. there is a directed arrow from that object to $\sigma'(X)$.
That is, there is no Y-variant, σ'', of σ' s.t. $\sigma''(Y) \neq \sigma''(X)$ and
$M, \sigma'' \models a(X,Y)$.

So $M, \sigma' \models \neg \exists Y(X \neq Y \wedge a(X,Y))$

So $M \models \exists X \neg \exists Y(X \neq Y \wedge a(X,Y))$

//

(iv) Claim: $M \not\models \forall X(\neg s(X) \to \exists Y(c(Y) \wedge b(Y) \wedge a(X,Y)))$

Proof:

Let σ be any variable assignment and consider σ', the X-variant of σ s.t.
$\sigma'(X) = \varphi(j)$.

We claim $M, \sigma' \not\models \neg s(X) \to \exists Y(c(Y) \wedge b(Y) \wedge a(X,Y))$

First, note $M, \sigma' \models \neg s(X)$ as the object reffered by j is not a square

~~Note also that the only object from which an arrow points to $\varphi(\ell)$ is $\varphi(\ell)$ and $\varphi(\ell)$ is not black.~~

~~Hence $M, \sigma' \not\models \exists Y(c(Y) \wedge b(Y) \wedge a(X,Y))$~~

~~So $M, \sigma' \not\models \neg s(X) \to \exists Y(c(Y) \wedge b(Y)$~~

Note also that the only object o s.t. an arrow points from $\varphi(j)$ to o is $\varphi(\ell)$ and that $\varphi(\ell)$ is not black.

That is, there doesn't exist a Y-variant, σ'', of σ' s.t. $M, \sigma'' \models c(Y) \wedge b(Y) \wedge a(X,Y)$

so $M, \sigma' \not\models \exists Y(c(Y) \wedge b(Y) \wedge a(X,Y))$

So $M, \sigma' \not\models \neg s(X) \to \exists Y(c(Y) \wedge b(Y) \wedge a(X,Y))$

So $M \not\models \forall X(\neg s(X) \to \exists Y(c(Y) \wedge b(Y) \wedge a(X,Y)))$

//

(v) Claim: $M \nvDash \forall X(\exists Y(X \neq Y \wedge a(X,Y)) \rightarrow \exists Y(a(X,Y) \wedge a(Y,X)))$

Proof:

Let σ be any variable assignment and consider σ', the X-variant of σ s.t. $\sigma'(X) = \varphi(k)$.

We claim $M, \sigma' \nvDash \exists Y(X \neq Y \wedge a(X,Y)) \rightarrow \exists Y(a(X,Y) \wedge a(Y,X))$

Note that $\varphi(j) \neq \varphi(k)$ and that $M, \sigma' \vDash a(k,j)$ so:

$M, \sigma' \vDash \exists Y(X \neq Y \wedge a(X,Y))$

Note also that there is no ~~[illegible]~~ object o s.t. there is both an arrow from o to $\varphi(k)$ and an arrow from $\varphi(k)$ to o.

That is: $M, \sigma' \nvDash \exists Y(a(X,Y) \wedge a(Y,X))$

So $M, \sigma' \nvDash \exists Y(X \neq Y \wedge a(X,Y)) \rightarrow \exists Y(a(X,Y) \wedge a(Y,X))$

So $M \nvDash \forall X(\exists Y(X \neq Y \wedge a(X,Y)) \rightarrow \exists Y(a(X,Y) \wedge a(Y,X)))$

//

(vi) Claim: $M \nvDash \forall X \forall Y(a(X,j) \wedge a(Y,j) \rightarrow a(X,Y) \vee a(Y,X))$

Proof:

Let σ be any variable assignment and consider σ', the X-variant of σ s.t. $\sigma'(X) = \varphi(k)$.

We claim $M, \sigma' \nvDash \forall Y((a(X,j) \wedge a(Y,j)) \rightarrow a(X,Y) \vee a(Y,X))$

Consider $\sigma'' = \sigma'$, a Y-variant of σ'.

Then $\sigma''(X) = \sigma''(Y) = \varphi(k)$ and we see ~~tha~~ from the diagram that $M, \sigma'' \vDash a(X,j) \wedge a(Y,j)$.

However, it is also clear $M, \sigma'' \nvDash a(X,Y) \vee a(Y,X)$ as there is no arrow from $\varphi(k)$ to itself.

Hence $M, \sigma' \nvDash \forall Y(a(X,j) \wedge a(Y,j) \rightarrow a(X,Y) \vee a(Y,X))$

So $M \nvDash \forall X \forall Y(a(X,j) \wedge a(Y,j) \rightarrow a(X,Y) \vee a(Y,X))$

//