DAXBACHER, Patricia (pd720)

Imperial College London

Department of Computing
Academic Year 2020-2021
Page created Tue Nov 3 23:15:07 GMT 2020

CATE

70051 rac101 2 t5 pd720 v1 0012000014924	Electronic submission **Mon - 02 Nov 2020 16:14:36** pd720

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: 2020-10-31 20:21:31

For Markers only: (circle appropriate grade)

DAXBACHER, Patricia (pd720)	01975493	t5	2020-10-31 20:21:31	**A* A B C D E F**

Introduction to Symbolic AI
Coursework 1: Logic

Patricia Daxbacher

November 2, 2020

1. i. If Michel isn't either fulfilled or rich, he won't live another five years.
   $((\neg(p \vee q)) \to (\neg r))$
   p: Michel is fulfilled.
   q: Michel is rich.
   r: Michel will live another five years.

   ii. Unless the snowstorm doesn't arrive, Raheem will wear his boots; but I'm sure it *will* arrive.
   $(((\neg p) \vee q) \wedge r)$
   p: The snowstorm arrives.
   q: Raheem will wear his boots.
   r: I am sure the snowstorm will arrive.

   iii. If Akira and Toshiro are on set, then filming will begin if and only if the caterers have cleared out.
   $((p \wedge q) \to (r \leftrightarrow s))$
   p: Akira is on set.
   q: Toshiro is on set.
   r: Filming will begin.
   s: The caterers have cleared out.

   iv. Either Irad arrived, or Sarah didn't: but not both!
   $((p \vee (\neg q)) \wedge (\neg(p \wedge (\neg q))))$
   p: Irad arrived.
   q: 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.
   $((\neg q) \to (\neg(p \wedge q)))$
   p: Herbert heard the performance.
   q: Anne-Sophie heard the performance.
   r: Anne-Sophie answered her phone calls.

2. i. A formula A of propositional logic is *satisfiable* if there exists an atomic evaluation v such that $h_v(A) = \mathbf{t}$.

   ii. Two formulas A and B of propositional logic are *logically equivalent*, noted as $A \equiv B$, if for all atomic evaluations v it holds that $h_v(A) = h_v(B)$.

   iii. Prove that $\neg A$ is satisfiable if and only if $\neg\neg A \not\equiv \top$.

   Proof. Let $\neg A$ be a satisfiable formula. Then there is some atomic evaluation v such that $h_v(\neg A) = \mathbf{t}$. By the propositional evaluation of $\neg$ it follows that $h_v(A) = \mathbf{f}$ and

thus $h_v(\neg\neg A) = \mathbf{f}$. Therefore, $\neg\neg A$ is not logically equivalent to $\top$ as $h_v(\neg\neg A) = \mathbf{f} \neq \mathbf{t} = h_v(\top)$.
Let $\neg\neg A \not\equiv \top$. It holds that $\neg\neg A \equiv A$ by the propositional evaluation of $\neg$. Thus, there is some evaluation v such that $h_v(A) = \mathbf{f}$ and $h_v(\neg A) = \mathbf{t}$. It follows by definition that $\neg A$ is satisfiable. ■

3. We use truth-tables to determine whether the following is valid or not:
$(p \wedge \neg q \leftrightarrow \neg(\neg r \vee \neg p)) \to (\neg\neg q \to r)$.

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

The given formula is not valid as there are valuations such that the formula is evaluated to false (marked in red in the truth table).

4. i. a. $p \wedge (\neg q \vee r)$
The formula is in CNF. The formula is not in DNF.
b. $\neg p$
The formula is in CNF and in DNF.
c. $p \wedge (q \vee (ß \wedge r))$
The formula is neither in CNF nor in DNF.
d. $\top$
The formula is in CNF and in DNF.
e. $(p \wedge q) \vee (p \wedge q)$
The formula is in DNF. The formula is not in CNF.
f. $\neg\neg p \wedge (q \vee p)$
The formula is neither in CNF nor in DNF.
g. $p \wedge q$
The formula is in CNF and in DNF.
h. $p \vee q$
The formula is in CNF and in DNF.

ii. Let S be a formula in CNF. It holds that $S \models \bot$ if and only if there is some derivation of the empty clause from S, i.e., $S \vdash_{\text{res(PL)}} \emptyset$. Where $S \models \bot$ means nothing else than S is unsatisfiable. This property can be applied to derive algorithms for SAT solvers. Applying it naively to check SAT, we build all resolution-derivations from S. But also more efficient algorithms like Davis-Putnam or Davis-Logemann-Loveland have the property of the refutation-soundness and -completeness of a resolution derivation at its heart.

