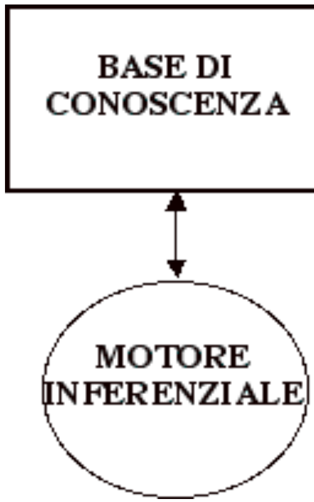
SISTEMI BASATI SULLA CONOSCENZA: PRINCIPI ARCHITETTURALI

- Ogni sistema basato sulla conoscenza deve riuscire ad esprimere **due** tipi di conoscenza in modo **separato** e **modulare**:
  - Conoscenza sul dominio dell'applicazione (**COSA**);
  - Conoscenza su **COME** utilizzare la conoscenza sul dominio per risolvere problemi (**CONTROLLO**).
- **Problemi:**
  - Come esprimere la conoscenza sul problema?
  - Quale strategia di controllo utilizzare?

Rappresentazione

Strategia



To sum up …

- Modulo di controllo applicato per:
  - Cercare sequenze di azioni: ricerca in uno spazio degli stati (cieca *vs* euristica)
  - Determinare la prossima mossa (*adversary search*)
  - Determinare la/le soluzioni di un CSP
    - Algorimi di propagazione
    - Tecniche di consistenza
- Rappresentazione del problema (azioni, stato, CSP, etc)

ESEMPIO (*già visto ...*)

ESEMPIO

- **Semplicissimo problema di diagnostica:**
  - prescrivere una medicina in base ai risultati di un esame di laboratorio.

- **GOAL:** `prescribe (Drug)`
  - cioè "prescrivere una medicina adeguata per un determinato paziente".

BASE DI CONOSCENZA (COSA)

- FATTI:

```
gram(neg).
not(allergic(antb)).
```

- REGOLE:
  - **R1: gram (neg) → id (ecoli).**

    Se il risultato dell'esame è *gram-negativo* allora l'identità è *enterium-coli*
  - **R2: gram (pos) → id (strep).**

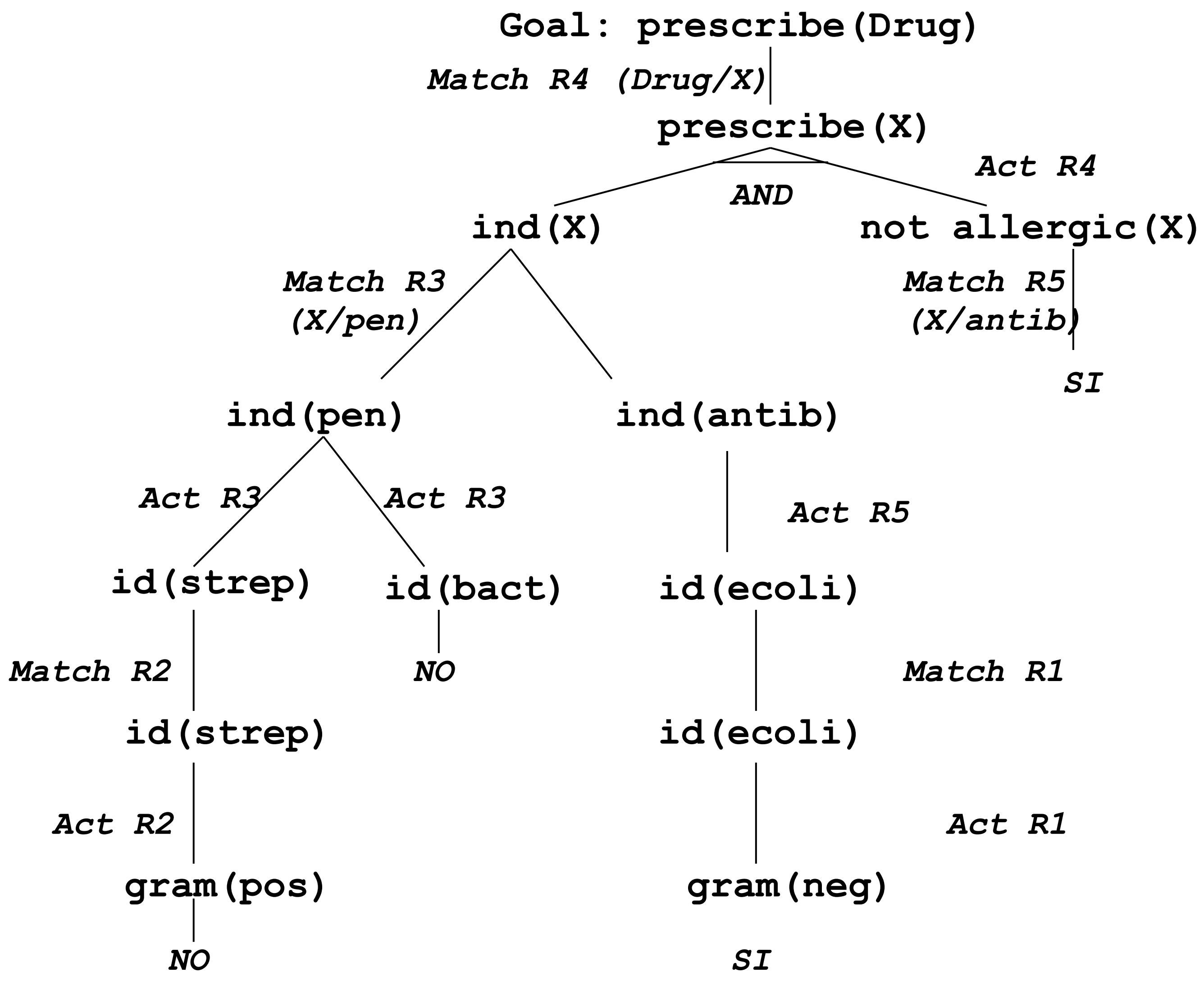
    Se il risultato dell'esame è *gram-positivo* allora l'identità è *streptococco*
  - **R3: id(strep) OR id(bact) → ind(pen).**

