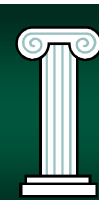
Predicate Logic

Part 9

1



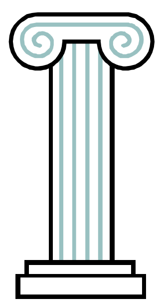
Predicates

Just the facts…

2

Predicates

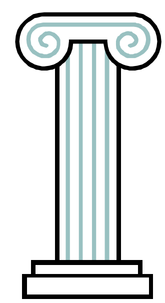
- A predicate is a statement about one or more variables
- It is stated as a fact – being true for the data provided



3

Predicates

- Predicates express *properties*
- These can apply to a single entity or *relations* which may hold on more than one individual



4

Predicate Notation

- It follows the same basic syntax as function calls in Java (and most programming languages)
- However, type case is important:
  - constants start with lower case letters
  - predicates start with upper case letters

5

Single variable predicate

- Predicates can have one variable (at a minimum)
- The following sentence states one that the cat named Pattycakes has the "sleepy" property

```
"Pattycakes The Cat is sleepy"
```


6

Single variable predicate

- Alternatively, we can write that property in predicate form
- "Sleepy" predicate for "Pattycakes" is true
- Note the uppercase and lowercase!

```
Sleepy(pattycakes)
```


7

Two Variable Predicate

- Predicates can have multiple variables (unlimited actually… well within reason)
- The following is a classic example of a two-variable relationship

```
x < y
```


8

Two Variable Predicate

- The LessThan predicate is true for x, y

```
LessThan(x, y)
```


9

Predicates Summary

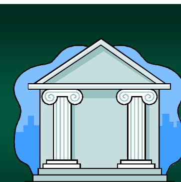
- 1-place predicates assign properties to individuals:
  - ___ is a cat
  - ___ is sleepy
- 2-place assign relations to a pair
  - ___ is sleeping on ___
  - ___ is the capitol of ___

10

Predicates Summary

- 3-place predicates assign relations to triples
  - ___ wants ___ to _____
  - Cat named ____ likes to ___ on ____
- Etc…

11



Quantified Statements

More Symbols

12

Quantified Statements

- Sometimes we want to say that *every element in the universe* has some property
- Let's say the universe is the people in this Zoom "room" & we want to say "*everyone in the room is awake*"

13

Propositional Approach #1

- We can write out a descriptive sentence
- Shortcomings:
  - it is monolithic an inflexible
  - not "mathematical" enough

```
Everyone in this room is awake.
```


14

Propositional Approach #2

- We can also write out that sentence using a long list of predicates
- So, we list them all <u>or</u> make a pattern
- Shortcomings: cumbersome & verbose

```
P(moe) and P(larry) and P(curly) and …
```


15

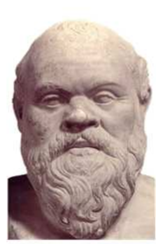
Limitations of Propositional Logic

- While propositional logic, which we covered, can express a great deal of complex logical expressions, it ultimately is <u>insufficient</u> for all arguments
- Why? The premises (and conclusions) in propositional logic have <u>no</u> internal structure.

16

Limitations of Propositional Logic

- For example, it cannot show the validity of Socrates Argument
- This arguments states: *"All humans are mortal. Socrates is a human. Therefore, he is mortal."*



17

Socrates Argument

- The following is the propositional logic form of the Socrates Argument
- Can we prove the conclusion?

```
All humans are mortal
Socrates is a human
________________________
Therefore, Socrates is mortal
```


18

The Socrates Argument

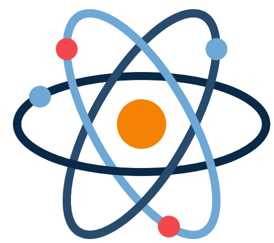
- The following is the argument in normal form
- A problem arises since the validity of this argument comes from the internal structure which propositional logic cannot "see"

```
H
S
—
M
```


19

Solution

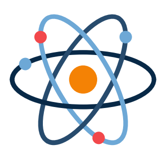
- It's time to break apart the logic and see the internal structure
- So, ***we are splitting the atom!***



20

Why Go Nuclear?

