E.B.1.4 (FOL: Aldo, inferenza)

```
𝒫 = {
  CaneDaCaccia/1, AbbaiaDiNotte/1, Gatto/1,
  HaIlTopoInCasa/1, HaSonnoLeggero/2, Possiede/2
}
ℱ = { Aldo/0 }
KB = {
  ∀ c CaneDaCaccia(c) → AbbaiaDiNotte(c),
  ∀ p, g Gatto(g) ∧ Possiede(p, g) → ¬HaIlTopoInCasa(p),
  ∀ p, a HaSonnoLeggero(p) ∧ Possiede(p, a) → ¬AbbaiaDiNotte(a),
  ∃ a Possiede(Aldo, a) ∧ (Gatto(a) ∨ CaneDaCaccia(a))
}
```

1.1 Inferenza

Si dimostri che KB ⊨ HaSonnoLeggero(Aldo) → ¬HaIlTopoInCasa(Aldo) *(primo tenativo fallimentare perché non mi ero accorto che ci fosse una clausola con più di un letterale positivo)*

α =
HaSonnoLeggero(Aldo) → ¬HaIlTopoInCasa(Aldo) =
{A → B ≡ ¬A ∨ B}
¬HaSonnoLeggero(Aldo) ∨ ¬HaIlTopoInCasa(Aldo)

HaSonnoLeggero(p) ∧ Possiede(p, a) → ¬AbbaiaDiNotte(a) =
{A → B ≡ ¬A ∨ B}
¬(HaSonnoLeggero(p) ∧ Possiede(p, a)) ∨ ¬AbbaiaDiNotte(a) =
{De Morgan}
¬HaSonnoLeggero(p) ∨ ¬Possiede(p, a) ∨ ¬AbbaiaDiNotte(a) =
{Commutatività di ∨}
¬Possiede(p, a) ∨ ¬AbbaiaDiNotte(a) ∨ ¬HaSonnoLeggero(p) =
{De Morgan al contrario}
¬(Possiede(p, a) ∧ AbbaiaDiNotte(a)) ∨ ¬HaSonnoLeggero(p) =
{A → B ≡ ¬A ∨ B}
Possiede(p, a) ∧ AbbaiaDiNotte(a) → ¬HaSonnoLeggero(p)

KB' = {
  CaneDaCaccia(c1) → AbbaiaDiNotte(c1),
  Gatto(g1) ∧ Possiede(p1, g1) → ¬HaIlTopoInCasa(p1),
  Possiede(p, a) ∧ AbbaiaDiNotte(a) → ¬HaSonnoLeggero(p)
  Possiede(Aldo, A_1),
  Gatto(A_1) ∨ CaneDaCaccia(A_1)
}

KB' **non** è in forma di **Horn** (a causa di $\texttt{Gatto}(A_1) \lor \texttt{CaneDaCaccia}(A_1)$), per cui non si può applicare l'algoritmo di concatenazione in avanti. Infatti non è possibile ottenere $\texttt{AbbaiaDiNotte}(A_1)$ che serve per ottenere $\neg\texttt{HaSonnoLeggero(Aldo)}$.

$$\frac{\textit{Possiede}(\texttt{Aldo}\ A_1),\ \texttt{AbbaiaDiNotte}(A_1) \qquad (\textit{Possiede}(\texttt{p, a}) \land \texttt{AbbaiaDiNotte(a)} \to \neg\texttt{HaSonnoLeggero(p)})}{\neg\texttt{HaSonnoLeggero(Aldo)}}$$

$$\frac{\textit{Possiede}(\texttt{Aldo}\ A_1) \qquad (\textit{Possiede}(\texttt{p1, g1}) \to \neg\texttt{HaIlTopoInCasa(p1)})}{\neg\texttt{HaIlTopoInCasa(Aldo)}}$$

$$\frac{\neg\texttt{HaSonnoLeggero(Aldo)},\ \neg\texttt{HaIlTopoInCasa(Aldo)} \qquad \texttt{A} \land \texttt{B} \to \texttt{A} \lor \texttt{B}}{\neg\texttt{HaSonnoLeggero(Aldo)} \lor \neg\texttt{HaIlTopoInCasa(Aldo)}}$$

1.1.1 CNF

$$\begin{gathered}\neg\alpha = \\ \neg(\neg\texttt{HaSonnoLeggero(Aldo)} \lor \neg\texttt{HaIlTopoInCasa(Aldo)}) = \\ \{\texttt{De Morgan}\} \\ \texttt{HaSonnoLeggero(Aldo)} \land \texttt{HaIlTopoInCasa(Aldo)}\end{gathered}$$

$\texttt{KB}_{\texttt{CNF}}$ = {
  $\neg\texttt{CaneDaCaccia(c1)} \lor \texttt{AbbaiaDiNotte(c1)}$,
  $\neg\texttt{Gatto(g1)} \lor \neg\textit{Possiede}(\texttt{p1, g1}) \lor \neg\texttt{HaIlTopoInCasa(p1)}$,
  $\neg\textit{Possiede}(\texttt{p, a}) \lor \neg\texttt{AbbaiaDiNotte(a)} \lor \neg\texttt{HaSonnoLeggero(p)}$,
  $\textit{Possiede}(\texttt{Aldo}, A_1)$,
  $\texttt{Gatto}(A_1) \lor \texttt{CaneDaCaccia}(A_1)$
}

