Lifting di risoluzione e refutazione

- Nel proposizionale la **regola di inferenza di risoluzione** unita all'**algoritmo di refutazione** costituisce una procedura di inferenza completa
- Risoluzione e refutazione possono essere applicate anche a FOL:
  - La KB va tradotta in **CNF** (conjunctive normal form, cioè una congiunzione di clausole, ognuna delle quali è una disgiunzione di letterali)
  - Le variabili sono intese essere **quantificate universalmente**
  - Ogni KB FOL può essere tradotta in una KB CNF inferenzialmente equivalente. In particolare una formula **CNF** è **insoddisfacibile** solo quando l'originale **FOL** è **insoddisfacibile**, per questo è possibile applicare la procedura di refutazione

Tradurre una KB FOL in CNF

- La traduzione segue passi simili a quelli visti per il caso proposizionale (elimina biimplicazione, elimina implicazione, ecc.)
- La differenza è che bisogna gestire i quantificatori
- Vediamo un esempio. Supponiamo che la KB contenga la formula:

  $\forall x\ [\forall y\ Animale(y) \Rightarrow Ama(x, y)] \Rightarrow [\exists y\ Ama(y, x)]$
  Tutti coloro che amano gli animali sono amati da qualcuno

Tradurre una KB FOL in CNF

FOL: $\forall x\ [\forall y\ Animale(y) \Rightarrow Ama(x, y)] \Rightarrow [\exists y\ Ama(y, x)]$

Tutti coloro che amano gli animali sono amati da qualcuno

1) **Elimina l'implicazione:**
   $\forall x\ \neg\ [\forall y\ Animale(y) \Rightarrow Ama(x, y)] \vee [\exists y\ Ama(y, x)]$
   $\forall x\ \neg\ [\forall y\ \neg Animale(y) \vee Ama(x, y)] \vee [\exists y\ Ama(y, x)]$

2) **Sposta la negazione all'interno ($\neg \forall \equiv \exists \neg$):**
   $\forall x\ \ [\exists y\ Animale(y) \wedge \neg\ Ama(x, y)] \vee [\exists y\ Ama(y, x)]$

3) **Standardizzazione delle variabili in ($\exists y$ …) $\vee$ ($\exists y$ …):**
   $\forall x\ \ [\exists y\ Animale(y) \wedge \neg\ Ama(x, y)] \vee [\exists z\ Ama(z, x)]$

4) **Skolemizzazione (eliminazione degli esistenziali): …**

Tradurre una KB FOL in CNF

FOL: ∀x [∀y Animale(y) ⇒ Ama(x, y)] ⇒ [∃y Ama(y, x)]

Tutti coloro che amano gli animali sono amati da qualcuno

1) **Elimina l'implicazione:** …
2) **Sposta la negazione all'interno (¬ ∀ ≡ ∃¬): ...**
3) **Standardizzazione delle variabili in (∃y …) ∨ (∃y …): ...**
   ∀x [∃y Animale(y) ∧ ¬ Ama(x, y)] ∨ [∃z Ama(z, x)]
4) **Skolemizzazione (eliminazione degli esistenziali):**
   non possiamo applicare la regola EI perché la formula non segue il pattern
   ∃x F(x). Otterremmo ∀x [Animale(A) ∧ ¬ Ama(x, A)] ∨ [Ama(B, x)]
   che si legge: tutti amano uno specifico animale A oppure sono amati da un qualche di specifico B. A e B sono costanti e hanno lo stesso valore per tutti gli x!!

