Section 03: Quantifiers and Proofs

1. ctrl-z

Translate these logical expressions to English. For each of the translations, assume that domain restriction is being used and take that into account in your English versions.

Let your domain be all UW Students. Predicates $\texttt{143Student}(x)$ and $\texttt{311Student}(x)$ mean the student is in CSE 143 and 311, respectively. $\texttt{BioMajor}(x)$ means x is a bio major, $\texttt{DidHomeworkOne(x)}$ means the student did homework 1 (of 311). Finally $\texttt{KnowsJava}(x)$ and $\texttt{KnowsDeMorgan}(x)$ mean x knows Java and knows DeMorgan's Laws, respectively.

(a) $\forall x(\texttt{143Student}(x) \rightarrow \texttt{KnowsJava}(x))$ **Solution:**

> Every 143 student knows java.
>
> "If a UW student is a 143 student, then they know java" is a valid translation of the original sentence, but it is not taking advantage of the domain restriction.

(b) $\exists x(\texttt{143Student}(x) \wedge \texttt{BioMajor}(x))$ **Solution:**

> "There is a 143 student who is a bio major"
>
> "There is a UW student who is a 143 student and is a bio major" is a valid translation of the original sentence, but is not taking advantage of the domain restriction.

(c) $\forall x([\texttt{311Student}(x) \wedge \texttt{DidHomeworkOne(x)}] \rightarrow \texttt{KnowsDeMorgan}(x))$ **Solution:**

> If a 311 student does homework one then they know DeMorgan's Laws.

2. Domain Restriction

Translate each of the following sentences into logical notation. These translations require some of our quantifier tricks. You may use the operators $+$ and $\cdot$ which take two numbers as input and evaluate to their sum or product, respectively. Remember:

- To restrict the domain under a $\forall$ quantifier, add a hypothesis to an implication.
- To restrict the domain under an $\exists$ quantifier, AND in the restriction.
- If you want variables to be different, you have to explicitly require them to be not equal.

(a) Domain: Positive integers; Predicates: $\texttt{Even}, \texttt{Prime}, \texttt{Equal}$
"There is only one positive integer that is prime and even." **Solution:**

> $\exists x(\texttt{Prime}(x) \wedge \texttt{Even}(x) \wedge \forall y[\neg \texttt{Equal}(x, y) \rightarrow \neg(\texttt{Even}(y) \wedge \texttt{Prime}(y))])$

(b) Domain: Real numbers; Predicates: $\texttt{Even}, \texttt{Prime}, \texttt{Equal}$
"There are two different prime numbers that sum to an even number." **Solution:**

$\exists x \exists y(\texttt{Prime}(x) \wedge \texttt{Prime}(y) \wedge \neg Equal(x, y) \wedge \texttt{Even}(x + y))$

(c) Domain: Real numbers; Predicates: `Even`, `Prime`, `Equal`
"The product of two distinct prime numbers is not prime." **Solution:**

$\forall x \forall y([\texttt{Prime}(x) \wedge \texttt{Prime}(y) \wedge \neg \texttt{Equal}(x, y)] \rightarrow \neg \texttt{Prime}(xy))$

(d) Domain: Real numbers; Predicates: `Even`, `Prime`, `Equal`, `Postivite`, `Greater`, `Integer`
"For every positive integer, there is a greater even integer" **Solution:**

$\forall x(\texttt{Positive}(x) \wedge \texttt{Integer}(x) \rightarrow [\exists y(\texttt{Integer}(y) \wedge \texttt{Even}(y) \wedge \texttt{Greater}(y, x))])$
Or equivalently: $\forall x \exists y(\texttt{Positive}(x) \wedge \texttt{Integer}(x) \rightarrow (\texttt{Integer}(y) \wedge \texttt{Even}(y) \wedge \texttt{Greater}(y, x)))$

3. Quantifier Switch

Consider the following pairs of sentences. For each pair, determine if one implies the other, if they are equivalent, or neither.

(a) $\forall x \; \forall y \; P(x, y)$ $\quad$ $\forall y \; \forall x \; P(x, y)$ **Solution:**

These sentences are the same; switching universal quantifiers makes no difference.