1. Expose the internal structure of those "atomic" sentences
2. Create new terminology to describe the semantics
3. Introduce laws to use and manage them



21

New Notation: For All

- The "For-All" symbol states every element *x* in the universe makes P(*x*) true
- So, it is true if and only if the every element *x* in the universe has P as true

$\forall_x \; P(x)$

22

New Notation: Exists

- The "Exists" States at least one element *x* in the universe makes P(*x*) true
- True if just a single P is true

$\exists_x \; P(x)$

23

Example: Pineapple Pizza

- Let's create the quantified statement for *"Someone doesn't like pineapple pizza!"*
- Let's create a predicate P(x) means *"x likes pineapple pizza"*
- What does someone mean? At least one person?

24

Pineapple Pizza: Try #1

- How about the following expression?
- It's not true if at least one person likes pineapple
- This means *"nobody likes pineapple pizza"*

$\neg (\exists_x P(x))$

25

Pineapple Pizza: Try #2

- We can also negate the predicate
- Means, for at least one person, they dislike pineapple pizza
- *"Someone doesn't like pineapple pizza!"*

$\exists_x \neg P(x)$

26

Quantifier Equivalence

Quantifier Conversion

27

Equivalence

- Just like propositional logic, quantitative expressions have equivalencies
- They follow the same basic logic we have seen before

28

Example: Opposite Expression

- Example: "Everyone in the room is awake"
- Let's create the reverse of this expression *(that still says the same thing)*

"Everyone in the room is awake."

29

Example: Opposite Expression

- So, let's just negate the predicate "is awake" into "is asleep"
- Does that work? No.

"Everyone in the room is asleep."

30

Example: Opposite Expression

- Now, let's negate the quantifier "everyone'" (for-all) into "someone" (exists)
- The expression below works: almost

"Someone in the room is asleep."

31

Example: Opposite Expression

- Well, what if we change the quantifier "everyone'" (for-all) into "someone" (exists)
- The expression below works: yes

"There is no one in the room that is asleep."

32

Exists and For-All

- The previous two laws allows us to extrapolate two additional laws
- Note: we simply push the negative and remove the double-negation

$$\forall_x \ P(x) \equiv \neg \exists_x \neg P(x)$$

$$\exists_x \ P(x) \equiv \neg \forall_x \neg P(x)$$

33

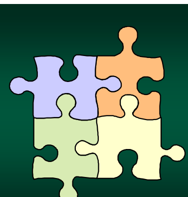
Equivalence – Moving Negation

- You can push a negation through a quantifier by toggling the quantifier
- Read the expressions below carefully…

$$\neg \exists_x \ P(x) \equiv \forall_x \neg P(x)$$

$$\neg \forall_x \ P(x) \equiv \exists_x \neg P(x)$$

34



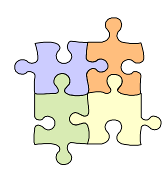
Conjunction & Disjunction

Breaking Apart and Combining

35

Conjunction & Disjunction

- Both the Exists and For-All quantifiers can be broken apart (and combined)
- This can occur if the expression contains an AND or an OR



36

Exists Disjunction

- If the Exists quantifier is used on a disjunction, it can be broken into two Exists
- This only works with $\lor$

$$\exists_x (P(x) \lor Q(x)) \equiv \exists_x P(x) \lor \exists_x Q(x)$$

For-All Conjunction

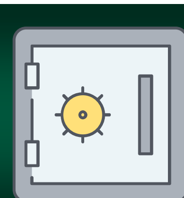
- If the For-All is used on a conjunction (and), it can be broken into two For-All
- This only works with $\land$

$$\forall_x (P(x) \land Q(x)) \equiv \forall_x P(x) \land \forall_x Q(x)$$

Please Wait

CSC 28
will begin shortly

(open the chat window)

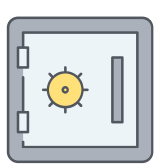


Bound & Free Variables

Some variables are not variable

Bound & Free Variables

- Not all variables used in a quantified expression is treated the same
- Each variable in an expression is either considered "bound" or "free"



Bound & Free Variables

- A variable is *free* if a value must be supplied to it before expression can be evaluated
- A variable is *bound* if it not free (usually a dummy variable) and contains values that are not needed to be evaluated