iii. a. $\{\{p, s\}, \{q, r\}, \{\neg s, q\}, \{\neg p, \neg r, \neg s\}\}$
$\Rightarrow \{\{p, s\}, \{\neg p, \neg r, \neg s\}\}$ [q was pure]

$\Rightarrow \{\{p, s\}\}$ [$\neg r$ was pure]

b. $\{\{\neg p, q, r\}, \{\neg q\}, \{p, r, q\}, \{\neg r, q\}\}$
$\Rightarrow \{\{\neg p, r\}, \{p, r\}, \{\neg r\}\}$ [unit propagation by unit clause $\{\neg q\}$]
$\Rightarrow \{\{\neg p\}, \{p\}\}$ [unit propagation by unit clause $\{\neg r\}$]
$\Rightarrow \{\{\}\}$ [unit propagation by unit clause $\{p\}$]

5. If I'm going, then you aren't.
If you're not going, then neither is Tara.
Either Tara's going or I'm not.
Tara's going unless I am.
So, you're going.
I formalize it as: $p \to \neg q, \neg q \to \neg r, r \vee \neg p, r \vee p$, so q.
p: I am going.
q: You are going.
r: Tara is going.
To check the validity of the argument in propositional logic, we have to show that

$$p \to \neg q, \neg q \to \neg r, r \vee \neg p, r \vee p \models q.$$

Since $A_1, A_2, \ldots, A_n \models B$ iff $A_1 \wedge A_2 \wedge \ldots \wedge A_n \wedge \neg B$ is unsatisfiable, we can check the satisfiability of $(p \to \neg q) \wedge (\neg q \to \neg r) \wedge (r \vee \neg p) \wedge (r \vee p) \wedge \neg q$ to determine the validity of the aforementioned argument.
First, we convert it to clausal-form CNF: $\{\{\neg p, \neg q\}, \{q, \neg r\}, \{r, \neg p\}, \{r, p\}, \{\neg q\}\}$. Now we can use Davis Putnam to determine the satisfiability:
$\{\{\neg p, \neg q\}, \{q, \neg r\}, \{r, \neg p\}, \{r, p\}, \{\neg q\}\}$
$\Rightarrow \{\{\neg r\}, \{r, \neg p\}, \{r, p\}\}$ [unit propagation by unit clause $\{\neg q\}$]
$\Rightarrow \{\{\neg p\}, \{p\}\}$ [unit propagation by unit clause $\{\neg r\}$]
$\Rightarrow \{\{\}\}$ [unit propagation by unit clause $\{p\}$]
$\Rightarrow$ UNSATISFIABLE [since $\emptyset$ is in the set]
Thus, the argument is valid.

6. I consider all sets in the signature of FOL, which are not specified, to be empty throughout the exercise.

   i. All of Andreas's aunts' aunts gave a cupcake to someone other than Andrea.
   Consider the following FOL signature $\mathcal{L}$ with:
   $\mathcal{C} = \{Andrea\}$
   $\mathcal{P}_1 = \{cupcake\}$
   $\mathcal{P}_2 = \{aunt\}$
   $\mathcal{P}_3 = \{gave\}$
   $\forall X \forall Y (aunt(X, Y) \wedge aunt(Y, Andrea) \to \exists Z \exists W (gave(X, Z, W) \wedge cupcake(Z) \wedge \neg(W = Andrea)))$
   Where the predicates have the meaning:
   $cupcake(X)$ ('X is a cupcake')
   $aunt(X, Y)$ ('X is an aunt of Y')
   $gave(X, Y, Z)$ ('X gave Y to Z')

   ii. There's a computer connected to every computer which isn't connected to itself.
   Consider the following FOL signature $\mathcal{L}$ with:
   $\mathcal{P}_1 = \{computer\}$