(b) $\exists x \; \exists y \; P(x, y)$ $\quad$ $\exists y \; \exists x \; P(x, y)$ **Solution:**

These sentences are the same; switching existential quantifiers makes no difference.

(c) $\forall x \; \exists y \; P(x, y)$ $\quad$ $\forall y \; \exists x \; P(x, y)$ **Solution:**

These are only the same if P is symmetric (i.e., the order of the arguments doesn't matter). If the order of the arguments does matter, then these are different statements. For instance, if $P(x, y)$ is "$x < y$", then the first statement says "for every x, there is a corresponding y such that $x < y$", whereas the second says "for every y, there is a corresponding x such that $x < y$". In other words, in the first statement y is a function of x, and in the second x is a function of y.

If your domain of discourse is "positive integers", for example, the first is true and the second is false; but for "negative integers" the second is true while the first is false.

(d) $\forall x \; \exists y \; P(x, y)$ $\quad$ $\exists x \; \forall y \; P(x, y)$ **Solution:**

These two statements are usually different.

(e) $\forall x \; \exists y \; P(x, y)$ $\quad$ $\exists y \; \forall x \; P(x, y)$ **Solution:**

The second statement is "stronger" than the first (that is, the second implies the first). For the first, y is allowed to depend on x. For the second, one specific y must work for all x. Thus if the second is true,

whatever value of y makes it true, can also be plugged in for y in the first statement for every x. On the other hand, if the first statement is true, it might be that different y's work for the different x's and no single value of y exists to make the latter true.

As an example, let you domain of discourse be positive real numbers, and let $P(x, y)$ be $xy = 1$. The first statement is true (always take y to be $1/x$, which is another positive real number). The second statement is not true; it asks for a single number that always makes the product 1.

4. Domain Restriction Negated

When we negate a sentence with a domain restriction, the restriction itself remains intact (i.e. not negated) after we have fully simplified. That should make sense; if I claim something is true for every even number, I won't be convinced by you showing me an odd number. In this problem, you'll do the algebra to see why.

(a) Consider the statement $\neg\exists x\exists y(\texttt{Domain1}(x) \wedge \texttt{Domain2}(y) \wedge [P(x, y) \wedge Q(x, y)])$. We know that we should end up with $\forall x\forall y(\texttt{Domain1}(x) \wedge \texttt{Domain2}(y) \rightarrow [\neg P(x, y) \vee \neg Q(x, y)])$ (that is flip the quantifiers, rewrite the domain restriction, and negate the other requirements).

But it can help to see the full algebra written out – write out a step-by-step simplification to get the simplified form (you don't have to label with rules). **Solution:**

$$\begin{aligned}
\neg\exists x\exists y(\texttt{Domain1}(x) \wedge \texttt{Domain2}(y) \wedge [P(x, y) \wedge Q(x, y)]) &\equiv \forall x\neg[\exists y(\texttt{Domain1}(x) \wedge \texttt{Domain2}(y) \wedge [P(x, y) \wedge Q(x, y)])] \\
&\equiv \forall x\forall y\neg[(\texttt{Domain1}(x) \wedge \texttt{Domain2}(y) \wedge [P(x, y) \wedge Q(x, y)])] \\
&\equiv \forall x\forall y(\neg\,\texttt{Domain1}(x) \vee \neg\,\texttt{Domain2}(y) \vee \neg[P(x, y) \wedge Q(x, y)]) \\
&\equiv \forall x\forall y(\neg\,\texttt{Domain1}(x) \vee \neg\,\texttt{Domain2}(y) \vee [\neg P(x, y) \vee \neg Q(x, y)]) \\
&\equiv \forall x\forall y([\neg\,\texttt{Domain1}(x) \vee \neg\,\texttt{Domain2}(y)] \vee [\neg P(x, y) \vee \neg Q(x, y)]) \\
&\equiv \forall x\forall y(\neg[\texttt{Domain1}(x) \wedge \texttt{Domain2}(y)] \vee [\neg P(x, y) \vee \neg Q(x, y)]) \\
&\equiv \forall x\forall y([\texttt{Domain1}(x) \wedge \texttt{Domain2}(y)] \rightarrow [\neg P(x, y) \vee \neg Q(x, y)])
\end{aligned}$$