    Se l'identità è streptococco o bactero allora è bene indicare penicillina
  - **R4: ind(X) AND not (allergic(X)) → prescribe(X)**

    Se è bene indicare una certa medicina e il paziente non è allergico a tale medicina, allora si può prescrivere tale medicina al paziente
  - **R5: id(ecoli) → ind(antb).**

    Se l'identità è ***enterium-coli*** allora è bene indicare antibiotici

CONTROLLO (BACKWARD)
GRAFO AND/OR



Da albero AND/OR a albero OR

GRAFO OR

SI', Drug=antib

?- prescribe(Drug)

Match R4 (Drug/X)

?- ind(X),not allergic(X)

Match R3 (X/pen)

Match R5 (X/antib)

?-ind(pen), not allergic(pen)

?-ind(antib),not allergic(antib)

R3

R3

Act R5

?-id(strep),not__

?-id(bact),not__

?-id(ecoli),not__

R2

NO

R1

?-gram(pos),not___

NO

? gram(neg),not allergic(antib)

Fatto 1

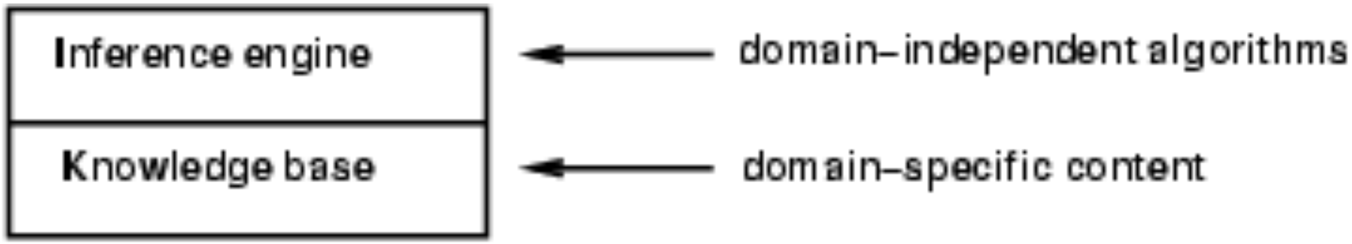
? not allergic(antib)

Fatto 2

SI'

Lo spazio di ricerca delle soluzioni è un albero OR, esplorato secondo opportune strategie

Basi di Conoscenza



- Knowledge base (KB) = insiemi di frasi / sentenze /proposizioni scritte in un linguaggio formale.

Le risposte devono “discendere” dalla KB.

- Inference Engine: strutture dati ed algoritmi per manipolare la KB ed arrivare ad una risposta.