$\mathcal{P}_2 = \{connected\}$
$\exists X(computer(X) \wedge \forall Y(computer(Y) \wedge \neg connected(Y,Y) \rightarrow connected(X,Y)))$
Where the predicates have the meaning:
$computer(X)$ ('X is a computer')
$connected(X,Y)$ ('X is connected to Y')

iii. Any painting by Paul Klee in a British gallery hangs in a room where all Kandinsky paintings in that gallery hang.
Consider the following FOL signature $\mathcal{L}$ with:
$\mathcal{C} = \{paulKlee, kandinsky\}$
$\mathcal{P}_1 = \{painting, room, gallery, british\}$
$\mathcal{P}_2 = \{hangIn, locatedIn, painter\}$
$\forall X \forall Y(painting(X) \wedge painter(paulKlee, X) \wedge locatedIn(X,Y) \wedge british(Y) \wedge gallery(Y)$
$\rightarrow \exists R(room(R) \wedge locatedIn(Y,R) \wedge \forall Z(painting(Z) \wedge painter(kandinsky, Z) \wedge$
$locatedIn(Z,Y) \rightarrow hangIn(Z,R)) \wedge hangIn(X,R)))$
Where the predicates have the meaning:
$painting(X)$ ('X is a painting')
$room(X)$ ('X is a room')
$gallery(X)$ ('X is a gallery')
$british(X)$ ('X is British')
$hangIn(X,Y)$ ('X hangs in Y')
$locatedIn(X,Y)$ ('X is (located) in Y')
$painter(X,Y)$ ('X is painter of Y')

iv. If there's somebody who loves nobody, then it's false that everybody loves somebody.
$\exists X \forall Y \neg loves(X,Y) \rightarrow \neg(\forall X \exists Y loves(X,Y))$
Where the predicate has the meaning:
$loves(X,Y)$ ('X loves Y')

7. All the given formulas are sentences. Thus, by Corollary 3.12, it follows that $M, \sigma \models S$ does not depend on σ which means we can prove the following for an arbitrary M-assignment σ.

i. $\forall X(a(k,X) \rightarrow \neg(X = j))$
Let σ be an arbitrary M-assignment. Then $M, \sigma \models \forall X(a(k,X) \rightarrow \neg(X = j))$ iff for all X-variant assignments σ^* of σ it holds true that $M, \sigma^* \models a(k,X) \rightarrow \neg(X = j)$. Consider the X-variant σ^* with $\varphi_{\sigma^*}(X) = \varphi(j)$. Then it is true that $M, \sigma^* \models a(k,X)$ as $(\varphi_{\sigma^*}(k), \varphi_{\sigma^*}(X)) \in \varphi(a)$. But clearly $M, \sigma^* \not\models \neg(X = j)$ and by the evaluation of '$\rightarrow$' it follows that $M, \sigma \not\models \forall X(a(k,X) \rightarrow \neg(X = j))$.

ii. $c(l) \rightarrow \exists X(b(X) \wedge c(X) \wedge a(l,X))$
Let σ be an arbitrary M-assignment. Then $M, \sigma \models c(l)$ iff $\varphi_\sigma(l) \in \varphi(c)$ which is true. Therefore, by the evaluation of '$\rightarrow$' we must show that $M, \sigma \models \exists X(b(X) \wedge c(X) \wedge a(l,X))$. This is true iff for some X-variant assignment σ^* it holds that $M, \sigma^* \models b(X) \wedge c(X) \wedge a(l,X)$. Consider $\varphi_{\sigma^*}(X) = \varphi(j)$. Then $\varphi_{\sigma^*}(X) \in \varphi(b) \cap \varphi(c)$ and $(\varphi_{\sigma^*}(l), \varphi_{\sigma^*}(X)) \in \varphi(a)$.
So, $M \models c(l) \rightarrow \exists X(b(X) \wedge c(X) \wedge a(l,X))$.

iii. $\exists X \neg \exists Y(\neg(X = Y) \wedge a(X,Y))$
Let σ be an arbitrary M-assignment. Then $M, \sigma \models \exists X \neg \exists Y(\neg(X = Y) \wedge a(X,Y))$ iff for some X-variant σ^* it holds true that $M, \sigma^* \models \neg \exists Y(\neg(X = Y) \wedge a(X,Y))$. Consider $\varphi_{\sigma^*}(X) = \blacksquare$. We want to show that for this σ^* the aforementioned is true in M. $M, \sigma^* \models \neg \exists Y(\neg(X = Y) \wedge a(X,Y))$ holds true iff $M, \sigma^* \not\models \exists Y(\neg(X = Y) \wedge a(X,Y))$. It holds that $\varphi(a)(\varphi_{\sigma^*}(X), \cdot) = \{(\blacksquare, \blacksquare)\}$. So, $\neg(X = Y) \wedge a(X,Y)$ is false in M for

