CSCI 2824: Discrete Structures

Lecture 7: Nested Quantifiers

Reminders

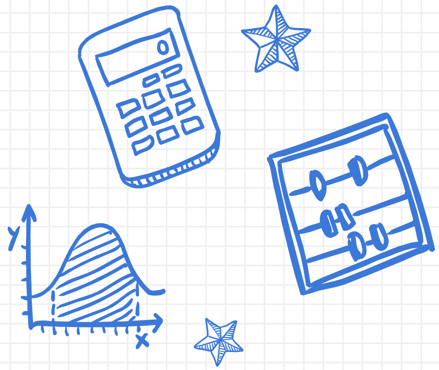
Submissions:

- Homework 2: Fri 9/13 at noon – 1 try
- Quizlet 2: due Wednesday 9/11 at **8pm**

Disabilities forms – please bring them by Friday

Readings:

- 1.4-1.6 this week
- 1.6-1.8 through next week



Predicate Logic

What did we do last time?

Predicates and **propositional functions**

-- a more flexible framework to describe the truthiness of the world

Quantifiers:

- **Universal quantifier:** $\forall x\ P(x)$ means "for all x in my **domain**, $P(x)$"
- **Existential quantifier:** $\exists x\ P(x)$ means "there exists an x in my **domain**, $P(x)$"

Recap

Two rules we found to be true:

1. $\forall x\,(P(x) \land Q(x)) \equiv \forall x\,P(x) \land \forall x\,Q(x)$

distribution of universal quantifier ($\forall$) … **only** over conjunctions

2. $\exists x\,(P(x) \lor Q(x)) \equiv \exists x\,P(x) \lor \exists x\,Q(x)$

distribution of existential quantifier ($\exists$) … **only** over disjunctions

Example: Are these equivalent?

$$\exists x\, (P(x) \land Q(x)) \equiv \exists x\, P(x) \land \exists x\, Q(x)$$

Answer: No - (**FYOG**, think about it!)

This is **False**, which we can demonstrate with a **counterexample**.

Example: Are these equivalent?

$$\exists x\, (P(x) \wedge Q(x)) \equiv \exists x\, P(x) \wedge \exists x\, Q(x)$$

Answer: No - (**FYOG**, think about it!)

This is **False**, which we can demonstrate with a **counterexample**.

Let $P(x)$ represent "x is an even integer" and let $Q(x)$ represent "x is an odd integer", and suppose the domain is all integers.

Then the right-hand side is certainly True - there exist integers that are even, and there exist integers that are odd.

But the left-hand side is certainly False - there do not exist any integers that are both even *and* odd.

Predicate Logic – Negation (last time)

TABLE 2 **De Morgan's Laws for Quantifiers.**

Negation	*Equivalent Statement*	*When Is Negation True?*	*When False?*
$\neg\exists x P(x)$	$\forall x \neg P(x)$	For every x, $P(x)$ is false.	There is an x for which $P(x)$ is true.
$\neg\forall x P(x)$	$\exists x \neg P(x)$	There is an x for which $P(x)$ is false.	$P(x)$ is true for every x.

And then we had the distribution laws

- $\forall x\, (P(x) \land Q(x)) \equiv \forall x\, P(x) \land \forall x\, Q(x)$
- $\exists x\, (P(x) \lor Q(x)) \equiv \exists x\, P(x) \lor \exists x\, Q(x)$

With these rules, and the logical equivalences we found for regular propositions, we can prove all kinds of equivalences of quantifier propositions

Predicate Logic

Example: Prove that $\neg\forall x\,(P(x) \rightarrow Q(x)) \equiv \exists x(P(x) \land \neg Q(x))$

Use DeMorgan's Laws and other equivalency rules

Predicate Logic

Example: Prove that $\neg\forall x\,\big(P(x) \rightarrow Q(x)\big) \equiv \exists x\big(P(x) \land \neg Q(x)\big)$

Solution:

$\neg\forall x\,\big(P(x) \rightarrow Q(x)\big) \equiv \exists x \neg(P(x) \rightarrow Q(x))$ De Morgan's Laws

$\neg(P(x) \rightarrow Q(x)) \equiv \neg\big(\neg P(x) \lor Q(x)\big)$ RBI

$\neg\big(\neg P(x) \lor Q(x)\big) \equiv \big(P(x) \land \neg Q(x)\big)$ Negation

Let's read it out loud to see if it makes sense.
"It is not the case that for all x, if P(x) then Q(x)."
"There is some x such that P(x) and not Q(x)."

