A→B -AvB

-(AvB) -A^-B

-(A^B) -Av-B

-Esiste x Per ogni x -x

-Per ogni x Esiste x -x

Frase 1: Esiste un cane nero.

* #1: ∃x nero(x)

Valida, ma quando il dominio è formato da solo cani.

* #2: ∃x cane(x) ∧ nero(x)

Ok, nel dominio non vi sono solo cani.

* #3: Ǝx cane(x) → nero(x)

È valida in ogni dominio. C’è un oggetto nero o c’è un oggetto che non è un cane.

* #4: cane(fido) ∧ nero(fido)

È la skolemizzazione della #3. In cui fido è la costante di Skolem.

Frase 2: Tutti i corvi sono neri.

* #1: ∀x nero(x)

Valida, ma quando il dominio è solo di corvi.

* #2: ∀x corvo(x) → nero(x)

Per domini che hanno anche altri oggetti che non sono corvi.

* #3: ∀x corvo(x) ∧ nero(x)

No, indica che tutti gli oggetti del dominio sono corvi e tutti sono neri.

* #4: ∀x -corvo(x) ⋁ nero(x)

Trasformazione in clausole della #2 (la formula corretta)

Frase 3: Tutte le scimmie sono fuggite su un albero.

Predicati disponibili: scimmia(x), albero(x), fugge(x,y)

* #1: ∀x scimmia(x) → ∃y albero(y) ⋀ fugge(x,y)
* #2: ∀x scimmia(x) → albero(t(x)) Λ fugge(x, t(x))

Applicando la skolemizzazione alla #1.

Frase 4: Esiste una tartaruga che è più vecchia di ogni essere umano.

Predicati disponibili: tartaruga(x), umano(x), vecchio(x,y)

* #1: ∃x tartaruga(x) Λ (∀y umano(y) → vecchio(x,y)
* #2: ∀y tartaruga(t) ∧ (-umano(y) V vecchio(x,y))

Trasformazione in clausole della formula #1

Compito 13/9/18: Se il Segretario propone un membro del Consiglio Direttivo, allora il Presidente dell’Associazione lo nomina.

Predicati disponibili: presidente(e), segretario(x), propone(x,y), nomina(x,y)

Fatti: presidente(Giovanni), segretario(Mario), proposto(Mario, Luca), -nomina(Giovanni, Luca)

* #1: ∃x ∃z segretario(x) Λ presidente(z) Λ (∀y propone(x,y) → nomina(z,y))

Soluzione sviluppata a lezione.

* #2: ∀x ∀y ∀z [segretario(x) Λ presidente(z) Λ propone(x,y) → nomina(z,y)]

Soluzione proposta. Userò #2 per la trasformazione in clausole e la risoluzione.

* #3: -segretario(x) v -presidente(z) v -propone(x,y) v nomina(z,y)

Trasformazione in clausole della #2.

Applicazione della risoluzione. Nomino le clausole

* #1: presidente(Giovanni)
* #2: segretario(Mario)
* #3: propone(Mario, Luca)
* #4: -nomina(Giovanni, Luca)
* #5: -segretario(x) v -presidente(z) v -propone(x,y) v nomina(z,y)

Risoluzione:

* #6 = #5-#2: x/mario → -presidente(z) v -propone(mario,y) v nomina(z,y)
* #7 = #6-#1: z/giovanni → -propone(mario,y) v nomina(giovanni,y)
* #8 = #7-#4: y/luca → -propone(mario,luca)
* #9 = #8-#3 → []

Compito 14/6/18:

Frase 1: Mario sa risolvere gli stessi problemi di logica che sa risolvere Maria.

Frase 2: Maria sa risolvere gli stessi problemi di logica che sa risolvere Mario.

Frase 3: Chi sa risolvere un problema di logica difficile, prende 30 all’esame di IA.

Frase 4: Esiste un problema di logica difficile che Mario sa risolvere.

Dim: Maria prenderà 30 all’esame di IA.

Predicati disponibili: problema(x), difficile(x), prende30(x), risolve(x,y)

* #1: ∀x (problema(x) Λ risolve(maria,x)) → risolve(mario,x)
* #2: ∀x (problema(x) Λ risolve(mario,x)) → risolve(maria,x)
* #3: ∀x ∀y (problema(x) Λ difficile(x) Λ risolve(y,x)) → prende30(y)
* #4: Ǝx (problema(x) ⋀ difficile(x) Λ risolve(mario,x))

Trasformazione in clausole.

* #1: -problema(x) ⋁ -risolve(maria,x) V risolve(mario, x)
* #2: -problema(x) ⋁ -risolve(mario,x) V risolve(maria, x)
* #3: -problema(x) V -difficile(x) V -risolve(y,x) V prende30(y)
* #4a: problema(p)
* #4b: difficile(p)
* #4c: risolve(mario, p)
* #5: -prende30(maria)

Risoluzione:

* #6 = #5-#3: y/maria → -problema(x) V -difficile(x) V -risolve(maria,x)
* #7= #6-#4a: x/p → difficile(p) V -risolve(maria,p)
* #8 = #7-#4b → -risolve(maria,p)
* #9 = #8-#2: x/p → -problema(p) ⋁ -risolve(mario,p)
* #10 = #9-#4a → -risolve(mario,p)
* #11 = #10-#4c → []

Compito 9/2/18

Frase 1: Ogni opera di Mozart è un capolavoro o è incompiuta.

Frase 2: Il flauto magico è un’opera di Mozart e non è incompiuta.

Frase 3: Esiste almeno un’opera di Mozart che è un capolavoro.

Dim: Il flauto magico è un’opera di Mozart ed è un capolavoro.

Predicati disponibili: capolavoro(x), incompiuta(x), opera(x,y)

* #1: ∀x opera(x, mozart) → (inc(x) xor cap(x))
  + ∀x opera(x,mozart) → [(inc()x) Λ -cap(x) V (-inc(x) Λ cap(x))]
  + Trasformando l’xor.
* #2: opera(fl, mozart) Λ -inc(fl)
* #3: ∃x opera(x,mozart) → Λ cap(x)

cl([], \_).

cl([H|T], N) :- \+member(N, H), cl(T,N).

diff([], L):- L\==[].

diff([H|T], L):- \+member(H,L), diff(T,L).