Si usa l'algoritmo di risoluzione per dimostrare che $\texttt{KB}_{\texttt{CNF}} \land \neg\alpha$ non è soddisfacibile, quindi $\texttt{KB}_{\texttt{CNF}} \vDash \alpha$

1. $(\neg\texttt{Gatto(g1)} \lor \neg\textit{Possiede}(\texttt{p1, g1}) \lor \neg\texttt{HaIlTopoInCasa(p1)}) \land (\texttt{Gatto}(A_1) \lor \texttt{CaneDaCaccia}(A_1))$
   $\vDash (\neg\textit{Possiede}(\texttt{p1}, A_1) \lor \neg\texttt{HaIlTopoInCasa(p1)} \lor \texttt{CaneDaCaccia}(A_1))$
2. $(\neg\textit{Possiede}(\texttt{p1}, A_1) \lor \neg\texttt{HaIlTopoInCasa(p1)} \lor \texttt{CaneDaCaccia}(A_1)) \land (\textit{Possiede}(\texttt{Aldo}, A_1))$
   $\vDash (\neg\texttt{HaIlTopoInCasa(Aldo)} \lor \texttt{CaneDaCaccia}(A_1))$
3. $(\neg\texttt{HaIlTopoInCasa(Aldo)} \lor \texttt{CaneDaCaccia}(A_1)) \land (\neg\texttt{CaneDaCaccia(c1)} \lor \texttt{AbbaiaDiNotte(c1)})$
   $\vDash (\neg\texttt{HaIlTopoInCasa(Aldo)} \lor \texttt{AbbaiaDiNotte}(A_1))$
4. $(\neg\texttt{HaIlTopoInCasa(Aldo)} \lor \texttt{AbbaiaDiNotte}(A_1)) \land (\texttt{HaIlTopoInCasa(Aldo)})$
   $\vDash (\texttt{AbbaiaDiNotte}(A_1))$

5. (¬*Possiede*(p, a) ∨ ¬AbbaiaDiNotte(a) ∨ ¬HaSonnoLeggero(p)) ∧ (HaSonnoLeggero(Aldo))
   ⊨ (¬*Possiede*(p, a) ∨ ¬AbbaiaDiNotte(a))
6. (¬*Possiede*(p, a) ∨ ¬AbbaiaDiNotte(a)) ∧ (*Possiede*(Aldo, A_1))
   ⊨ (¬AbbaiaDiNotte(A_1))
7. ¬AbbaiaDiNotte(A_1) ∧ AbbaiaDiNotte(A_1) ⊨ ()

Avendo ottenuto la clausola vuota si ha che $\mathrm{KB}_{\mathrm{CNF}} \land \neg\alpha$ non è soddisfacibile, quindi $\mathrm{KB}_{\mathrm{CNF}} \vDash \alpha$

1.2 Prover9 / Mace4

```
formulas(sos).
    (all x (CaneDaCaccia(x) -> AbbaiaDiNotte(x))).
    (all x (all y (Gatto(y) & Possiede(x,y) -> -HaIlTopoInCasa(x)))).
    (all x (all y (HaSonnoLeggero(x) & Possiede(x,y) -> -AbbaiaDiNotte(y)))).
    (exists y (Possiede(aldo,y) & (Gatto(y) | CaneDaCaccia(y)))).
end_of_list.

formulas(goals).
    (HaSonnoLeggero(aldo) -> -HaIlTopoInCasa(aldo)).
end_of_list.
```

```
% Maximum clause weight is 0.000.
% Given clauses 0.

1 (all x (CaneDaCaccia(x) -> AbbaiaDiNotte(x))) # label(non_clause).  [assumption].
2 (all x all y (Gatto(y) & Possiede(x,y) -> -HaIlTopoInCasa(x))) # label(non_clause).  [assumption]
3 (all x all y (HaSonnoLeggero(x) & Possiede(x,y) -> -AbbaiaDiNotte(y))) # label(non_clause).  [assumption]
4 (exists y (Possiede(aldo,y) & (Gatto(y) | CaneDaCaccia(y)))) # label(non_clause).  [assumption]
5 HaSonnoLeggero(aldo) -> -HaIlTopoInCasa(aldo) # label(non_clause) # label(goal).  [goal].
6 Gatto(c1) | CaneDaCaccia(c1).  [clausify(4)].
7 -CaneDaCaccia(x) | AbbaiaDiNotte(x).  [clausify(1)].
8 Gatto(c1) | AbbaiaDiNotte(c1).  [resolve(6,b,7,a)].
9 -Gatto(x) | -Possiede(y,x) | -HaIlTopoInCasa(y).  [clausify(2)].
10 HaSonnoLeggero(aldo).  [deny(5)].
11 -HaSonnoLeggero(x) | -Possiede(x,y) | -AbbaiaDiNotte(y).  [clausify(3)].
12 AbbaiaDiNotte(c1) | -Possiede(x,c1) | -HaIlTopoInCasa(x).  [resolve(8,a,9,a)].
13 Possiede(aldo,c1).  [clausify(4)].
14 -Possiede(aldo,x) | -AbbaiaDiNotte(x).  [resolve(10,a,11,a)].
15 AbbaiaDiNotte(c1) | -HaIlTopoInCasa(aldo).  [resolve(12,b,13,a)].
16 HaIlTopoInCasa(aldo).  [deny(5)].
17 AbbaiaDiNotte(c1).  [resolve(15,b,16,a)].
18 -AbbaiaDiNotte(c1).  [resolve(14,a,13,a)].
19 $F.  [resolve(17,a,18,a)].
```