- **Considereremo come linguaggio formale la logica dei predicati del primo ordine (FOL, First Order Logic) e Prolog come linguaggio di programmazione**
- **Vedremo poi altri approcci object-based, parleremo di Logiche Descrittive, ontologie e Semantic Web**

A cosa serve la logica?

- La logica serve a rappresentare conoscenza nota (assiomi), in un opportuno linguaggio formale
- L'apparato di calcolo (ad es., il calcolo dei predicati del I ordine) deriva teoremi (nuove formule) dagli assiomi

Partiamo con un po' di terminologia …

Agenti logici

- Gli agenti logici applicano inferenze a una base di conoscenza per derivare nuove informazioni.

- Concetti base della logica:
  - sintassi: struttura formale delle sentenze (*frasi*)
  - semantica: verità di sentenze rispetto ad interpretazioni/modelli
  - conseguenza logica (entailment): sentenza necessariamente vera data un'altra sentenza
  - inferenza: derivare (sintatticamente) sentenze da altre sentenze
  - correttezza (soundness): la derivazione produce solo sentenze che sono conseguenza logica.
  - completezza (completeness): la derivazione può produrre tutte le conseguenze logiche.

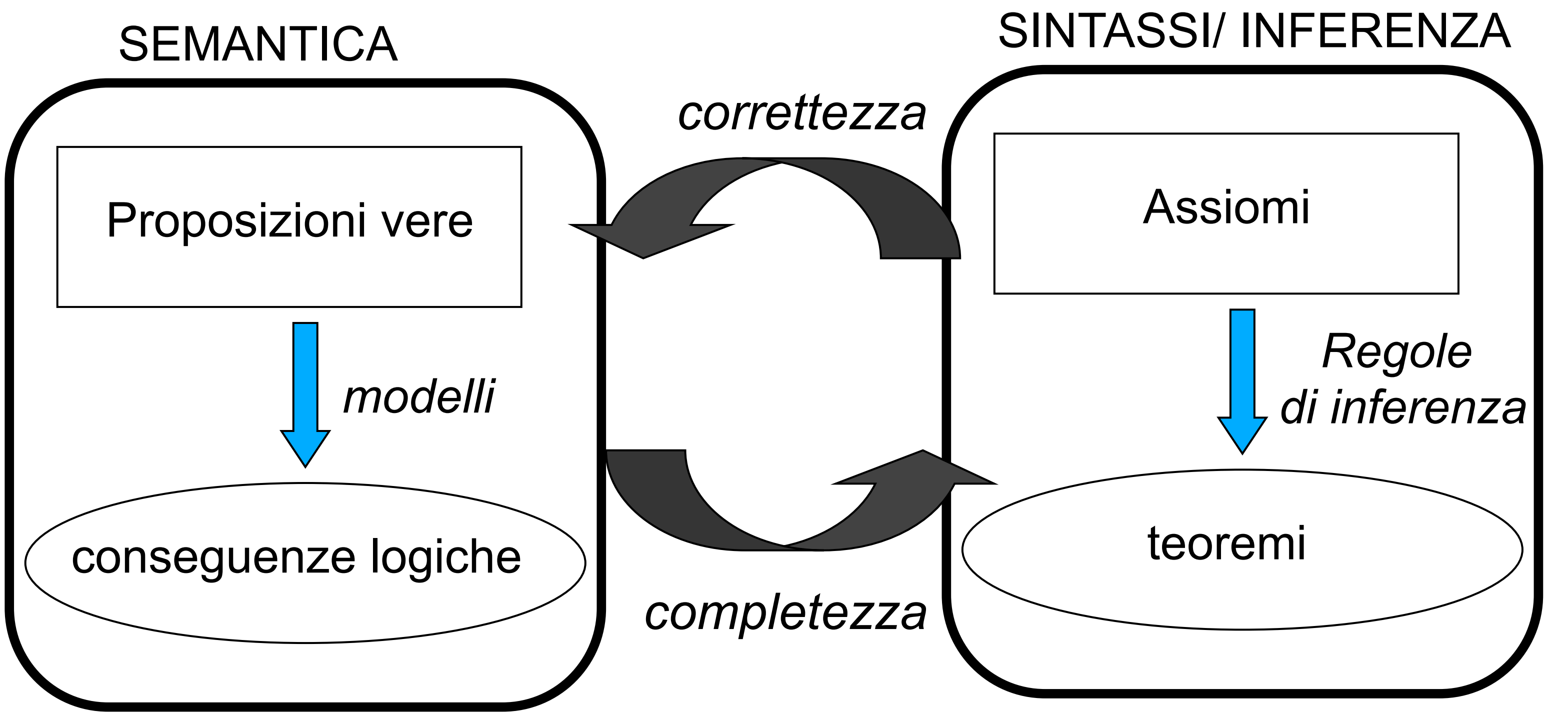
☞In ogni linguaggio abbiamo due aspetti:

- sintassi: struttura formale delle frasi
- semantica: “significato” delle frasi

- In C, ad esempio, sintassi espressa in notazione EBNF, semantica operazionale data dalla macchina astratta C (supporto run-time)

- In logica, formule ben formate costruite con parentesi, connettivi, quantificatori su variabili, atomi e termini …
- Semantica a modelli (model-theoretic) e semantica operazionale (proof system)

Logica: apparato semantico e sintattico



LA LOGICA DEI PREDICATI DEL PRIMO ORDINE

- *Materiale dal libro: L. Console, E. Lamma, P. Mello, M.Milano: Programmazione Logica e Prolog, Seconda Edizione UTET editore.*

- La logica è quella scienza che fornisce all'uomo gli strumenti indispensabili per controllare con sicurezza la rigorosità dei ragionamenti.

- **La logica** fornisce gli strumenti formali per:
  - analizzare le inferenze in termini di **operazioni su espressioni simboliche**;
  - dedurre conseguenze da certe premesse;
  - studiare la **verità o falsità di certe proposizioni** data la verità o falsità di altre proposizioni;
  - stabilire la consistenza e la validità di una data teoria.

LOGICA CLASSICA

- Si suddivide in due classi principali:
  - logica proposizionale
  - logica dei predicati
- La principale differenza tra le due classi è in termini di espressività: nella logica dei predicati è possibile esprimere variabili e quantificazioni, mentre questo non è possibile nella logica proposizionale
- First Order Logic (FOL)

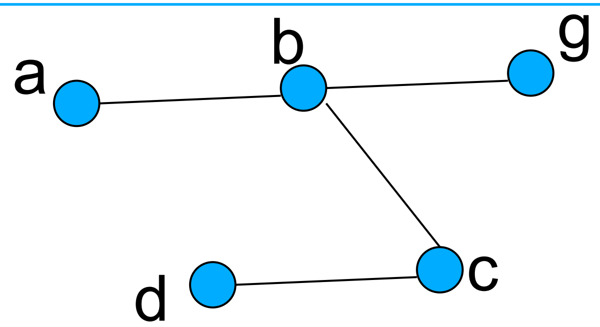
LOGICA dei PREDICATI DEL I ORDINE

- Il linguaggio della logica dei predicati del primo ordine è definito da:
  - una **sintassi**: caratteristiche strutturali del linguaggio formale (mediante una grammatica) senza attribuire alcun significato ai simboli;
  - una **semantica**, che interpreta le frasi sintatticamente corrette del linguaggio. Si dà una interpretazione alle formule stabilendo se una frase è vera o falsa.

ESEMPI

- $p \wedge q$ (E0)
- parente(giovanna, maria) (E1)
- $\exists X$ (uomo(X) $\wedge$ felice(X)) (E2)
- $\forall X$ (uomo(X) $\rightarrow$ mortale(X)) (E3)
- $\exists X$ (uomo(X) $\wedge$) (E4)
- $\exists X$ (uomo(f(X)) (E5)

LOGICA DEI PREDICATI DEL I ORDINE: esempio



{ arco(a,b).

arco(b,c).

arco(c,d).

arco (b,g).

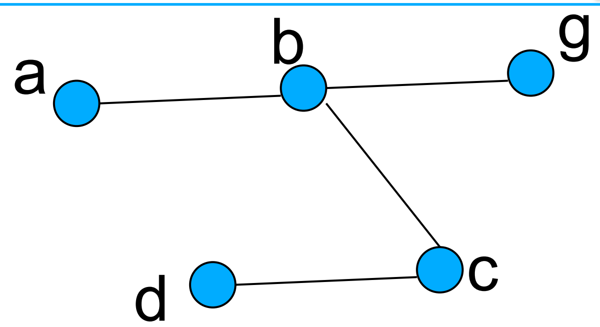
$\forall X\ \forall Y$ (connesso(X,Y) $\leftarrow$ arco(X,Y)).

