University of Colorado **Boulder**

Department of Computer Science

CSCI 2824: Discrete Structures

Chris Ketelsen

Lecture 6:
Nested Quantifiers

Nested Quantifiers

Last Time:

- Introduced predicates and propositional functions
- Started on universal and existential quantifiers

Universal Quantifier:

- $\forall x\ P(x)$: "For all x in my domain $P(x)$ is true "

Existential Quantifier:

- $\exists x\ P(x)$: "There exists an x in my domain s.t. $P(x)$ is true"

Nested Quantifiers

Warm-Up Problems: Let the domain for x be the set of all Natural Numbers, $\mathbb{N} = \{0, 1, 2, \ldots\}$

Example: Determine the truth value of $\forall n \; (3n \leq 4n)$

$n = 0$

$3 \cdot 0 \leq 4 \cdot 0$

$3 \cdot 7 \leq 4 \cdot 7$

Nested Quantifiers

Warm-Up Problems: Let the domain for x be the set of all Natural Numbers, $\mathbb{N} = \{0, 1, 2, \ldots\}$

Example: Determine the truth value of $\forall n\ (3n \leq 4n)$

This is true. **Q**: What if the domain was the set of all integers?

Example: Determine the truth value of $\exists x\ (x^2 = x)$

Nested Quantifiers

Warm-Up Problems: Let the domain for x be the set of all Natural Numbers, $\mathbb{N} = \{0, 1, 2, \ldots\}$

Example: Determine the truth value of $\forall n\ (3n \leq 4n)$

This is true. **Q**: What if the domain was the set of all integers?

Example: Determine the truth value of $\exists x\ (x^2 = x)$

This is true. We just need to find one x in the domain that works, and in this case there are two: $x = 0$ and $x = 1$.

Nested Quantifiers

Last time we showed the following equivalences

DeMorgan's Laws for Quantifiers:

- $\neg\forall x\, P(x) \equiv \exists x\, \neg P(x)$
- $\neg\exists x\, P(x) \equiv \forall x\, \neg P(x)$

Distribution Laws for Quantifiers:

- $\forall x\, (P(x) \wedge Q(x)) \equiv \forall x\, P(x) \wedge \forall x\, Q(x)$
- $\exists x\, (P(x) \vee Q(x)) \equiv \exists x\, P(x) \vee \exists x\, Q(x)$

Note: Distribution of $\forall$ over $\vee$ and $\exists$ over $\wedge$ didn't work

Nested Quantifiers

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

Nested Quantifiers

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

Nested Quantifiers

Interesting things happen when we include multiple quantifiers

Example: What does this say: $\forall x (\exists y\ (x + y = 0))$?

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

$\Rightarrow \forall x\ Q(x)$

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

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

Let $R(x) = \forall y\ (xy = 0)$

$\exists x\ R(x)$

Nested Quantifiers

Nested Quantifiers

Interesting things happen when we include multiple quantifiers

Example: What does this say: $\forall x \; \exists y \; (x + y = 0)$?

It really helps to read these outloud: ``For all x, there exists a y, such that the sum of x and y is zero"

What do you think? Is this true or false?

Nested Quantifiers

Nested Quantifiers

Interesting things happen when we include multiple quantifiers

Example: What does this say: $\forall x \, \exists y \, (x + y = 0)$?

It really helps to read these outloud: ``For all x, there exists a y, such that the sum of x and y is zero"

What do you think? Is this true or false?

This is totally **true.** It's the expression of the fact that all numbers have an **additive inverse**.