From Practice Problems

Example 2: Think of a domain, and specific propositional functions $P(x)$ and $Q(x)$ to illustrate this equivalence

$$\neg \forall x\, (P(x) \rightarrow Q(x)) \equiv \exists x\, (P(x) \land \neg Q(x))$$

Solution: Let x be the set of all shapes, $P(x)$ mean x is a polygon, and $Q(x)$ mean x is a rectangle.

The first proposition says it's not the case that if x is a polygon then it is necessarily a rectangle

The second proposition says that there exists a shape that is a polygon and is not a rectangle (e.g. a triangle)

Predicate Logic

Translating from English into Logical Expressions

Example: Translate the following into symbols: “Every student in CSCI 2824 has passed Calculus 1.” Let the domain be all the students in CSCI 2824.

Predicate Logic

Translating from English into Logical Expressions

<u>Example:</u> Translate the following into symbols: “Every student in CSCI 2824 has passed Calculus 1.” Let the domain be all the students in CSCI 2824.

Rewrite: For every student in CSCI 2824, that student has passed Calc1.

We introduce **C(x),** which is the statement “**x has passed Calc I**.”

If the domain for x consists of the students in the class, we can translate our statement as **$\forall x\ C(x)$**

Predicate Logic

Translating from English into Logical Expressions

Example: Translate the following into symbols: “Every student in CSCI 2824 has passed Calculus 1.” but use the domain, all students at CU.

Predicate Logic

Translating from English into Logical Expressions

Example: Translate the following into symbols: “Every student in CSCI 2824 has passed Calculus 1.” but use the domain, all students at CU.

Rewrite: “For every CU student x, if x is a student in CSCI 2824, then x has passed Calc 1.”

We introduce **S(x),** which is the statement “**x is in CSCI 2824**.”

If the domain for x consists of the CU students, we can translate our statement as $\forall x\ (S(x) \rightarrow C(x))$

Predicate Logic

Translating from English into Logical Expressions

<u>Example:</u> Translate the following into symbols: "Every student in CSCI 2824 has passed Calculus 1." but use the domain, all students at CU.

Answer: $\forall x\ (S(x) \rightarrow C(x))$

Note: This does **not** work as a translation: $\forall x\ (S(x) \wedge C(x))$

(That would say that all students **<u>at CU</u>** are in CSCI 2824 and have passed Calculus 1.)

Predicate Logic

Translating from English into Logical Expressions

Example: Let the domain be the set of all CU students, and translate: "Every student in CSCI 2824 is either taking Data Structures, or has already passed it."

Predicate Logic

Translating from English into Logical Expressions

Example: Let the domain be the set of all CU students, and translate: "Every student in CSCI 2824 is either taking Data Structures, or has already passed it."

"For every CU student x, if x in this class, x has the property that x has taken Data Structures or x has passed Data Structures."

Predicate Logic

"For every CU student x, if x in this class, x has the property that x has taken Data Structures or x has passed Data Structures."

Solution:

Let $S(x)$ represent "is a CSCI 2824 student".

Let $P(x)$ represent "has passed Data Structures".

Let $D(x)$ represent "is taking Data Structures".

Then our statement becomes $\boldsymbol{\forall x\, (S(x) \rightarrow (D(x) \vee P(x)))}$

Predicate Logic

"For every CU student x, if x in this class, x has the property that x has taken Data Structures or x has passed Data Structures."

Solution:
Let S(x) represent "is a CSCI 2824 student".
Let P(x) represent "has passed Data Structures".
Let D(x) represent "is taking Data Structures".
Then our statement becomes $\boldsymbol{\forall x\,(S(x) \rightarrow (D(x) \vee P(x)))}$

Although it might be more accurate with real-world experience to write it as

$\boldsymbol{\forall x\,(S(x) \rightarrow (D(x) \oplus P(x)))}$ (with the exclusive or, because most people don't take the class again just for fun…)

Quantifiers as loops

A Computer Sciency Way of Viewing Quantifiers

Think of quantified statements as loops that do logic checks

Example: $\forall x\ P(x)$

In [ ]:
```
for x in domain:
    if P(x) == False:
        return False
return True
```

- If we find an x in domain where $P(x)$ is False, return False
- If we make it through loop then return True

Quantifiers as loops

A Computer Sciency Way of Viewing Quantifiers

Think of quantified statements as loops that do logic checks

Example: $\exists x\ P(x)$

In [ ]:
```
for x in domain:
    if P(x) == True:
        return True
return False
```

- If we find an x in domain where $P(x)$ is True, return True
- If we make it through loop without finding one, return False

Nested Quantifiers

Where things **really** get interesting, because we include multiple quantifiers for a propositional function.