$\forall X\ \forall Y$ (connesso(X,Y) $\leftarrow \exists Z$(arco(X,Z), connesso(Z,Y)))). }

- Posso verificare <u>automaticamente</u> se “a è connesso a d”, cioè se la formula connesso(a,d) è vera?

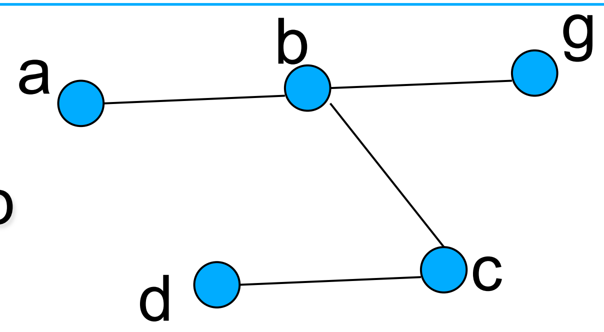
Calcolo del modello (se non è infinito)

{ arco(a,b). connesso(a,b).
arco(b,c). connesso(b,c).
arco(c,d). connesso(c,d).
arco (b,g). connesso(b,g).
connesso(a,g).
connesso(a,c).
connesso(b,d).
connesso(a,d). }



- E' l'approccio di Answer Set Programming (ASP) (approccio *bottom-up, forward*, dal corpo alla testa della "regola")

Derivazione (alla Prolog)



?- connesso(a,d)

?- arco(a,Z), connesso(Z,d) Z=b

?- arco(a,b), connesso(b,d)

?- connesso(b,d)

?- arco(b,Z’),connesso(Z’,d) Z’=c

?- arco(b,c),connesso(c,d)

?-connesso(c,d)

?-arco(c,d)

?-

Approccio *top-down*, *backward*, dalla testa al corpo della “regola”

☞In ogni linguaggio abbiamo due aspetti:

- sintassi: struttura formale delle frasi
- semantica: “significato” delle frasi

- In C, ad esempio, sintassi espressa in notazione EBNF, semantica operazionale data dalla macchina astratta C (supporto run-time)

- In logica, formule ben formate costruite con parentesi, connettivi, quantificatori su variabili, atomi e termini …
- Semantica a modelli (model-theoretic) e semantica operazionale (proof system)

LOGICA DEI PREDICATI: SINTASSI

- Alfabeto, che consiste di cinque insiemi:
  - l'insieme dei simboli di costante, C;
  - l'insieme dei simboli di funzione, F;
  - l'insieme dei simboli di predicato (o relazione), P;
  - l'insieme dei simboli di variabile, V;
  - i connettivi logici:

    ~ (negazione),

    ∧ (congiunzione),

    ∨ (disgiunzione),

    ← (implicazione),

    ↔ (equivalenza),

    le parentesi “(“ “)”

    e i quantificatori esistenziale (∃) e universale (∀).

LOGICA DEI PREDICATI: SINTASSI

- Costanti: singole entità del dominio del discorso.
  - Es. “maria”, “giovanna”, “3” $\Rightarrow$ iniziale minuscola
- Variabili: entità non note del dominio,
  - Es. X, Y $\Rightarrow$ iniziale maiuscola
- Funzioni n-arie: individua univocamente un oggetto del dominio del discorso mediante una relazione tra altri “n” oggetti del dominio.
  - Es. madre(maria)
- Importante: le funzioni, in logica, non presuppongono alcun concetto di valutazione
- Predicati n-ari: generica relazione (che può essere vera o falsa) fra “n” oggetti del dominio del discorso.
  - Es. parente(giovanna,maria)

CON L'ALFABETO SI COSTRUISCONO:

- Termine (definito ricorsivamente):
  - una variabile è un termine;
  - una costante è un termine;
  - se f è un simbolo di funzione n-aria e $t_1,...t_n$ sono termini, allora $f(t_1,...,t_n)$ è un termine.
  - Es. maria, f(X)
- Atomo o formula atomica:
  - l'applicazione di un simbolo di predicato n-ario p a n termini $t_1,...,t_n$: $p(t_1,..,t_n)$.
  - Es. parente(giovanna,maria)

CON L'ALFABETO SI COSTRUISCONO:

