ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Logica: esempi

Federico Chesani

DISI

Department of Informatics – Science and Engineering

Disclaimer & Further Reading

- These slides are largely based on previous work by Prof. Paola Mello

Logica dei Predicati e Linguaggio Naturale

Non sempre è facile rappresentare frasi del linguaggio naturale in logica.

"Esiste un cane nero"

Se il dominio dell'interpretazione (universo del discorso) è solo di cani:

$$\exists X \ nero(X).$$

Se il dominio ha anche altri oggetti che non sono cani, devo aggiungere la proprietà di essere cani:

$$\exists X \ (nero(X) \wedge cane(X)).$$

Errore !:

$\exists X \ (cane(X) \rightarrow nero(X))$ è equivalente a $\exists X \ (nero(X) \vee \neg cane(X))$.

Tale formula è vera in ogni dominio per cui c'è un oggetto nero o c'è un oggetto che non è un cane.

Logica dei Predicati e Linguaggio Naturale (cont.)

"Tutti i corvi sono neri"

Se il dominio dell'interpretazione (universo del discorso) è solo di corvi:

$$\forall X \; nero(X).$$

Se il dominio ha anche altri oggetti che non sono corvi devo aggiungere la proprietà di essere corvi:

$$\forall X \; (corvo(X) \rightarrow nero(X)).$$

Diverso significato:

$\forall X \; (corvo(X) \land nero(X))$ è equivalente a:

$$\forall X \; (nero(X)) \land \forall X \; (corvo(X)).$$

Tutti gli oggetti del dominio sono corvi e sono neri

Logica dei Predicati e Linguaggio Naturale (cont.)

"Tutte le scimmie sono fuggite su un albero"

Il dominio contiene differenti oggetti: (scimmie, alberi + il predicato fuggire)

Procedimento Top-down:

$$\forall X\ (scimmia\ (X) \rightarrow A(X)).$$

Dove A(X) è una formula logica non atomica che rappresenta "X è fuggito su un albero", ovvero esiste un albero su cui X è fuggito:

$$\exists Y\ (albero(Y) \wedge fugge(X,Y)).$$

Dunque:

$$\forall X\ \exists Y\ (scimmia\ (X) \rightarrow fugge(X,Y) \wedge albero(Y)).$$

Significato, alberi possibilmente diversi per scimmie diverse (l'albero dipende da X)

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Logica dei Predicati e Linguaggio Naturale (cont.)

Altro significato:

"Tutte le scimmie sono fuggite sullo stesso albero"

In altro modo:

Esiste un albero su cui sono fuggite tutte le scimmie

Procedimento Top-down:

$$\exists Y\ (albero(Y) \wedge \forall X\ (scimmia\ (X) \rightarrow fugge(X,Y)))$$

Errore:

$$\forall X\ \exists Y\ (scimmia\ (X) \wedge fugge(X,Y) \wedge albero(Y)).$$

Ovvero:

$$\forall X\ scimmia\ (X) \wedge \exists Y\ (fugge(X,Y) \wedge albero(Y)).$$

Afferma che tutti gli oggetti sono scimmie e tutti gli oggetti sono fuggiti sull'albero.

Logica dei Predicati e Linguaggio Naturale (cont.)

"esiste una taratartaruga che è più veccchia di ogni essere umano"

$$\exists X\ (\text{tartaruga}(X) \ \wedge\ C(X)).$$

Dove C(X) e` una formula logica non atomica che rappresenta "X e' piu' vecchio di tutti gli esseri umani":

$$\forall Y\ \ \text{umano}(Y)\ \rightarrow\ \text{piu-vecchio}(X,Y).$$

Dunque:

$$\exists X\ (\text{tartaruga}(X)\ \ \wedge\ \ \forall Y\ \text{umano}(Y)\ \rightarrow\ \text{piu-vecchio}(X,Y)).$$

Sbagliata:

$$\exists X\ (\text{tartaruga}(X) \rightarrow \forall Y\ \text{umano}(Y)\ \rightarrow\ \text{piu-vecchio}(X,Y)).$$

(Significato vero se non esistono tartarughe mentre la frase originale lo afferma)

Esercizio 3 – compito del 5/11/2003

(si consultino altri compiti nel materiale del corso in virtuale.unibo.it)

Si formalizzino le seguenti frasi in logica dei predicati:

- Esiste almeno uno studente di Ingegneria che conosce la logica booleana.
- Chi conosce la logica booleana ha capacità logiche.
- Chi non ha capacità logiche, si contraddice.
- Chi si contraddice, non ha capacità logiche.
- Piero studia ad ingegneria e conosce la logica booleana.

Si trasformino in clausole e si usi poi il principio di risoluzione per dimostrare che c'è uno studente di Ingegneria che non si contraddice.

Soluzione

- Esiste almeno studente di Ingegneria che conosce la logica booleana.

  $\exists$Y (studIng(Y) and conosce(Y,boole))

- Chi conosce la logica booleana ha capacità logiche.

  $\forall$ **X (conosce(X,boole) => haLogica(X))**

- Chi non ha capacità logiche, si contraddice.

  $\forall$ **X (not haLogica(X) => contraddice(X))**

- Chi si contraddice, non ha capacità logiche.

  $\forall$ **X (contraddice(X) => not haLogica(X))**

- Piero studia ad ingegneria e conosce la logica booleana.

  studIng(piero) and conosce(piero,boole)

Goal: $\exists$**Y studIng(Y) and not contraddice(Y)**

Soluzione

Clausole:

- C1 studIng(c)
- C2 conosce(c,boole)
- C3 not conosce(X,boole) or haLogica(X)
- C4 haLogica(X) or contraddice(X)
- C5 not contraddice(X) or not haLogica(X)
- C6 studIng(piero)
- C7 conosce(piero,boole)

- C8 not studIng(Y) or contraddice(Y) (goal negato)

Soluzione

- C1 studIng(c)
- C2 conosce(c,boole)
- C3 not conosce(X,boole) or haLogica(X)
- C4 haLogica(X) or contraddice(X)
- C5 not contraddice(X) or not haLogica(X)
- C6 studIng(piero)
- C7 conosce(piero,boole)
- C8 not studIng(Y) or contraddice(Y)

Risoluzione:

- C9 not haLogica(Y) or not studIng(Y) (da C5 e C8)
- C10 not conosce(Y,boole) or not studIng(Y) (da C9 e C3)
- C11 not conosce(piero,boole) (da C10 e C6)
- C12 Clausola vuota (da C11 e C7)

OR esclusivo: come si traduce

- Ogni studente è promosso o bocciato (or esclusivo)

  ∀ X studente(X) -> promosso (X) or-ex bocciato (X).

  ∀ X studente(X) ->(promosso (X) or bocciato (X)) and
  (not promosso (X) or not bocciato (X))

  (∀ X studente(X) -> promosso (X) or bocciato (X)) and
  (∀ X studente(X) -> not promosso (X) or not bocciato (X))

- C1: not studente (X) or promosso (X) or bocciato (X)
- C2: not studente(X) or not promosso (X) or not bocciato (X))