CSCI 2824: Discrete Structures

Lecture 7: Nested Quantifiers

Rachel Cox

Department of Computer Science

Announcements & Reminders

- Written HW2 due by noon on Friday to Gradescope.
- Quizlet 02 Due tonight by 8 PM

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 $\forall n\ (3n \leq 4n)$

True.

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

True $\qquad x = 0, 1$

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:

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

Note: Distribution of $\forall$ over $\lor$ and $\exists$ over $\land$ 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

domain: all real numbers

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?

True! This reflects the additive inverse property.

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)
```

$x = -3$

Out[7]: True

Nested Quantifiers

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)
```

$x = 3$

$3 - 2$
$3 - 1$
$3 + 0$
$3 + 1$
$3 + 2$
$3 + 3$

Out[8]: False

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 $x + y = y + x$)?

Let domain be all real numbers.

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

ok to interchange the order.

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

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?

False.

Nested Quantifiers

Rules for Switching Quantifiers:

- OK to switch $\forall x$ and $\forall y$
- OK to switch $\exists x$ and $\exists y$
- **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**

e.g. $x = 5$ then $y = \frac{1}{5}$ $5 \cdot \frac{1}{5} = 1$

what if $x = 0$, ... does not have a multiplicative inverse!

For every real number x, except 0, there exists a multiplicative inverse, y, such that $xy = 1$

1) domain $\mathbb{R} - \{0\}$ $\forall x \exists y (xy = 1)$

2) domain $\mathbb{R}$ $\forall x (x \neq 0 \rightarrow \exists y (xy = 1))$

Nested Quantifiers

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

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

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

$F(p,t)$: You fool person p at time t.

$$\boxed{\exists p\ \forall t\ F(p,t)}$$

Nested Quantifiers

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

$F(p,t)$

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

vs.

$\exists t \ \forall p \ F(p,t)$ *

Nested Quantifiers

$-2 \cdot 3 = 2 \cdot -3$

$\neg \forall x \exists y \equiv \exists x \neg \exists y$

$\equiv \exists x \forall y \neg$

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

$$\boxed{\neg(\forall p \, \forall t \; F(p,t))}$$

$$\exists p \, \exists t \; \neg F(p,t)$$

Nested Quantifiers

Domain: Real numbers, $\mathbb{R}$

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

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

cycle through $\mathbb{R}$

$z^* = x + y$

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]:
```python
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

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