(b) Now do the same process for: $\neg\forall x\forall y([\texttt{Domain1}(x) \wedge \texttt{Domain2}(y)] \rightarrow [P(x, y) \wedge Q(x, y)])$ **Solution:**

$$\begin{aligned}
\neg\forall x\forall y([\texttt{Domain1}(x) \wedge \texttt{Domain2}(y)] \rightarrow [P(x, y) \wedge Q(x, y)]) &\equiv \exists x\neg[\forall y([\texttt{Domain1}(x) \wedge \texttt{Domain2}(y)] \rightarrow [P(x, y) \wedge Q(x, y)])] \\
&\equiv \exists x\exists y\neg[([\texttt{Domain1}(x) \wedge \texttt{Domain2}(y)] \rightarrow [P(x, y) \wedge Q(x, y)])] \\
&\equiv \exists x\exists y\neg[(\neg[\texttt{Domain1}(x) \wedge \texttt{Domain2}(y)] \vee [P(x, y) \wedge Q(x, y)])] \\
&\equiv \exists x\exists y(\neg\neg[\texttt{Domain1}(x) \wedge \texttt{Domain2}(y)] \wedge \neg[P(x, y) \wedge Q(x, y)]) \\
&\equiv \exists x\exists y(\texttt{Domain1}(x) \wedge \texttt{Domain2}(y) \wedge \neg[P(x, y) \wedge Q(x, y)]) \\
&\equiv \exists x\exists y(\texttt{Domain1}(x) \wedge \texttt{Domain2}(y) \wedge [\neg P(x, y) \vee \neg Q(x, y)])
\end{aligned}$$

5. Quantifier Ordering

Let your domain of discourse be a set of `Element` objects given in a list called `Domain`. Imagine you have a predicate $\texttt{pred}(x, y)$, which is encoded in the java method `public boolean pred(int x, int y)`. That is you call your predicate `pred` true if and only if the java method returns `true`.

(a) Consider the following Java method:

```
public boolean Mystery(Domain D){
    for(Element x : D) {
        for(Element y : D) {
```

```
            if(pred(x,y))
                return true;
        }
    }
}
```

`Mystery` corresponds to a quantified formula (for `D` being the domain of discourse), what is that formula?
Solution:

$\exists x \exists y(\texttt{pred}(x, y))$. If any combination of x and y causes pred to evaluate to true, we return true; that is we just want x, y to exist.

(b) What formula does `mystery2` correspond to

```
public boolean Mystery2(Domain D){
    for(Element x : D) {
        boolean thisXPass = false;
        for(Element y : D) {
            if(pred(x,y))
                thisXPass = true;
        }
        if(!thisXPass)
            return false;
    }
    return true;
}
```

Solution:

$\forall x \exists y(\texttt{pred}(x, y))$.

For a given x, when we come across a y that makes $\texttt{pred}(x, y)$ true, we set the given x to pass (so one y suffices for a given x) but we require **every** x to pass, so x is universally quantified. Since y is allowed to depend on x, we have x as the outermost variable.

6. Direct Proof

(a) Let the domain of discourse be integers. Define the predicates $\text{Odd}(x) := \exists k(x = 2k + 1)$, and $\text{Even}(x) := \exists k(x = 2k)$. Translate the following claim to predicate logic:

The sum of an even and odd integer is odd.

Solution:

$\forall n \forall m((\text{Even}(n) \wedge \text{Odd}(m)) \rightarrow \text{Odd}(n + m))$

(b) Prove that the claim holds. **Solution:**

Let n and m be arbitrary integers. Suppose n is even and m is odd. Then by definition of even, $n = 2k$ for some integer k. By definition of odd, $m = 2j + 1$ for some integer j. Then consider $n + m$:

$$\begin{aligned} n + m &= 2k + 2j + 1 \\ &= 2(k + j) + 1 \end{aligned}$$

Since k and j are integers, $k + j$ is an integer.

Then $n + m$ is 2 times an integer plus 1. Thus by definition of odd, $n + m$ is odd. Since n and m were arbitrary, the sum of any even and odd integer is odd.