$\varphi_{\sigma^*}(X) = \blacksquare$ and every assignment of Y. Therefore, $M, \sigma^* \not\models \exists Y(\neg(X = Y) \wedge a(X, Y))$ is true.
To conclude: $M \models \exists X \neg \exists Y(\neg(X = Y) \wedge a(X, Y))$.

iv. $\forall X(\neg s(X) \rightarrow \exists Y(c(Y) \wedge b(Y) \wedge a(X, Y)))$
Let σ be an arbitrary M-assignment. We have to show that for every X-variant σ^* of σ it holds that $M, \sigma^* \models \neg s(X) \rightarrow \exists Y(c(Y) \wedge b(Y) \wedge a(X, Y))$. $\neg s(X)$ is true iff $\varphi_{\sigma^*}(X) \in D \setminus \varphi(s) = \{\varphi(j), \varphi(k), \varphi(l)\}$. Consider σ^* such that $\varphi_{\sigma^*}(X) = \varphi(j)$. So, we need to check if the RHS evaluates to true for this assignment. Consider a Y-variant σ^{**} of σ^*, so we have that $\varphi_{\sigma^{**}}(X) = \varphi_{\sigma^*}(X)$. But since $\varphi(c) \cap \varphi(b) = \{\varphi(j), \varphi(k)\}$ and $(\varphi(j), \varphi(k)) \notin \varphi(a)$, $(\varphi(j), \varphi(j)) \notin \varphi(a)$ it follows that $M, \sigma^{**} \not\models c(Y) \wedge b(Y) \wedge a(X, Y)$ and therefore $M \not\models \forall X(\neg s(X) \rightarrow \exists Y(c(Y) \wedge b(Y) \wedge a(X, Y)))$.

v. $\forall X(\exists Y(\neg(X = Y) \wedge a(X, Y)) \rightarrow \exists Y(a(X, Y) \wedge a(Y, X)))$
Let σ be an arbitrary M-assignment and σ^* an X-variant of σ such that $\varphi_{\sigma^*}(X) = \varphi(k)$. The LHS of the conditional is true iff there is some Y-variant σ^{**} such that $M, \sigma^{**} \models a(X, Y)$ and $M, \sigma^{**} \models \neg(X = Y)$. $\neg(X = Y) \wedge a(X, Y)$ holds true in M for the following object pairs: $\{(\varphi(k), \varphi(j)), (\varphi(j), \varphi(l)), (\varphi(l), \varphi(j)), (\varphi(l), \varphi(k)), (\varphi(l), \square)\}$. Thus, for the specific assignment σ^* the LHS evaluates to true. To evaluate the RHS of the conditional to true we must now find some Y-variant σ^{***} of σ^* such that $M, \sigma^{***} \models a(X, Y) \wedge a(Y, X)$. We have that $\varphi_{\sigma^{***}}(X) = \varphi_{\sigma^*}(X) = \varphi(k)$. But there is no object $\mathbf{obj}$ such that $(\mathbf{obj}, \varphi(k)) \in \varphi(a)$ and $(\varphi(k), \mathbf{obj}) \in \varphi(a)$.
Therefore, $M \not\models \forall X(\exists Y(\neg(X = Y) \wedge a(X, Y)) \rightarrow \exists Y(a(X, Y) \wedge a(Y, X)))$

vi. $\forall X \forall Y(a(X, j) \wedge a(Y, j) \rightarrow (a(X, Y) \vee a(Y, X)))$
Let σ be an arbitrary M-assignment and σ^* an X-variant of σ and σ^{**} a Y-variant of σ^*. For every combination of such assignments it must hold that $M, \sigma^{**} \models a(X, j) \wedge a(Y, j) \rightarrow (a(X, Y) \vee a(Y, X))$. Consider $\sigma^{**}(X) = \sigma^{**}(Y) = \varphi(k)$. Then the LHS of the conditional evaluates to true as $(\varphi(k), \varphi(j)) \in \varphi(a)$ but the RHS evaluates to false as $(\varphi(k), \varphi(k)) \notin \varphi(a)$ which evaluates the conditional to false.
Therefore, $M \not\models \forall X \forall Y(a(X, j) \wedge a(Y, j) \rightarrow (a(X, Y) \vee a(Y, X)))$.