4) Skolemizzazione

- Vogliamo poter dire che animale e persona dipendono da x
- Sostituiamo ogni variabile quantificata esistenzialmente con una funzione che ha per argomenti tutte le variabili quantificate universalmente nel cui scope ricade:
  $\forall x1, x2, \ldots$ [$\exists y$ P(y, x1,...) ...$\exists z$ Q(z, x1,...)] diventa
  $\forall x1, x2, \ldots$ [P(**S1(x1, x2, ...), ...**) ... Q(**S2(x1, x2, ...), ...**)]
- **S1**, **S2** sono dette **funzioni di Skolem**
- Nel caso particolare in cui l'esistenziale non ricade nello scope di alcun universale tali funzioni diventano **costanti di Skolem** (**EI**)
- Nell'esempio otterremo quindi
  $\forall x$ [Animale(F(x)) $\wedge \neg$ Ama(x, F(x))] $\vee$ [Ama(G(x), x)]

Tradurre una KB FOL in CNF

5) **Cancella i quantificatori universali**
$\cancel{\forall x}$ [Animale(F(x)) ∧ ¬ Ama(x, F(x))] ∨ [Ama(G(x), x)]

6) **Distribuisci ∨ su ∧**
[Animale(F(x)) ∨ Ama(G(x), x)]∧ [¬ Ama(x, F(x))] ∨ [Ama(G(x), x)]

Il risultato non è più leggibile ma non è inteso essere usato da un essere umano. La traduzione è automatizzabile e le clausole servono al processo di inferenza

Binary resolution in FOL

$$\frac{l_1 \vee \ldots \vee l_k, \qquad m_1 \vee \ldots \vee m_n}{\text{SUBST}(\theta\, , l_1 \vee \ldots l_{i-1} \vee l_{i+1} \vee \ldots \vee l_k \vee m_1 \vee \ldots m_{j-1} \vee m_{j+1} \vee \ldots \vee m_n)}$$

- Per tutti i valori di i la sostituzione unifica l_i e $\neg m_j$
- Esempio: consideriamo Re(John) e ¬Re(x). Sono opposte e la sostituzione θ = {x/John} rende la prima equivalente al negato della seconda: Re(John)/θ ≡ ¬¬Re(x) /θ
- Le due clausole da risolvere **non condividono variabili**
- Occorre fare il **lifting della fattorizzazione**: due letterali sono ridotti ad uno non se sono uguali ma se sono unificabili. L'unificatore va applicato alle clausole intere
- **Binary resolution + fattorizzazione costituisce una regola di inferenza completa**

Dimostrazione per refutazione

- Grazie al lifting della resolution diventa possibile applicare l'inferenza per refutazione a FOL
- Vediamo un esempio:

  A) Tutti coloro che amano gli animali sono amati da qualcuno

  B) Tutti coloro che uccidono animali non sono amati da nessuno

  C) Jack ama tutti gli animali

  D) O Jack o Curiosity hanno ucciso il gatto, il cui nome è Tuna

  E) Curiosity ha ucciso il gatto?

Esempio

A) $\forall x\ [\forall y\ Animale(y) \Rightarrow Ama(x, y)] \Rightarrow [\exists y\ Ama(y, x)]$

B) $\forall x\ [\exists z\ Animale(z) \wedge Uccide(x, z)] \Rightarrow [\forall y\ \neg\ Ama(y, x)]$

C) $\forall x\ Animale(x) \Rightarrow Ama(Jack, x)$

D) $Uccide(Jack, Tuna) \vee Uccide(Curiosity, Tuna)$

E) $Gatto(Tuna)$

F) $\forall x\ Gatto(x) \Rightarrow Animale(x)$

Alle precedenti aggiungiamo il goal negato:

G) $\neg\ Uccide(Curiosity, Tuna)$

Esempio: da FOL a CNF

A1) Animale(F(x)) ∨ Ama(G(x), x)

A2) ¬ Ama(x, F(x)) ∨ Ama(G(x), x)

B) ¬ Ama(y, x) ∨¬ Animale(z) ∨¬ Uccide(x, z)

C) ¬ Animale(x) ∨Ama(Jack, x)

D) Uccide(Jack, Tuna) ∨ Uccide(Curiosity, Tuna)

E) Gatto(Tuna)

F) ¬ Gatto(x) ∨ Animale(x)

G) ¬ Uccide(Curiosity, Tuna)