Linguaggi e assunzioni ontologiche/epistemiologiche

Language	Ontological Commitment (What exists in the world)	Epistemological Commitment (What an agent believes about facts)
Propositional logic	facts	true/false/unknown
First-order logic	facts, objects, relations	true/false/unknown
Temporal logic	facts, objects, relations, times	true/false/unknown
Probability theory	facts	degree of belief $\in [0,1]$
Fuzzy logic	facts with degree of truth $\in [0,1]$	known interval value

Cosa esiste nel mondo per F.O.L. (modelli): esempio

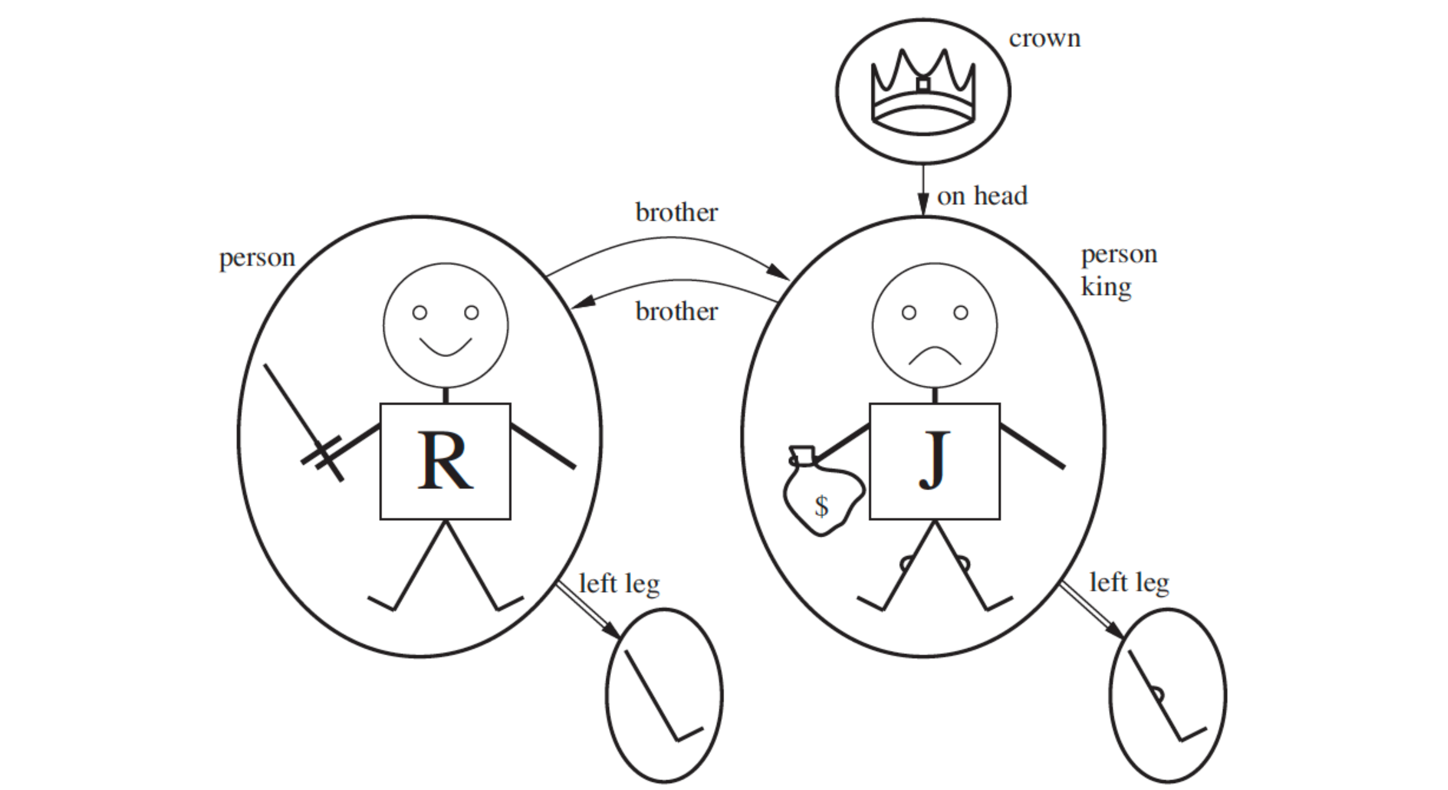

Figure 8.2 A model containing five objects, two binary relations, three unary relations (indicated by labels on the objects), and one unary function, left-leg.

