1. Propositional Logic Review

Fill in this table. There are two valid negations each for A $\wedge$ B and A $\leftrightarrow$ B.

Expression	English Translation	Negation of Expression
A $\wedge$ B		
A $\vee$ B		
$\neg$A		
A $\to$ B		
A $\leftrightarrow$ B		
$\top$		
$\bot$		

2. First-Order Logic Review

Fill in this table:

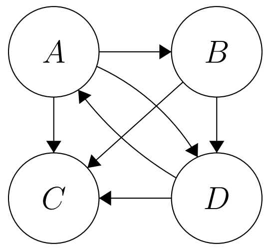
Expression	English Translation	Negation of Expression
$\forall x.(P(x))$		
$\exists x.(P(x))$		
$\forall x.(\neg P(x))$		
$\exists x.(\neg P(x))$		
$\forall x.(A(x) \to B(x))$		
$\exists x.(A(x) \wedge B(x))$		
$\forall x.(A(x) \to \neg B(x))$		
$\exists x.(A(x) \wedge \neg B(x))$		

Then, explain why writing $\exists x.(A(x) \to B(x))$ is usually a bad idea.

3. Translating First-Order Logic to English

For each statement, translate it into English, then decide whether it's true or false.

Interpersonal Dynamics: This diagram represents a set P of people named A, B, C and D. If there's an arrow from a person x to a person y, then person x loves person y. We'll denote this by writing $Loves(x, y)$.



a. $\forall y \in P.\ \exists x \in P.\ Loves(x, y)$

b. $\exists x \in P.\ \forall y \in P.\ Loves(x, y)$

c. $\exists x \in P.\ \forall y \in P.\ (x \neq y \rightarrow Loves(x, y))$

d. $\forall x \in P.\ \forall y \in P.\ (x \neq y \rightarrow (Loves(x, y) \lor Loves(y, x)))$

e. $\forall x \in P.\ \forall y \in P.\ (x \neq y \rightarrow (Loves(x, y) \leftrightarrow \neg Loves(y, x)))$

4. Translating English to First-Order Logic

Translate each statement into first-order logic, given these predicates: $HasHat(x)$ says that x is wearing a hat, $Dog(x)$ says that x is a dog, $Robot(x)$ says that x is a robot, and $Loves(x, y)$ says that x loves y.

a. All dogs wear hats.

b. There is a dog with a hat.

c. Dogs don't wear hats.

d. Some, but not all, robots wear hats.

e. For each dog, either the dog is wearing a hat or there's a robot the dog doesn't love (or both).

f. A robot loves a dog if and only if both the dog and the robot are wearing hats.

g. Some robot loves exactly one dog. (Hint: you can express "there is exactly one thing with a certain property" by saying "there is something with that property, and if something else has that property, then they're the same thing.")

h. There are at least two dogs. (Hint: you can express things about multiple objects by nesting quantifiers and, if necessary, checking that the two objects you are looking at are not the same. See the Guide to Logic Translation for more on this.)

5. Negating Statements in First-Order Logic

For each of the first-order logic formulas below, find a first-order logic formula that is the negation of the original statement. The only negations your final formula should have are direct negations of predicates. For example, the negation of the formula $\forall x.\ (P(x) \to \exists y.\ (Q(x) \land R(y)))$ could be found by pushing the negation from the outside inward as follows:

$$\neg(\forall x.\ (P(x) \to \exists y.\ (Q(x) \land R(y))))$$

$$\exists x.\ \neg(P(x) \to \exists y.\ (Q(x) \land R(y)))$$

$$\exists x.\ (P(x) \land \neg(\exists y.\ (Q(x) \land R(y))))$$

$$\exists x.\ (P(x) \land \forall y.\ \neg(Q(x) \land R(y)))$$

$$\exists x.\ (P(x) \land \forall y.\ (Q(x) \to \neg R(y)))$$

Show every step of the process of pushing the negation into the formula, along the lines of what is done above.

a. $\exists k.\ \big(Coder(k) \land Athlete(k) \land Painter(k)\big)$

   (Hint: Add parentheses to make the inside statement look more like a basic form.)

b. $\forall t.\ \big(Leafy(t) \land Thorny(t) \to Plant(t)\big)$

c. $\exists r.\ \big(Silly(r) \leftrightarrow \neg Serious(r)\big)$

d. $\exists u.\ \Big(Unicorn(u)\Big) \to \exists h.\ \Big(Horse(h) \land Magical(h)\Big)$

e. $\forall x.\ \Big(Person(x) \to \exists y.\ \exists z.\big(CanJuggle(x, y) \land \neg CanJuggle(x, z)\big)\Big)$

Key tips for negations:

- Go slowly, one step at a time.
- Understand the parentheses in the formula before you start the process.
- If you have a complicated expression, replace it with a symbol and negate it later. (Demoed in class)