Mar 1-3 Slides

Elizabeth Orrico

Good to have in your back pocket:
$p \to q$

"if p, then q"
"if p, q"
"p is sufficient for q"
"q if p"
"q when p"
"a necessary condition for p is q"
"q unless $\neg p$"

"p implies q"
"p only if q"
"a sufficient condition for q is p"
"q whenever p"
"q is necessary for p"
"q follows from p"
"q provided that p"

Predicates and First-Order Logic

We can only do so much with atomic propositions. To say more interesting things, like:

- *All files that are larger than 1,000 blocks are to be moved to backup provided that they have not been referenced within the last 100 days and that they are not system files.*

We need more.

Predicates and First-Order Logic

We can only do so much with atomic propositions. To say more interesting things, like:

- *All files that are larger than 1,000 blocks are to be moved to backup provided that they have not been referenced within the last 100 days and that they are not system files.*
- *All system files that are either source code or have not been referenced with the last 300 days are to be moved to backup.*
- *Any user file that is larger than 10,000 blocks is to be compressed irrespective of the most recent reference date.*

Predicates

[FOL stands for **F**irst **O**rder **L**ogic]

https://www.youtube.com/watch?v=r3wQM4vQUTc&feature=emb_logo

Predicates

“A function that evaluates to True or False”

“A proposition missing the noun(s)”

“A proposition template”

Predicates Example

Determine the predicate and the arguments of the following:

“Sam loves Diane”

Predicates Example

________ loves ________ = *L(x, y)*

x ***y***

“Sam loves Diane”

Formalizes to

L(Sam, Diane)

Predicates Example

________ loves ________ = $L(x, y)$

x y

"Sam loves Diane" = L(Sam, Diane)

"Diane doesn't love Sam" = ????

Predicates Example

________ loves _______ = $L(x, y)$ $H(x)$....

x y

"Sam loves Diane" = L(Sam, Diane)

"Diane doesn't love Sam" = ¬L(Diane, Sam)

"I Love Lucy" = ????

Predicates Example

________ loves ________ = $L(x, y)$

x y

"Sam loves Diane" = L(Sam, Diane)

"Diane doesn't love Sam" = ¬L(Diane, Sam)

"I Love Lucy" = L(me, Lucy)

"Everyone Loves Raymond" = ????

Predicates Example

________ loves ________ = $L(x, y)$

x y

$\forall$ = "for all"

Domain: people

"Sam loves Diane" = L(Sam, Diane)

"Diane doesn't love Sam" = ¬L(Diane, Sam)

"I Love Lucy" = L(me, Lucy)

"Everybody Loves Raymond" = $\forall$x L(x, Raymond)

Predicates Example

No predicates in predicates

No T/F in arguments

Universal Quantifier (∀)

∀ = "for all" or "given any"

It expresses that a **propositional function** can be satisfied by **every member of the domain**

Domain: People   L(x, y) = *x loves y*

∀x L(x, Raymond) means ???

Universal Quantifier (∀)

∀ = "for all" or "given any"

It expresses that a **propositional function** can be satisfied by **every member of the domain.**

Domain: People L(x, y) = *x loves y*

∀x L(x, Raymond) means "For all people x, each one loves Raymond"
"Given any person x, that person loves Raymond"
"Every person loves Raymond"

Predicates Example

________ loves ________ = $L(x, y)$ **$\forall$ = "for all"**

x   ***y***

Domain: people

"Everybody Loves Raymond" = $\forall x$ L(x, Raymond)

"Everybody does not love Chris" = ????

Predicates Example

Domain: People   L(x, y) = *x loves y*

“Everybody does not love Chris”

How could I rephrase this?

Predicates Example

Domain: People   L(x, y) = *x loves y*

“Everybody does not love Chris”

How could I rephrase this?

“For all people, each one does not love Chris”

“There does not exist one person who loves Chris”

Predicates Example

Domain: People   L(x, y) = *x loves y*

"Everybody does not love Chris"

How could I formalize this?