Composto da **oggetti** (5) che sono relazionati attraverso **relazioni** (crown, person, person king, brother) e **funzioni** (left leg)

Sintassi logica I ordine

$$
\begin{array}{rcl}
\textit{Sentence} & \rightarrow & \textit{AtomicSentence} \mid \textit{ComplexSentence} \\
\textit{AtomicSentence} & \rightarrow & \textit{Predicate} \mid \textit{Predicate}(\textit{Term},\ldots) \mid \textit{Term} = \textit{Term} \\
\textit{ComplexSentence} & \rightarrow & (\ \textit{Sentence}\ ) \mid [\ \textit{Sentence}\ ] \\
& \mid & \neg\ \textit{Sentence} \\
& \mid & \textit{Sentence} \land \textit{Sentence} \\
& \mid & \textit{Sentence} \lor \textit{Sentence} \\
& \mid & \textit{Sentence} \ \Rightarrow \ \textit{Sentence} \\
& \mid & \textit{Sentence} \ \Leftrightarrow \ \textit{Sentence} \\
& \mid & \textit{Quantifier}\ \textit{Variable},\ldots\ \textit{Sentence} \\
\\
\textit{Term} & \rightarrow & \textit{Function}(\textit{Term},\ldots) \\
& \mid & \textit{Constant} \\
& \mid & \textit{Variable} \\
\\
\textit{Quantifier} & \rightarrow & \forall \mid \exists \\
\textit{Constant} & \rightarrow & A \mid X_1 \mid \textit{John} \mid \cdots \\
\textit{Variable} & \rightarrow & a \mid x \mid s \mid \cdots \\
\textit{Predicate} & \rightarrow & \textit{True} \mid \textit{False} \mid \textit{After} \mid \textit{Loves} \mid \textit{Raining} \mid \cdots \\
\textit{Function} & \rightarrow & \textit{Mother} \mid \textit{LeftLeg} \mid \cdots \\
\textsc{Operator Precedence} & : & \neg, =, \land, \lor, \Rightarrow, \Leftrightarrow
\end{array}
$$

Elementi delle Formule F.O.L. (esempi)

- Predicato:
- Formula atomica: …
- Termine costante: …
- Termine funzione: …
- Termine variabile: …
- Termine ground (istanziato): …
- Quantificazione esist.: …
- Quantificazione univ.: ...

Importanza Ordine dei Quantificatori e nome variabili

$\exists x \forall y \ Ama(x, y)$ e $\exists y \forall x \ Ama(x, y)$

Hanno lo stesso significato?

Esprimere significato in linguaggio naturale …

$\exists x \forall y \ Ama(x, y)$ e $\forall x \exists y \ Ama(x, y)$

Hanno lo stesso significato?

Esprimere significato in linguaggio naturale …

Nomi delle variabili

$\forall x\, Q(x)\ e\ \forall y\, Q(y)$ *semanticamente equivalenti*

$\exists x\, P(x)\ e\ \exists y\, P(y)$ *semanticamente equivalenti*

$\exists x\, (P(x) \Rightarrow \forall x\, Q(x))$ si legge
$\exists x\, (P(x) \Rightarrow (\forall x\, Q(x)))$

Le variabili quantificate hanno campi d'azione (scope) come nei linguaggi di programmazione

De Morgan e Quantificatori

$$\begin{aligned}
\forall x \;\; \neg P &\equiv \neg\exists x \;\; P \\
\neg\forall x \;\; P &\equiv \exists x \;\; \neg P \\
\forall x \;\; P &\equiv \neg\exists x \;\; \neg P \\
\exists x \;\; P &\equiv \neg\forall x \;\; \neg P
\end{aligned}
\qquad\qquad
\begin{aligned}
\neg(P \vee Q) &\equiv \neg P \wedge \neg Q \\
\neg(P \wedge Q) &\equiv \neg P \vee \neg Q \\
P \wedge Q &\equiv \neg(\neg P \vee \neg Q) \\
P \vee Q &\equiv \neg(\neg P \wedge \neg Q)\,.
\end{aligned}$$