Example: Consider the domain of all real numbers. What does the following statement mean?

$$\forall x \, \exists y \, (x + y = 0)$$

Nested Quantifiers

Where things **really** get interesting, because we include multiple quantifiers for a propositional function.

Example: Consider the domain of all real numbers. What does the following statement mean?

$$\forall x\ \exists y\ (x + y = 0)$$

For all x, there exists an y such that x + y = 0

T or F?

Nested Quantifiers - Can we express this logic check with a loop?

$\forall x \, \exists y \, P(x,y)$ – "For all x, there exists an y such that P(x,y)"

Nested Quantifiers

Nested Quantifiers as Loops

Example: $\forall x\ \exists y\ P(x, y)$?

In [ ]:

```
for x in domain:
    exists_y = False
    for y in domain:
        if P(x,y) == True:
            exists_y = True
    if exists_y == False:
        return False
return True
```

- If we make it through y-loop without finding a True, return False
- If we make it through entire x-loop then return True

Nested Quantifiers – Example: What is the output? T or F

Nested Quantifiers as Loops

Example: $\forall x\ \exists y\ (x + y = 0)$?

In [7]:
```
def check_additive_inverse(domain):

    for x in domain:
        exists_y = False
        for y in domain:
            if x + y == 0:
                exists_y = True
        if exists_y == False:
            return False
    return True

domain = [-3, -2, -1, 0, 1, 2, 3]
check_additive_inverse(domain)
```

Nested Quantifiers – Example: What is the output? T or F

Nested Quantifiers as Loops

Example: $\forall x\ \exists y\ (x + y = 0)$?

In [8]:
```
def check_additive_inverse(domain):

    for x in domain:
        exists_y = False
        for y in domain:
            if x + y == 0:
                exists_y = True
        if exists_y == False:
            return False
    return True

domain = [-2, -1, 0, 1, 2, 3]
check_additive_inverse(domain)
```

Nested Quantifiers – Example

Nested Quantifiers as Loops

Example: $\forall x\, \forall y\, P(x, y)$?

In [ ]:
```
for x in domain:
    for y in domain:
        if P(x,y) == False:
            return False
return True
```

- If we ever find an (x, y)-pair that makes $P(x, y)$ False, return False
- If we make it through both loops, return True

Nested Quantifiers

Example: How could we express the law of **commutation of addition** (that is, that $x + y = y + x$)?

Nested Quantifiers

Example: How could we express the law of **commutation of addition** (that is, that $x + y = y + x$)?

- If it's a law, then it must work for **all x** and **all y** in the domain.
- Question: What happens if we change the order of $\forall x \, \forall y$?
  - Nothing!

$$\forall x \, \forall y \, (x + y = y + x) \equiv \forall y \, \forall x \, (x + y = y + x)$$

Nested Quantifiers

Let's go back to the previous example:

Example: $\forall x \, \exists y \, (x + y = 0)$

Question: What happens if we change the order here?

Answer: A lot! The new expression $\exists y \, \forall x \, (x + y = 0)$ says

- ``There exists some number y such that for every x out there, $x + y = 0$"

Can you think of such a number?

Nested Quantifiers

Rules for Switching Quantifiers:

- OK to switch $\forall x$ and $\forall y$
- OK to switch $\exists x$ and $\exists y$ Check that this is true!)
- **NOT** OK to switch $\forall x$ and $\exists y$

Nested Quantifiers

Example: Now we'll switch the domain to all real numbers

How can you express the fact that all numbers of have a **multiplicative inverse**

Nested Quantifiers

Example: Now we'll switch the domain to all real numbers

How can you express the fact that all numbers of have a **multiplicative inverse**

(That is, a number we can multiply the original by to get 1.)

Nested Quantifiers

Example: Now we'll switch the domain to all real numbers

How can you express the fact that all numbers of have a **multiplicative inverse**

(That is, a number we can multiply the original by to get 1.)

Nested Quantifiers

Solution: First off, is this even true? Do *all* real numbers have a multiplicative inverse?

- Answer: No, but all **nonzero** numbers do!

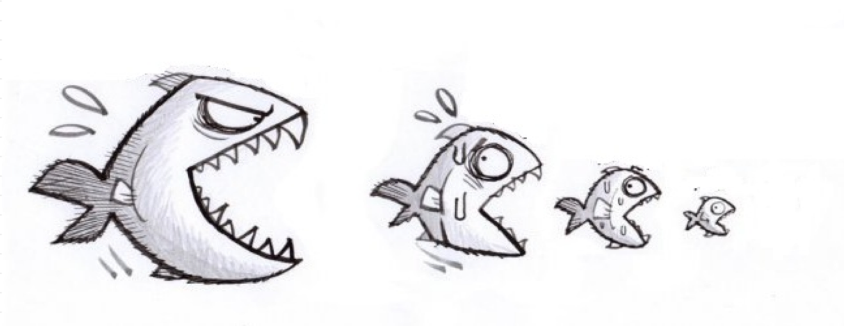
First, in plain English:

"For all x that aren't 0, there exists some number y such that $xy = 1$."