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

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)
```

Out[7]:
```
True
```

Nested Quantifiers

Nested Quantifiers as Loops

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

In [8]:
```python
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)
```

Out[8]:
```
False
```

Nested Quantifiers

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

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

Nested Quantifiers

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

How about $\forall x \, \forall y \; (x + y = y + x)$

Nested Quantifiers

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

Nested Quantifiers as Loops

EFY: Cook up an example of a statement 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!

EFY: Cook up an example of a statement 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!

Nested Quantifiers

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

How about $\forall x\ \forall y\ (x + y = y + x)$

Question: What happens if we change the order of $\forall x$ and $\forall y$?

Nested Quantifiers

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

How about $\forall x \, \forall y \, (x + y = y + x)$

Question: What happens if we change the order of $\forall x$ and $\forall y$?

So we'd have $\forall y \, \forall x \, (x + y = y + x)$

Answer: Not much! Still looping over all pairs of x's and y's

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?

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

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

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?

Me neither! In fact, after switching the order of the quantifiers the proposition becomes false.

Nested Quantifiers

Rules for Switching Quantifiers:

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

OK, let's do a bunch more examples

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 that you can multiply by to get 1?

First of all, is it really true that **all** numbers of have a multiplicative inverse?

1: DOMAIN = $\mathbb{R} - \{0\}$ $\quad \forall x \exists y \, (xy = 1)$

2: DOMAIN = $\mathbb{R}$ $\quad \forall x \exists y \, ((xy = 1) \vee (x = 0))$

* $\forall x \, [\exists y \, (xy = 1) \vee (x = 0)]$

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 that you can multiply by to get 1?

First of all, is it really true that **all** numbers of have a multiplicative inverse?

Nope! But all **nonzero** numbers do

So how could we say this with quantifiers?

Nested Quantifiers

Let's say it in logic-y English

``For all x's that aren't zero, there exists a y such that $xy = 1$"

Nested Quantifiers

Let's say it in logic-y English

``For all x's that aren't zero, there exists a y such that $xy = 1$"

Note that x not being zero is a **condition** that has to happen before we consider looking for an inverse. Let's rephrase:

``For all x, if $x \neq 0$ then there exists a y such that $xy = 1$"

How about

$$\forall x\ ((x \neq 0) \rightarrow \exists y\ (xy = 1))$$

Nested Quantifiers

Example: How could you express that there are an infinite number of natural numbers?

$\forall x \, \exists y \, (y = x + 1)$ $\forall x \, \exists y \, (y > x)$

Nested Quantifiers

Example: How could you express that there are an infinite number of natural numbers?

If domains for x and y are the set of natural numbers, we could say

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

This just says that every natural number has a number that is larger

Nested Quantifiers

EFY: How could you express that if you multiply two negative numbers together you get a positive number?

EFY: 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)

"YOU CAN FOOL AT LEAST 2 PEOPLE ALL OF THE TIME"

$$\exists p \, \exists q \, \forall t \, [F(p,t) \wedge F(q,t) \wedge (p \neq q)]$$

Nested Quantifiers

OK, lets practice some non-mathy translations

Example: Translate the statement ``You can fool some of the people all of the time"

We need to define a propositional function that says a person can be fooled at a particular time

Let $F(p, t)$ represent ``you can fool person p at time t"

Then we have $\exists p \, \forall t \, F(p, t)$

Nested Quantifiers

Example: Translate the statement ``You can fool all of the people some of the time"

To me, this one is actually kinda ambiguous.

Does it mean ``There is a time when you can fool all of the people"?

In which case we would have $\exists t\ \forall p\ F(p, t)$

Or does it mean ``Each person has a time that they could be fooled"?

In which case we would have $\forall p\ \exists t\ F(p, t)$

Rule of Thumb: Logic and mathematics are **precise** but language **ISN'T** so you have to be cautions

Nested Quantifiers

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

This works out to be $\neg(\forall p\ \forall t\ F(p,t))$

What would happen if we pushed the negation through?

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

which translates to the equivalent (but more awkward) statement

``There is some person at some time that can't be fooled"

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

Example: One more! Translate the following using quantifiers:

Babies are illogical. Nobody is despised who can manage a crocodile. Illogical people are despised. Therefore, babies cannot manage crocodiles

Let $B(x)$ mean "x is a baby", $L(x)$ mean "x is logical", $C(x)$ mean "x can handle a crocodile", and $D(x)$ mean "x is despised"

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

Nested Quantifiers

EFY: 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

Nested Quantifiers

EFY: 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
```

Nested Quantifiers

EFY: 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

Nested Quantifiers

EFY: 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

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

Nested Quantifiers

EFY: 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

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