"For all people, each one does not love Chris"

$$\forall x \neg L(x, \text{Chris})$$

$$\neg \forall x \, L(x, \text{Chris}) = ???$$

Predicates Example

Domain: People   L(x, y) = *x loves y*

“Everybody does not love Chris”

How could I formalize this?

“For all people, each one does not love Chris”

$$\forall x\ \neg L(x, \text{Chris})$$

Predicates Example

Domain: People   L(x, y) = *x loves y*   ∃ = "there exists"

"Everybody does not love Chris"

How could I formalize this?

"There does not exist one person who loves Chris"

Existential Quantifier (∃)

∃ = "there exists", "there is at least one", or "for some"
It expresses that a **propositional function** can be satisfied by **at least one member of the domain.**

Domain: People   L(x, y) = *x loves y*

¬∃x L(x, Chris) means “There does not exist one person who loves Chris”

Existential Quantifier (∃)

∃ = "there exists", "there is at least one", or "for some"
It expresses that a **propositional function** can be satisfied by **at least one member of the domain.**

Domain: People L(x, y) = *x loves y*

¬∃x L(x, Chris) means “There does not exist one person who loves Chris”

(also see ∄)

∃ and ∀

Domain: People   L(x, y) = *x loves y*

¬∃x L(x, Chris) means "There does not exist one person who loves Chris"
∀x ¬L(x, Chris) means "For all people, each one does not love Chris"

$$\neg \exists x\, L(x, \text{Chris}) \equiv \forall x\, \neg L(x, \text{Chris})$$

∃ and ∀

Domain: People L(x, y) = *x loves y*

¬∃x L(x, Chris) means “There does not exist one person who loves Chris”
∀x ¬L(x, Chris) means “For all people, each one does not love Chris”

$$\neg \exists x\, L(x, \text{Chris}) \equiv \forall x\, \neg L(x, \text{Chris})$$

$$\not\equiv \left[\ \exists x\, \neg L(x, \text{Chris}) \equiv \neg \forall x\, L(x, \text{Chris})\ \right]$$

Another Example

Is the logical expression

$$\forall \mathbf{x}.\ \mathbf{Q(x)}$$

true or false

with

$$Q(x) = (x^2 \geq x)$$

Another Example

Is the logical expression

$$\forall \mathbf{x}.\ \mathbf{Q}(\mathbf{x})$$

true or false

with

$$Q(x) = (x^2 \geq x)$$

Q(4) = ???

Another Example

Is the logical expression

$$\forall \mathbf{x}.\ \mathbf{Q(x)}$$

true or false

with

$$Q(x) = (x^2 \geq x)$$

Q(4) = true

Q(0.5) = ???

Another Example

Is the logical expression true or false?

$$\forall x \in \mathbb{Z} \,.\, Q(x) \qquad \text{vs} \qquad \forall x \in \mathbb{R} \,.\, Q(x)$$

with

$$Q(x) = (x^2 \geq x)$$

∃ and ∀

Associate "for all" with AND's since it becomes false if just one truth value is false

Associate "there exists" with OR's since it becomes true if just one truth value is true

What about more than 1 quantifier:

Domain: People $L(x, y)$ = *x loves y*

Are these equivalent?

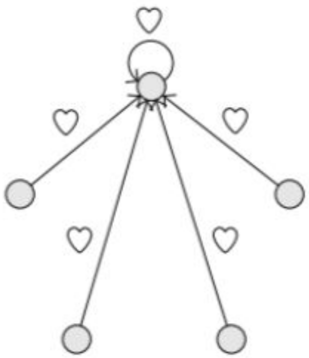
$$\exists y \forall x \, L(x,y) \overset{?}{\equiv} \forall x \exists y \, L(x,y)$$

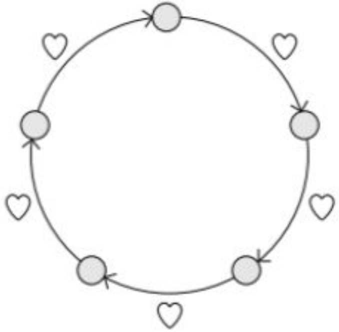
Quick Intro to Multiple Quantifiers:

Domain: People L(x, y) = *x loves y*

Are these equivalent?