Example 1

- Which variables need we supply a value before the expression can be evaluated?
- Both x and c
- Without knowing both we cannot evaluate the expression (both are free)

```
(x ^ 2 < 4 * c)
```


43

Example 2

- Which variables need to be supplied before the expression can be evaluated?
- x: no, it is a dummy variable
- c: yes, once we give a value for c, we can evaluate the expression

$\forall_x$ (x ^ 2 < 4 * c)

44

Multiple Quantifiers

Many E's and A's doing headstands

45

Multiple Quantifiers

- A quantified statement may have more than one quantifier
- In fact, most of the time, statements will contain several

46

Example

- x > y is an expression with two variables
- The expression is true if an x is supplied which is greater than y

$\forall_x \exists_y$ P(x, y)

47

Example

- ∃y (x > y) is an expression with one free variable
- Evaluates to true if x is supplied and there is a y greater than the supplied x

$\forall_x \exists_y$ P(x, y)

48

Example

- ∀x ∃y (x > y) contains no free variables
- Evaluates to true if ∃y (x > y) is true for every x in the universe.

```
∀x ∃y P(x, y)
```


49

Example 2

- The following in an implication with two quantifiers as operands
- It states that whenever "∀x P(x)" is a true statement, then so is "∀x Q(x)".

```
∀x P(x) → ∀x Q(x)
```


50

Converting English to Logic

Do it bit by bit

51

Difficult Example

- Let's create a quantified statement for the following logical statement
- *We will go slowly, since this is not easy*

Everyone who has a friend who has Covid will have to be quarantined

52

Difficult Example

- "Everyone" is a For-All relationship
- What is everyone referring to? People
- So, the abstract object is a person

```
∀x (if x has a friend with Covid,
   then x must be quarantined)
```


53

Difficult Example

- So, we can factor it out into the expression below – x is a person
- *Now*, let's look at the sub expression…

```
∀x (if x has a friend with Covid,
   then x must be quarantined)
```


54

Difficult Example

- The sentence *" if x has a friend with Covid, x must be quarantined"* is an implication!
- Let's look at the antecedent (hypothesis)

```
if x has a friend with Covid,
   then x must be quarantined
```


55

Difficult Example

- How do we write the concept: *"x has **a** friend with Covid"* ?
- They just need a single friend
- So, this is an Exists quantifier

$\exists_y$ (x is friends with y, and y has Covid)

56

Difficult Example

- Now that we have a basic form of the final version, let's make some predicates
- We will use single letter names for brevity

```
F(x, y) means "x and y are friends"
C(x)    means "x has Covid"
Q(x)    means "x must be quarantined"
```


57

Difficult Example

- This says: "There exists a *person y* where *y* is friends with *person x*, and *y* has Covid"
- Note: *x* is not bound in this expression

$$\exists_y \ (F(x, y) \land C(y))$$

58

Difficult Example

- So, what happens if a friend has Covid?
- Then, they must be quarantined
- Note: implication is outside the exists

$$\exists_y (F(x, y) \land C(y)) \rightarrow Q(x)$$

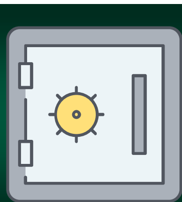
59

Difficult Example

- Now we can put it all together…
- The following is the quantified expression for our original statement

$$\forall_x (\exists_y (F(x, y) \land C(y)) \rightarrow Q(x))$$

60



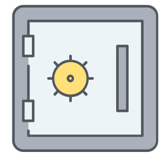
Bounded Quantifiers

Hidden Implication
(for those who hate to type)

61

Bound Quantifiers

- Some quantifiers can be more than meets the eye
- For brevity, many predicate and propositional expresses are merged with the ∀ and ∃



62

Shorthand Notation

- The following type of expression is quite common
- So much so that a shortcut notation is often employed

$$\forall_{x}\ (R(x) \rightarrow P(x))$$

63

Shorthand Notation

- The membership sub-expression is moved to the quantifier's subscript
- This is equivalent to the last

$$\forall_{R(x)} P(x)$$

64

Likewise…

- The sub-expression before the implication can be anything
- In this example, x > 5 is moved to the subscript

$$\forall_{x}\ (x > 5 \rightarrow P(x)) \equiv \forall_{x > 5}\ P(x)$$

65