Retinem ca orice enunt are valoarea de adevar adevarat sau fals, adevarat=/=fals, iar conectorii logici sunt definiti in modul obisnuit pentru enunturi cu una dintre aceste valori de adevar.

Consideram *non* mai prioritar decat *sau, si*, =>, ⇔, *xor*.

Retinem din cursul anterior ca, pentru orice enunturi p si q:

(p ⇔ q) ⇔ [(p=>q) si (q=>p)] ⇔ p si q au aceeasi valoare de adevar.

Am demonstrat ca: (p=>q) => (non q => non p). (*)

In implicatia de mai sus inlocuim:

p cu non q

si

q cu non p,

apoi aplicam faptul ca: p⇔non non p si la fel pentru q, apoi substitutivitatea pentru valorile booleene:

(non q => non p) => (non non p => non non q) ⇔ (p => q), asadar:

(non q => non p) => (p => q). (**)

(*),(**) => (p=>q) <=> (non q => non p). (1)

Interschimband p cu q in echivalenta de mai sus, obtinem:

(q=>p) <=> (non p => non q). (2)

Aplicam (1),(2), din nou substitutivitatea, apoi comutativitatea conjunctiei:

(p<=>q) ⇔ [(p=>q) si (q=>p)] ⇔ [(non q => non p) si (non p => non q)] ⇔ [(non p => non q) si (non q => non p)] ⇔ (non p <=> non q).

Pentru orice x (subinteles: arbitrar, fixat) si orice multimi A,B:

x ϵ AUB ⇔ (x ϵ A sau x ϵ B)

x ϵ A∩B ⇔ (x ϵ A si x ϵ B)

x ϵ A\B ⇔ (x ϵ A si non(x ϵ B))

x ϵ AΔB ⇔ [(x ϵ A si non(x ϵ B)) sau (x ϵ B si non(x ϵ A))] ⇔ (x ϵ A xor x ϵ B)

Pentru orice multimi A,B:

A=B ⇔ (∀x)(x ϵ A <=> x ϵ B)

A⊆B ⇔ (∀x)(x ϵ A => x ϵ B)

A⊂B ⇔ (A⊆B si non(A=B))

Cu definitiile de mai sus, proprietatile din calculul cu multimi se transcriu in proprietati ale conectorilor logici (si cuantificatorilor), care pot fi demonstrate semantic (i.e. prin calcul cu valori de adevar) in Prolog.