Esercizio: trasformare I quantificatori nelle seguenti formule;

$\forall x$ Piace(x,Gelato)
$\exists x$ Piace(x, Broccoli)

Alcune Restrizioni Pratiche di F.O.L. (database semantics)

- **Unique name assumption**

Ogni symbolo di costante si riferisce a un oggetto differente

- **Closed world assumption**

Le formule atomiche che non son asserite essere vere sono false

- **Domain closure**

Ogni modello contiene non più oggetti di quelli indicati dai simboli di costante presenti nelle formule

Compilazione formule F.O.L. in Logica Proposizionale (LP)

*Ogni formula che contine una variabile universale si può **riscrivere** con una congiunzioni di formule senza questa variabile*

$$\exists x \forall y\ Ama(x, y) = \exists x\ Ama(x, Maria) \wedge Ama(x, Paolo) \wedge \ldots.$$

*Ogni formula che contine una variabile esistenziale si può **riscrivere** con una disgiunzione di formule senza questa variabile*

$$\exists x \forall y\ Ama(x, y) = \forall y\ Ama(Maria, y) \vee Ama(Paolo, y) \vee \ldots$$

Formula Φ ➔ "Grounding" di Φ ➔ Formula Φ in LP

Allora Perché Usare F.O.L.?

Almeno tre motivi

1.Non compilabile se il dominio non è finito o troppo grande

2.Anche se compilabile è comunque utile come *linguaggio* per l'ingegnere della conoscenza perchè è conciso

3.Può essere più efficiente ragionare direttamente in F.O.L. che nella compilazione PL (vedremo dopo)

Esempi Formule F.O.L.

$$\forall m, c \;\; Mother(c) = m \;\Leftrightarrow\; Female(m) \wedge Parent(m, c)\ .$$

One's husband is one's male spouse:

$$\forall w, h \;\; Husband(h, w) \;\Leftrightarrow\; Male(h) \wedge Spouse(h, w)\ .$$

Male and female are disjoint categories:

$$\forall x \;\; Male(x) \;\Leftrightarrow\; \neg Female(x)\ .$$

Parent and child are inverse relations:

$$\forall p, c \;\; Parent(p, c) \;\Leftrightarrow\; Child(c, p)\ .$$

A grandparent is a parent of one's parent:

$$\forall g, c \;\; Grandparent(g, c) \;\Leftrightarrow\; \exists p \;\; Parent(g, p) \wedge Parent(p, c)\ .$$

A sibling is another child of one's parents:

$$\forall x, y \;\; Sibling(x, y) \;\Leftrightarrow\; x \neq y \wedge \exists p \;\; Parent(p, x) \wedge Parent(p, y)\ .$$

Operatori Tell & Ask per la KB

Tell(KB, Φ): asserisce Φ nella KB

Ask(KB,Ψ): chiede a KB se Ψ vera (cons logica di KB)

Nella KB precedente aggiungo:

Tell(KB,Female(Maria))
Tell(KB,Genitore(Maria,Piero))
Tell(KB,Genitore(Piero,Marco))

Se ora chiedo

Ask(KB,Grandparent(Maria,Marco)
Ask(KB, $\exists x$ Child(Piero,x)

Cosa ottengo?

Assiomi di Peano (N e +)

1) $NatNum(0)$.

2) $\forall n \ \ NatNum(n) \Rightarrow NatNum(S(n))$

3) $\forall n \ \ 0 \neq S(n)$.

4) $\forall m, n \ \ m \neq n \Rightarrow S(m) \neq S(n)$

5) $\forall m \ \ NatNum(m) \Rightarrow +(0, m) = m$.

