PY1003
Intro to Logic

Lecture 11

Summary from the last lecture:

- **Introduced the universal quantifier $\forall x$**
- **Examples involving the quantifier and other logical connectives**
- **Interrelationship between the *universal* and the *existential* quantifier**
- **Domains of quantification**

Structure of today's lecture:

I. **Recap of important terminology**
II. **The (instantiation) tree-rule for the existential quantifier**
III. **Examples using trees and the existential quantifier**
IV. **The (instantiation) tree-rule for the universal quantifier**
V. **Examples using trees and the universal quantifier**
VI. **Trees for ¬∃ and ¬∀**

1. Important Terminology

Predicates:

Px: x is rich

We use capital letters P,Q,R,S to abbreviate predicates.

Names:

a: David

We use small letters a,b,c for names. Sometimes they are also called *individual constants* or *constants*.

Atomic sentence:

Pa, Qb, Rt

A predicate combined with a name is an *atomic sentence.*

Terminology (cont)

Existential quantifier:

∃x

is always used in combination with a *variable* and a *predicate*: ∃x Px

***Universal quantifier*:**

∀x

is always used in combination with a *variable* and a *predicate*: ∀xQx

***Quantified Sentence*:**

∀xQx, ∃x Px

Variable

x,y,z

in contrast to *names* (*constants*), variables don't stand for specific objects. They are *placeholders* for a name or stand for an arbitrary object.

Bound variables – free variables

Variables can (and usually do) occur *bound* by a certain quantifier:

$\forall x Qx$, $\exists x\ Px$

Or they can occur *free*:

Fx,

$\forall x\ (Qy \rightarrow Px)$ (y is free, x is bound)

i.e. *not bound* by a quantifier.

A sentence involving a *free variable* is a so-called *open sentence*.

$\exists x\ (Px \wedge Qx)$ quantified sentence

$Px \wedge Qx$ corresponding *open* sentence

***Domain of discourse* or *domain of quantification*:**

The entities over which the quantifiers range.

2. The tree-rule for the existential quantifier

- **We will adopt all tree-rules from sentential logic**
  - **Trees for ∧, ∨, →, ↔, ¬**

- **Introduce two new tree-rules for ∃, ∀**

Motivation:

When we say that there is someone who is rich, then we think that there is a certain object that is rich and we should be able to give it a name.

$$\exists xRx \quad \surd$$
$$|$$
$$R\tau$$

for some new constant τ *not* occurring already on the branch.

Consider the following argument:

(1) David is rich

(2) There is someone who is both rich and happy

(3) David is happy

This argument is NOT valid.

Ra

∃ x (Rx ∧ Hx)

¬ Ha

|

Ra ∧ Ha

|

Ha

----- (closed)

The mistake was that a constant was used that *already* appeared on the branch. The correct tree is:

Ra

$\exists x (Rx \land Hx)$ √

$\neg Ha$

|

$Rb \land Hb$

|

Hb

|

Rb

The reason why we need a new constant is that otherwise we might attribute certain properties to objects that don’t have them.

3. Examples using trees and existential quantifier

(1) There is someone such that if she/he is happy, then she is rich.

(2) Dave is happy

(3) Therefore Dave is rich

```
   ∃x (Hx -> Rx)
        Ha
       ¬ Ra
         |
     Hb -> Rb
      /      \
   ¬ Hb       Rb
```

Again, if the additional constraint that a *new* constant were *not* in place, then this clearly invalid argument would be valid.

4. The (instantiation) tree-rule for the universal quantifier

If ∀xGx occurs in the branch, then we require that 'G' is true of everything in the domain of quantification.

Therefore, for *every* name τ we already have in the branch, we must write 'G(τ)' in the branch.

If no term yet occur in the branch, then we can enter 'G(a)' in the branch.

Fa

Fb

Fc

∀xGx

|

Ga

Gb

Gc

Can we now tick off? – No

The reason why we don't tick off is that there might be other formula, occurring later that introduce new terms of which ∀x Ax will be true off.

Everyone is mortal

Therefore, Dave is mortal

∀xMx

¬ Ma

|

Ma

The argument is valid.

Is ∀xFx -> Fb a tautology?

¬ (∀x Fx -> Fb)

|

∀xFx

¬ Fb

|

Fb

5. Examples using trees and the universal quantifier

George is a banker (Ba)

No banker is generous (∀x (Bx -> ¬ Gx))

Therefore, George is not generous. (¬ Ga)

Ba

∀x (Bx -> ¬ Gx)

¬ ¬ Ga

Examples (contd)

All men are mortal (∀x (Px -> Mx))
All mortals fear dying (∀x (Mx -> Fx))
Dave is a men
Therefore, Dave fears dying

$$\forall x\ (Px \rightarrow Mx)$$
$$\forall x\ (Mx \rightarrow Fx)$$
$$Pa$$
$$\neg Fa$$

6. Trees for ¬ ∃ and ¬ ∀

Remember at the last lecture we arrived at the following relationship between *existential* and *universal* quantifier.

<u>Quantifier interdefinability:</u>

$\exists x\ Px$ iff $\neg\ \forall x\ \neg\ Px$

$\forall x\ Px$ iff $\neg\ \exists x\ \neg\ Px$

It is not the case that someone is happy

$\neg\ \exists x\ Hx$

Everyone is unhappy

$\forall x\ \neg\ Hx$

they are logically equivalent!

➔ From this we can motivate the tree-rule for ¬ ∃

Tree-rule for ¬ ∃

$$\neg \exists x Fx \quad \surd$$

$$|$$

$$\forall x \neg Fx$$

The same idea applies to $\neg \forall x Fx$

It is not the case that everyone is happy

$\neg \forall x Hx$

There is someone who is not happy

$\exists x \neg Hx$

So, the tree rule will be

Tree-rule for ¬ ∀x

$$\neg \forall x\, Fx \ \surd$$

|

$$\exists x \neg Hx$$

Some examples

There is someone who is rich
Therefore there is someone who is rich or happy

$$\exists x\, Rx$$

$$\neg(\exists x(Rx \vee Hx))$$

All men are mortal (∀x (Px -> Mx))

All mortals fear dying (∀x (Mx -> Fx))

Therefore, all men fear dying (∀x (Px -> Fx))

$$\forall x\,(Px \rightarrow Mx)$$

$$\forall x\,(Mx \rightarrow Fx)$$

$$\neg(\forall x\,(Px \rightarrow Fx))$$