$\exists y \forall x\, L(x,y)$ **is not equivalent to** $\forall x \exists y\, L(x,y)$





Quick Intro to Multiple Quantifiers:

Domain: People   L(x, y) = *x loves y*

Are these equivalent?

$\exists y \forall x\ L(x,y)$ **is not equivalent to** $\forall x \exists y\ L(x,y)$

Remember

When you are dealing with mixed quantifiers, the order is very important. $\forall x \exists y\, R(x, y)$ is not logically equivalent to $\exists y \forall x\, R(x, y)$.

Think about nested loops

Domain: {Ann, Bob, Chris} $\exists y \forall x\ L(x,y)$

```
// since ∃ means stuff “or’d” together, start with false
existValue = False
for y in {Ann, Bob, Chris}:
    // since ∀ means stuff “and’d” together, start with true
    univValue = True
    for x in {Ann, Bob, Chris}:
        univValue = univValue ∧ L(x,y)
    end
    existValue = existValue ∨ univValue
end
Return existValue
```

Think about nested loops

Domain: {Ann, Bob, Chris} ∃y∀x L(x,y)

How will this code change for "∀x∃y L(x,y)"?

```
// since ∃ means stuff "or'd" together, start with false
existValue = False
for y in {Ann, Bob, Chris}:
    // since ∀ means stuff "and'd" together, start with true
    univValue = True
    for x in {Ann, Bob, Chris}:
        univValue = univValue ∧ L(x,y)
    end
    existValue = existValue ∨ univValue
end
Return existValue
```

Think about nested loops

Domain: {Ann, Bob, Chris} $\forall x \exists y\, L(x,y)$

```
// since ∀ means stuff "and'd" together, start with true
univValue = True
for x in {Ann, Bob, Chris}:
    // since ∃ means stuff "or'd" together, start with false
    existValue = False
    for y in {Ann, Bob, Chris}:
        existValue = existValue ∨ L(x,y)
    end
    univValue = existValue ∧ univValue
end
Return univValue
```

Think about boolean logic

Domain: {Ann, Bob, Chris} $\exists y \forall x\, L(x,y)$

$$
\begin{aligned}
& (\ L(Ann, Ann) \land L(Bob, Ann) \land L(Chris, Ann)\ ) \\
\lor\ & (\ L(Ann, Bob) \land L(Bob, Bob) \land L(Chris, Bob)\ ) \\
\lor\ & (\ L(Ann, Chris) \land L(Bob, Chris) \land L(Chris, Chris)\ )
\end{aligned}
$$

Think about boolean logic

Domain: {Ann, Bob, Chris} $\exists y \forall x\, L(x,y)$

How will this change for "$\forall x \exists y\, L(x,y)$"?

$$
\begin{aligned}
& (\ L(Ann, Ann) \land L(Bob, Ann) \land L(Chris, Ann)\ ) \\
\lor\ & (\ L(Ann, Bob) \land L(Bob, Bob) \land L(Chris, Bob)\ ) \\
\lor\ & (\ L(Ann, Chris) \land L(Bob, Chris) \land L(Chris, Chris)\ )
\end{aligned}
$$

Think about boolean logic

Domain: {Ann, Bob, Chris} $\exists y.\ \forall x.\ L(x,y) = \exists y(\forall x\ (L(x,y)))$

How will this change for "$\forall x\ \exists y\ L(x,y)$"?

$$
\begin{aligned}
& (\ L(Ann, Ann) \lor L(Ann, Bob) \lor L(Ann, Chris)\ ) \\
\land\ & (\ L(Bob, Ann) \lor L(Bob, Bob) \lor L(Bob, Chris)\ ) \\
\land\ & (\ L(Chris, Ann) \lor L(Chris, Bob) \lor L(Chris, Chris)\ )
\end{aligned}
$$

https://www.cs.virginia.edu/luther/2102/S2021/eng2quant.html