- Note that "that aren't 0" is a *condition* that we need to satisfy first in order to move on to the second part of this statement. ⇒ suggests we will need to use a conditional!
- So maybe: $\forall x\ ((x \neq 0) \rightarrow \exists y\ (xy = 1))$

Predicate and quantifier logic

Example: How can you express the fact that there are an infinite number of natural numbers? (Again, that's ℕ = {0, 1, 2, 3, …})

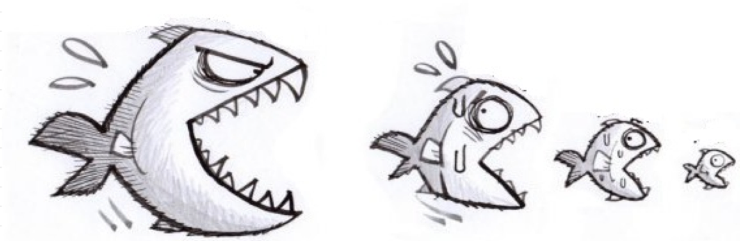


Predicate and quantifier logic

Example: How can you express the fact that there are an infinite number of natural numbers? (Again, that's $\mathbb{N} = \{0, 1, 2, 3, \ldots\}$)

$\forall x \, \exists y \, (y > x)$

(there's always a bigger fish)



Nested Quantifiers

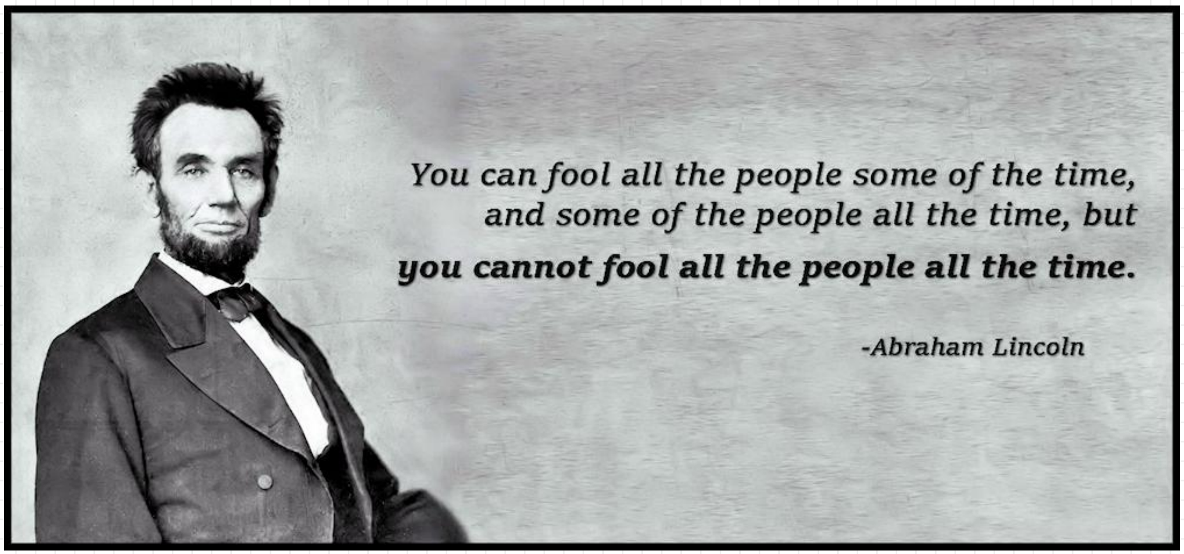
Example: Translate the following statement

"… ont pû tromper quelques hommes, ou les tromper tous dans certains lieux & en certains tems, mais non pas tous les hommes, dans tous les lieux & dans tous les siécles."

Nested Quantifiers

Example: Translate the following statement

"… ont pû tromper quelques hommes, ou les tromper tous dans certains lieux & en certains tems, mais non pas tous les hommes, dans tous les lieux & dans tous les siécles."



Nested Quantifiers

Let’s take it slow …

Example: Translate the following statement

“You can fool some of the people all of the time.”

Nested Quantifiers

Let's take it slow …

Example: Translate the following statement

"You can fool some of the people all of the time."