7. Proof of Biconditional

(a) Let the domain of discourse be integers. Define the predicates $\text{Odd}(x) := \exists k(x = 2k + 1)$, and $\text{Even}(x) := \exists k(x = 2k)$. Translate the following claim to predicate logic:

For all integers n, $n - 4$ is even if and only if $n + 17$ is odd.

Solution:

$\forall n(\text{Even}(n - 4) \leftrightarrow \text{Odd}(n + 17))$

(b) Prove that the claim holds.

Solution:

$\Rightarrow$ Let n be an arbitrary integer. Suppose that $n - 4$ is even. Then by definition of even, $n - 4 = 2k$ for some integer k. Then observe that:

$n - 4 = 2k$	
$n + 17 = 2k + 21$	Adding 21 to both sides
$n + 17 = 2(k + 10) + 1$	Factoring

Thus $n + 17 = 2(k + 10) + 1$. Since k is an integer, $k + 10$ is an integer. So $n + 17$ is 2 times an integer plus 1. Thus by definition of odd, $n + 17$ is odd. Since n was arbitrary, we have shown that for all integers n that if $n - 4$ is even, then $n + 17$ is odd.

$\Leftarrow$ Let n be an arbitrary integer. Suppose $n + 17$ is odd. Then by definition of odd, $n + 17 = 2k + 1$ for some integer k. Then observe that:

$n + 17 = 2k + 1$	
$n - 4 = 2k + 1 - 21$	Subtracting 21 from both sides
$n - 4 = 2(k - 10)$	Factoring

Thus $n - 4 = 2(k - 10)$. Since k is an integer, $k - 10$ is an integer. So $n - 4$ is 2 times an integer. So by definition of even, $n - 4$ is even. Since n was arbitrary, we have shown that for all integers n, if $n + 17$ is odd, then $n - 4$ is even.

8. Formal Proof

Show that $\neg p$ follows from $\neg(\neg r \vee t)$, $\neg q \vee \neg s$ and $(p \to q) \wedge (r \to s)$. **Solution:**

1.	$\neg(\neg r \vee t)$	[Given]
2.	$\neg q \vee \neg s$	[Given]
3.	$(p \to q) \wedge (r \to s)$	[Given]
4.	$\neg\neg r \wedge \neg t$	[DeMorgan's Law: 1]
5.	$\neg\neg r$	[Elim of $\wedge$: 4]
6.	r	[Double Negation: 5]
7.	$r \to s$	[Elim of $\wedge$: 3]
8.	s	[MP, 6,7]
9.	$\neg\neg s$	[Double Negation: 8]
10.	$\neg s \vee \neg q$	[Commutative: 2]
11.	$\neg q$	[Elim of $\vee$: 10, 9]
12.	$p \to q$	[Elim of $\wedge$: 3]
13.	$\neg q \to \neg p$	[Contrapositive: 12]
14.	$\neg p$	[MP: 11,13]

9. There is an implication

Implications are uncommon under existential quantifiers. Consider this expression (which we'll call "the original expression"): $\exists x(\mathtt{P}(x) \to \mathtt{Q}(x))$

(a) Suppose that $\mathtt{P}(x)$ is not always true (i.e. there is an element in the domain for which $\mathtt{P}(x)$ is false). Explain why the original expression is true in this case. (1-2 sentences should suffice. If you prefer, you may give a formal proof instead).

Solution:

Consider some y such that $\mathtt{P}(y)$ is false. Plugging y in for x we get a true implication (by vacuous truth). Thus this y is the y that must exist to make the quantified statement true.

(b) Suppose that $\mathtt{P}(x)$ is always true (i.e. $\forall x\, \mathtt{P}(x)$). There is a simpler statement which conveys the meaning of the original expression (i.e. is equivalent to it for all domains and predicates. By simpler, we mean "uses fewer symbols"). Give that expression, and briefly (1-2 sentences) explain why it works.

Solution:

$\exists x\, \mathtt{Q}(x)$. Since $\mathtt{P}(x)$ is always true, the hypothesis is always satisfied, so the implication is true if and only if the conclusion is true.