6) $\forall m, n \ \ NatNum(m) \wedge NatNum(n) \Rightarrow +(S(m), n) = S(+(m, n))$

+(3,2) = +(S(S(S(0))),S(S(0))) =
S(+(S(S(0)),S(S(0)))) = S(S(+(S(0),S(S(0))))) =
S(S(S(+(0,S(S(0)))))) = S(S(S(S(S(0)))))

Dimostrazione applicando 1), 5) e piu' volte 2) e 6)

Esercizio su Uso dei Quantificatori

Usando i predicati binari *Conosce* e *Studente* e le costanti *ing-inf, ing-ges, java*

Formalizzare in due formule FOL:

1. Gli studenti di ingegneria informatica conoscono Java
2. Qualche studente di ingegneria gestionale conosce Java

Quale regola generale che posso osservare da questi esempi?

Semantica di F.O.L.

Una **interpretazione** per un insieme di formule in FOL è definita da un **dominio** D (di oggetti) non vuoto e un **assegnamento** che mappa

- ogni simbolo di costante in un oggetto di D
- ogni simbolo di predicato n-ario in una relazione n-aria in D^n (un sottoinsieme di D x D x ….. x D n volte)
- ogni simbolo di funzione n-aria in una funzione da D^n a D

NB: “=“ sempre intepretato con la relazione di identità

Esempi: *Amico-di* (relazione), *Madre-di* (funzione)

E i simboli di variabile/quantificazione? (tra poco…)

Semantica Formula Atomica F.O.L.

La verità di una formula atomica **dipende dall'interpretazione**

Una formula atomica è vera rispetto a una certa interpretazione se la relazione a cui si riferisce il simbolo di predicato è soddisfatta dagli oggetti riferiti dagli argomenti del simbolo di predicato

NB: La relazione è soddisfatta se e solo se la tuple di oggetti è nella relazione

Semantica Formule Quantificate

- Una formula del tipo $\forall x\ \Phi$ è vera secondo una intepretazione se, per ogni elemento *d* del dominio, la formula Φ' ottenuta assegnado *d* a *x* *è vera* (altrimenti $\forall x\ \Phi$ è falsa)

- Una formula del tipo $\exists x\ \Phi$ è vera secondo una intepretazione se, esiste un element *d* del dominio tale per cui la formula Φ' ottenuta assegnado *d* a *x* *è vera* (altrimenti $\exists x\ \Phi$ è falsa)

Semantica FOL (Modelli)

- **Variabili libere**: variabili che non compaiono all'interno del campo d'azione di un quantificatore. Ad es. $\exists x\ Ama(x, y)$
- **Formule chiuse**: una FBF è chiusa se non ha variabili libere. Ad es. $\exists x \forall y\ Ama(x, y)$
- **Modello**: una interpretazione I per una FBF chiusa Φ è un modello per Φ se e solo se Φ è vera rispetto a I
- **Formula soddisfacibile**: Una FBF Φ è soddisfacibile se esiste un modello per Φ
- **Modello per una KB**: modello per la formula ottenuta mettendo in congiunzione le formule della KB

Teoria Assiomatica della F.O.L e Decidibilità

- Insieme di assiomi/formule (che formano una KB)
- Insieme di Regole di Inferenza (che agiscono sulla KB)

Una teoria è **decidibile** se esiste un metodo meccanico (algoritmo/i) per decidere se qualsiasi FBF è un teorema o no

La logica del I ordine è **semidecidibile** (la LP è decidibile)

Semidecidibilità:

- *se una formula è un teorema, esiste un algoritmo per derivarla in un numero finite di passi (potrei non conoscerlo).*
- *Se una formula non è un teorema,* ***non*** *esiste un algoritmo per determinarlo (la dimostrazione potrebbe non terminare)*

Esercizio di formalizzazione

Formalizzae in FOL il concetto di “Studente Modello” che ha

(1) almeno due corsi con esame superato per ogni sessione

(2) al massimo un appello d’esame non superato

(3) almeno un corso con appello superato e volto 30lode

Predicati e costanti da usare:

- *Studmodello(stud)*
- *Supera(stud,corso,sessione,voto)*
- *Bocciato(stud,appello)*
- $=$
- *30lode*