**Ch 11 - Multiple Quantifiers**

* In general, to say that **every pair of distinct objects stands in some relation,** use a sentence of the form  
    
  ∀x **∀**y ((x ≠ y) ***→ …***)
* To say that **there are two objects with a certain property,** use a sentence of the form  
    
  ∃x ∃y (x ≠ y &
* When evaluating a sentence with multiple quantifiers, don’t fall into the trap of thinking that distinct variables range over distinct objects
* When we have a sentence with a string of mixed quantifiers, the order of the quantifiers makes a difference

**Weak/Strong Reading**

*Every minute a man is mugged in NYC.*

* **Weak**
* ∀x (Minute(x) ***→*** ∃y (Man(y) & MuggedDuring(y,x)))
* For every object x, if x is a minute then there exists an object y such that y is a man and the man is mugged during minute x.
* **Strong**
* ∃y (Man(y) & ∀x (Minute(x) ***→*** *MuggedDuring(y,x*))
* There is an object y such that y is a man and for every minute x, the man is mugged during that minute.

**Translations Using Function Symbols**

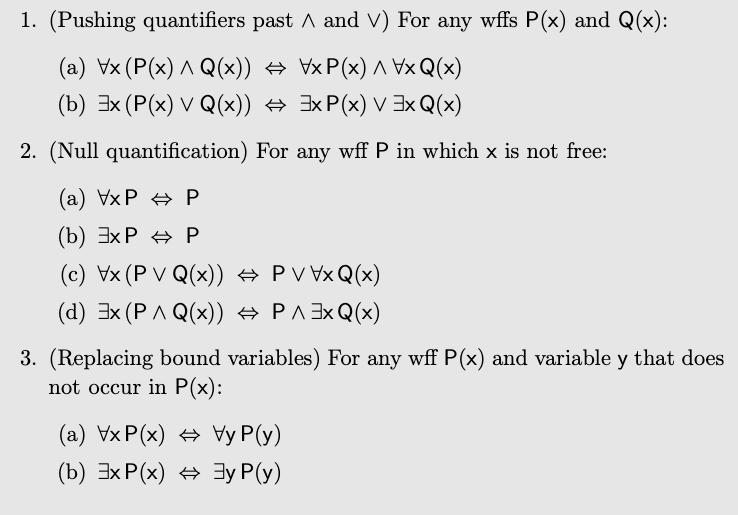
* anything we can say with the function symbol we can say in some other way using the relation symbol
  + **with function symbol:** mother(max) = nancy
  + **with relation symbol:** MotherOf(nancy, max)
* however, the sentence using relation symbols can be quite a bit more complicated
* **example**
  + ∀x OlderThan(mother(x), x)
  + ∀x ∃y [MotherOf(y,x) & OlderThan(y,x) & ∀z[MotherOf(z,x) ***→*** y = z]]
* anything you can express using an n-ary function symbol can also be expressed using an n+1-ary relation symbol, plus identity predicate, but at a cost in terms of the complexity of the sentences used

**Prenex Form**

* quantifiers are out in front, connectives are in the back
* prenex form: all quantifiers come first
  + ie, a wff is in prenex form it either it contains no quantifiers at all, or else is of the form  
      
    Q1 v1 Q2 v2 … Qn vn P  
      
    where Qi is either ∀ or ∃, each vi is some variable, and the wff P is quantifier-free
* advantages of prenex form
  + gives you nice measure of the logical complexity of the sentences
  + what turns out to matter is not so much the number of quantifiers, as the number of times you flip from one type of quantifier to the other
  + the more of these so-called **alternations,** the more complex the sentence is, logically speaking
* turns out that every sentence is logically equivalent to one (in fact many) in prenex form

**Basic Strategy for Getting Formula Into Prenex**

* we have the logical equivalences that are needed



* get rid of conditionals in favor of Boolean connectives, since these interact more easily with quantifiers in our principles
* work from the inside out

**example** ∀⟺ ∃*→*

∃x P(x) *→* ∃y Q(y)

write the conditional in a form using disjunction

~∃x P(x) *|* ∃y Q(y)

DeMorgan’s Law for Quantifier

∀x ~P(x) | ∃y Q(y)

Null quantification principle

∃y [∀x ~P(x) | Q(y)]

∃y ∀x [~P(x) | Q(y)]

**example**

**(**∃x P(x) | R(b)) *→* ∀x (P(x) & ∀x Q(x))

neither the antecedent nor the consequent of the conditional is in prenex form

Null quantification on the antecedent

∃x **(**P(x) | R(b)) *→* ∀x (P(x) & ∀x Q(x))

Distribution of ∀ and & on the consequent

∃x **(**P(x) | R(b)) *→* (∀x P(x) & ∀x Q(x))

∃x **(**P(x) | R(b)) *→* ∀x (P(x) & Q(x))

Get rid of conditional in favor of Boolean connectives

~∃x **(**P(x) | R(b)) *|* ∀x (P(x) & Q(x))

DeMorgan’s Law

∀x ~**(**P(x) | R(b)) *|* ∀x (P(x) & Q(x))

Replace one of the variable x with a new variable

∀x ~**(**P(x) | R(b)) *|* ∀z (P(z) & Q(z))

Null quantification, twice

∀x ∀z (~**(**P(x) | R(b)) *|* (P(z) & Q(z)))