Solution:
Let $F(p, t)$ represent "you can fool person p at time t"
Let the domain for p be all people
Let the domain for t be all times
Then we have $\exists p \, \forall t \, F(p, t)$

Nested Quantifiers

Let's take it slow …

Example: Translate the following statement

"You can fool all of the people some of the time."

Nested Quantifiers

Let’s take it slow …

Example: Translate the following statement

“You can’t fool all of the people all of the time.”

Nested Quantifiers

Example: Translate the following statement

"You can't fool all of the people all of the time."
"It is not the case that for every person, for all times, they can be fooled."

Nested Quantifiers

Example: Translate the statement "You can't fool all of the people all of the time."

Solution:

- "It is not the case that for every person, for all times, they can be fooled."
- In logical symbols:

  $$\neg(\forall p \, \forall t \, F(p, t))$$

- What if we push the negation through?

  $$\neg(\forall p \, \forall t \, F(p, t)) \equiv \exists p \, \neg(\forall t \, F(p, t)) \equiv \exists p \, \exists t \, \neg F(p, t)$$

- So "there exists some person for some time that can't be fooled" (but that reads a bit more awkwardly)

Nested Quantifiers

Quantifications with more than two quantifiers are also common

Example: Let $Q(x, y, z)$ mean "$x + y = z$". What are the truth values of

- $\forall x \, \forall y \, \exists z \, Q(x, y, z)$
- $\exists z \, \forall x \, \exists y \, Q(x, y, z)$

Nested Quantifiers

End of Representational Logic

- We now know how to represent standard propositions
- We know how to represent propositions with quantifiers
- We know how to prove and derive logical equivalences

Next Time We Start Learning to Argue

- Rules of inference
- Valid and sound arguments
- Proof types and strategies

Extra Practice

EX. 1 Cook up an example of the form $\exists x \; \forall y \; P(x, y)$ that is True, and write a Python function with nested for-loops that checks it!

EX. 2 Cook up an example of the form $\exists x \; \exists y \; Q(x, y)$ that is True, and write a Python function with nested for-loops that checks it!

EX. 3 Is it OK to switch the order of $\exists x \; \exists y$?

Question: What changes if we write it as "There exists an integer y and an integer x such that $x^2 + y^2 = 25$"?

EX. 4 How could you express that if you multiply two negative numbers together you get a positive number?

EX. 5 How could you express that the real numbers have a **multiplicative identity**. That is, that there's a number out there that when you multiply something by it, you get the same thing back. (**Note**: this is literally saying that the number 1 is a thing)

Solutions

EX. 1 Cook up an example of the form $\exists x\ \forall y\ P(x, y)$ that is True, and write a Python function with nested for-loops that checks it!

Solution: How about $\exists x\ \forall y\ xy = 0$ (essentially, 0 exists)

In [12]:
```
def check_multiply_to_zero(domain):

    for x in domain:
        all_y = True
        for y in domain:
            if x*y != 0:
                all_y = False
        if all_y == True:
            return True

    return False

domain = [-3, -2, -1, 0, 1, 2, 3]
check_multiply_to_zero(domain)
```

Out[12]:
```
True
```

EX. 2 Cook up an example of the form $\exists x \, \exists y \, Q(x, y)$ that is True, and write a Python function with nested for-loops that checks it!

Solution: How about $\exists x \, \exists y \; x^2 + y^2 = 25$

In [15]:
```
def check_sum_of_squares(domain):

    for x in domain:
        for y in domain:
            if x**2 + y**2 == 25:
                return True

    return False

domain = [ 0, 1, 2, 3, 4, 5]
check_sum_of_squares(domain)
```

Out[15]:
```
True
```

EX. 3 Is it OK to switch the order of $\exists x\ \exists y$?

Solution: Totally. Consider the example "There exists an integer x and an integer y such that $x^2 + y^2 = 25$".

This is true because we can let $x = 3$ and $y = 4$

Question: What changes if we write it as "There exists an integer y and an integer x such that $x^2 + y^2 = 25$"?

Answer: Literally nothing

EX. 4 How could you express that if you multiply two negative numbers together you get a positive number?

Solution: We want to say that if we take any pair of numbers, if those numbers are negative their product is positive.

How about

$$\forall x\, \forall y\, (((x < 0) \wedge (y < 0)) \rightarrow (xy > 0))$$

EX. 5 How could you express that the real numbers have a **multiplicative identity**. That is, that there's a number out there that when you multiply something by it, you get the same thing back. (**Note**: this is literally saying that the number 1 is a thing)

Solution: We want to say

``There exists a number such that for any x when you multiply that number by x the result is x"

How about

$$\exists y \ \forall x \ (xy = x)$$