CS 270 - First Order Logic Homework

M. Boady, B. Char, G. Long, S. Earth, and J. Johnson

This homework is worth 100 points.

Any proofs on this assignment should be constructed using **Natural Deduction Proof Editor** `http://proofs.openlogicproject.org`. Counterexamples can simply be typed out, but make sure that all proper symbols are used (LogicWriter `https://www.cs.drexel.edu/~charbw/logicwriter/standardConfigEarth/web/index.html` could be helpful here)

To submit the assignment

1. Create a word document *drexeluserid*_`hw3.doc`.
2. For each question, you must state whether the presented argument is valid or whether it is invalid.
3. If the argument is invalid, then you must provide a domain model for the objects and an interpretation for the predicates that serves as a counter example. For instance, consider the argument
   $\forall x \in S(Rx \implies Wx), \exists x \in S(Wx \wedge Px) \therefore \exists x \in S(Rx \wedge Px)$
   Then for a counterexample you could let S = {9, 12}, R=odd, W=multiple of 3, P=greater than ten. Now we must verify that all the premises are true. The first premise translates to all odd numbers being multiples of 3 (True for domain, since the only odd in S is "9" which is a multiple of 3). The second premise is also true since our domain contains a multiple of 3 greater than ten ("12"). Lastly, we see that the claimed conclusion (that there is an odd number greater than ten) is actually False (the only odd in S is 9, which is less than 10)
4. If the argument is valid, then you must complete a proof using the website and take a screenshot of the proof. You may only use any Rule of Deduction supported by the proof checker. You are no longer limited to the basic rules.
5. Every rule is now permitted! Copy-Paste the screenshot of any proof into the Word Document. Make sure your screenshot includes the part that says either **Congratulations! This proof is correct** or **Sorry there were errors**. If you cannot get your answer to work, partial credit will be given. You **must** include all errors generated by the checker to earn partial credit.
6. Export the Word document as a *single* PDF and submit *drexeluserid*_`hw3.pdf`. to GradeScope. Each page of the pdf should be a separate question.

For this homework, remember to set the Proof Checker to FOL mode.

This Homework uses one new rule you have not seen in class. If two values are equal, then we can replace one with the other. This is only true in FOL. Here is an example.

1.	$Fa \implies (a = b)$	Premise
2.	Fa	Premise
3.	Qb	Premise
4.	$a = b$	$\implies$E 1,2
5.	Qa	=E 3,4

The rule $=E3,4$ is justified by two numbers. Line 3 is the expression Qb. Line 4 says $a = b$. That means we can replace b with a and get the same result. You can only replace one variable at a time. If you know $a = b$ you can replace a with b or b with a, but not both at the same time. Each rule application can only use the equals in one direction, for example making b in a.

Construct a proof for the argument: ∀x(Lx → ¬Ox), ∃x(Lx ∧ Jx) ∴ ∃x(Jx ∧ ¬Ox)

1	∀x(Lx → ¬Ox)	
2	∃x(Lx ∧ Jx)	
3	Lc ∧ Jc	
4	Jc	∧E 3
5	∃xJx	∃I 4
6	Lc → ¬Oc	∀E 1
7	Lc	∧E 3
8	¬Oc	→E 6, 7
9	Jc ∧ ¬Oc	∧I 4, 8
10	∃x (Jx ∧ ¬Ox)	∃I 9
11	∃x(Jx ∧ ¬Ox)	∃E 2, 3–10

NEW LINE NEW SUBPROOF

Congratulations! This proof is correct.

Question 1 : 16 points

Prove or Disprove [note: this is the argument about locked jpeg files you translatec
$\forall x(Lx \implies \neg Ox), \exists x(Lx \wedge Jx) \therefore \exists x(Jx \wedge \neg Ox)$

Type your text

Question 2 : 16 points

Prove or Disprove: $\forall \neg \quad Mx \vee Ljx), \forall x(Bx \implies Ljx), \forall x(Mx \vee Bx) \therefore \forall x Ljx$

In a company, Alex and Jordan
demonstrate the invalidity of the argument that "every employee has Leadership Skills." E
ach employee must either work in Marketing (M) or Business Development (B). Alex works in the Marketing department and does not possess Leadership Skills (¬Lj), while Jordan works in the Business Development department and does possess Leadership Skills (Lj). The premises state that if an employee is not in Marketing, they have Leadership Skills, and if an employee is in Business Development, they have Leadership Skills. Additionally, every employee must be in either Marketing or Business Development. While all these premises hold true—Alex is in Marketing and Jordan is in Business Development—the conclusion that every employee has Leadership Skills is false because Alex, who is in Marketing, does not have Leadership Skills

Question 3 : 16 points

Prove or Disprove: $Pa \vee Qb, Qb \implies (b = c), \neg Pa \therefore Qc$

let Pa be false ¬Pa, Qb be true, b = c, Qc be false
Pa V Qb is true bc Qb is true, Qb ⇒(b=c) is true because b=c, ¬P is true but Qc is false. Therefore this argument is not valid

Question 4 : 16 points

Prove or Disprove: $(\exists x Px) \wedge (\exists x Qx) \therefore \exists x(Px \wedge Qx)$

In a company with employees Alice and Bob, and skills Programming and Design, we can use the predicates P(x) and Q(x) to describe their skill sets. Alice knows Programming but does not know Design, while Bob knows Design but does not know Programming. Thus, there is an employee who knows Programming (Alice) and another who knows Design (Bob), satisfying the premises. ∃xP(x) and ∃xQ(x) However, there is no single employee who knows both Programming and Design, which means the conclusion ∃x (P(x) ∧Q(x)) is false.

Question 5 : 16 points

Prove or Disprove: $\forall x \forall y(Rxy \implies (x = y)) \therefore Rab \implies Rba$

Construct a proof for the argument: ∀x∀y(Rxy → x = y) ∴ Rab → Rba

1	∀x∀y(Rxy → x = y)	
2	∀y(Ray → a = y)	∀E 1
3	(Rab → a = b)	∀E 2
4	Rab	
5	a = b	→E 3, 4
6	Rba	=E 5 4
7	Rab → Rba	→I 4–6

Type your text

Question 6 : 20 points

Let D be the set of all data structures used in a particular programming project. Let L be the predicate representing Linked Lists (i.e. Lx is the statement "x is a Linked List"), and S is the predicate representing Stacks. Two facts are known about the project.

Fact1 is "Stacks are actually just a specialized type of Linked List".

Fact2 is "The project contains at least one Stack data structure".

The programming manager makes the following claim to her clients: "The project contains at least one Linked List data structure".

Translate that argument into symbols (the only predicates should be L and S, do not create your own) and decide whether the claim is logically valid or not. (If it is, screenshot the proof, and if it's not, provide a counterexample)

∃xSx, Sx⟹Lx Therefore: ∃xLx

1	∀x(Sx → Lx)	
2	∃xSx	
3	Sa → La	∀E 1
4	Sa	
5	La	→E 3, 4
6	∃xLx	∃I 5
7	∃xLx	∃E 2, 4–6

NEW LINE NEW SUBPROOF

Congratulations! This proof is correct.