- Termine (definito ricorsivamente)
- Atomo o formula atomica
- Espressione o formula: sequenza di simboli appartenenti all'alfabeto.
  - `parente(giovanna, maria)` (E1)
  - `∃X (uomo(X) ∧ felice(X))` (E2)
  - `∀X (uomo(X) → mortale(X))` (E3)
  - `∃X (uomo(X) ∧)` (E4)
  - `∃X (uomo(f(X))` (E5)

- **Formule ben formate (fbf):** frasi sintatticamente corrette del linguaggio

LOGICA DEI PREDICATI: fbf

- Formule ben formate (fbf): **frasi sintatticamente corrette del linguaggio**. Si ottengono attraverso combinazione di formule atomiche, utilizzando i connettivi e i quantificatori. Sono definite ricorsivamente come segue:
  - ogni atomo è una fbf;
  - se A e B sono fbf, allora lo sono anche $\sim A$, $A \wedge B$, $A \vee B$, $A \rightarrow B$, $A \leftrightarrow B$ (eventualmente racchiuse tra parentesi tonde bilanciate);
  - se A è una fbf e X è una variabile, $\forall X\ A$ e $\exists X\ A$ sono fbf.
- Le espressioni (E1), (E2), (E3) sono formule ben formate, mentre non lo sono (E4) e (E5).
- Letterale: fbf atomica o la sua negazione. Ad esempio, la formula (E1) è un letterale.

FBF: ESEMPI

- $p \wedge q$ (E0) ✓
- parente(giovanna, maria) (E1) ✓
- $\exists X\ (uomo(X) \wedge felice(X))$ (E2) ✓
- $\forall X\ (uomo(X) \rightarrow mortale(X))$ (E3) ✓
- $\exists X\ (uomo(X) \wedge)$ (E4) ✗
- $\exists X\ (uomo(f(X))$ (E5) ✗

- **Formule ben formate (fbf):** frasi sintatticamente corrette del linguaggio

REGOLE DI PRECEDENZA TRA OPERATORI

$\sim \exists \forall$

$\wedge$

$\vee$

$\leftrightarrow \rightarrow$ $\quad A \rightarrow B$ equivale a $\sim A \vee B$

- Esempio

La fbf:

$$a \vee \sim b \wedge \exists X \ c(X) \rightarrow d(X,Y)$$

è equivalente a:

$$(a \vee ((\sim b) \wedge (\exists X \ c(X)))) \rightarrow d(X,Y)$$

FORME NORMALI CANONICHE

- fbf in forma normale prenessa disgiuntiva ("**disjunctive prenex normal form**"):
  - disgiunzione di una o più fbf composte da congiunzioni di letterali;
  - le quantificazioni compaiono tutte in testa a F
  - *come la forma canonica SP in Reti logiche (**OR di AND**)*

- fbf in forma normale prenessa congiuntiva ("**conjunctive prenex normal form**"):
  - congiunzione di una o più fbf composte da disgiunzioni di letterali;
  - le quantificazioni compaiono tutte in testa ad F
  - *come la forma canonica PS in Reti logiche (**AND di OR**)*

FORME NORMALI: ESEMPI

La fbf:

$\exists X \forall Y \exists Z\ (a(X) \land b(Y,Z)) \lor (c(X) \land \sim a(Z) \land d) \lor f$

è in forma normale disgiuntiva.

La fbf:

$\exists X \forall Y \exists Z\ (a(X) \lor b(Y,Z)) \land (c(X) \lor \sim a(Z) \lor d) \land f$

è in forma normale congiuntiva.

- Qualunque fbf può essere trasformata in forma normale prenessa (congiuntiva o disgiuntiva) attraverso opportune <u>trasformazioni sintattiche</u>.

CAMPO D'AZIONE (SCOPE)

- **Campo di azione (scope)** di un quantificatore: fbf che lo segue immediatamente. Nel caso di ambiguità si utilizzano le parentesi tonde.
- Esempio
  - Nella fbf:

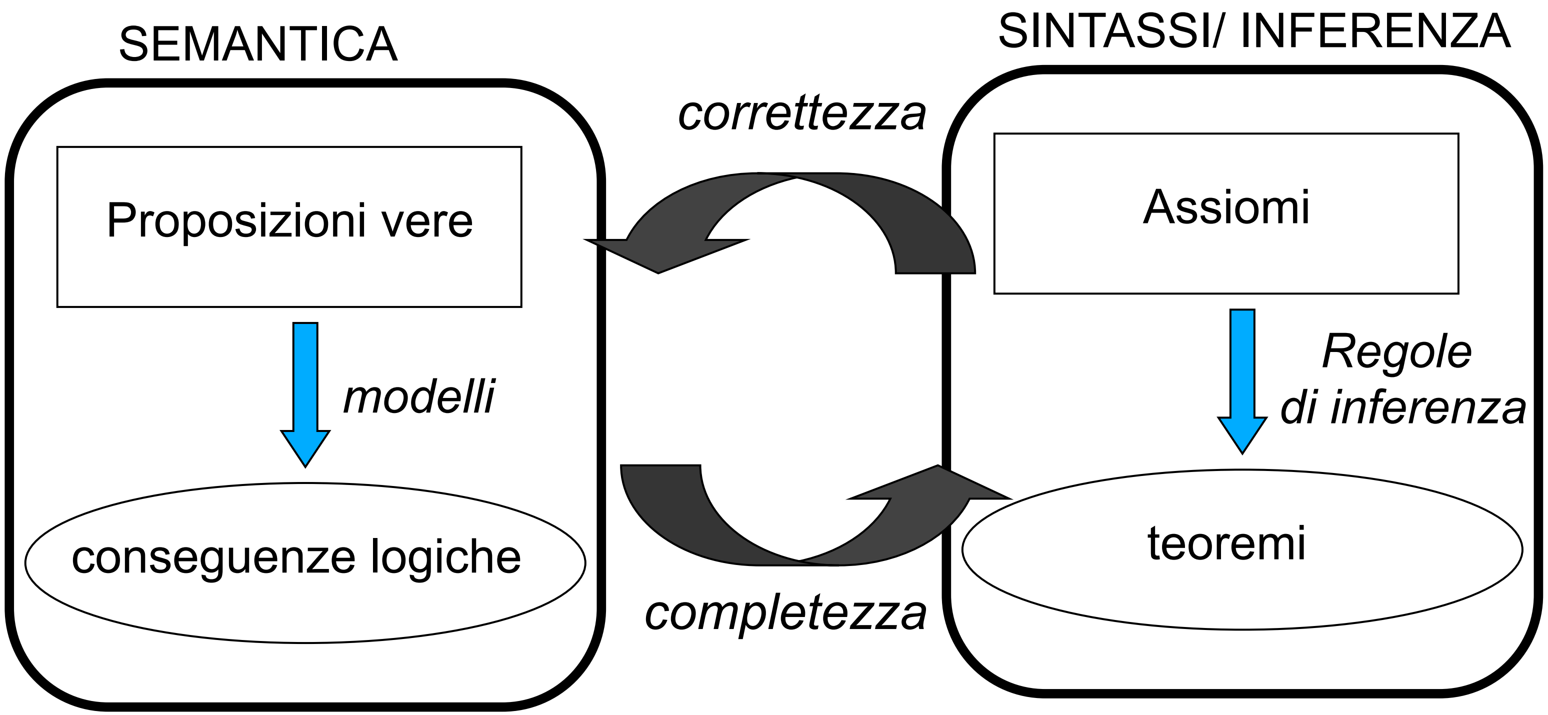
$$\forall X \ (p(X,Y) \wedge q(X)) \vee q(X)$$

la quantificazione sulla variabile X ha come campo d'azione la formula $p(X,Y) \wedge q(X)$

ALTRE DEFINIZIONI:

- **Variabili libere:** variabili che non compaiono all'interno del campo di azione di un quantificatore.
- Esempio nella fbf: $F = \forall X\ (p(X,Y) \land q(X))$ la variabile Y risulta libera in F.
- **Formule chiuse:** fbf che non contengono alcuna variabile libera. Ad esempio, le formule (E1), (E2) ed (E3) sono fbf chiuse. Nel seguito considereremo solo formule fbf chiuse.
- **Formule ground:** formule che non contengono variabili. Ad esempio la formula (E1) è una formula "ground".
- **Varianti:**una formula F2, ottenuta rinominando le variabili di una formula F1, è detta variante di F1.
- Esempio La formula: $\forall X\ \exists Y\ p(X,Y)$ è una variante della formula $\forall W\ \exists Z\ p(W,Z)$.

Due approcci: apparato semantico e sintattico



☞In ogni linguaggio abbiamo due aspetti:

- sintassi: struttura formale delle frasi
- semantica: "significato" delle frasi

- In C, ad esempio, sintassi espressa in notazione EBNF, semantica operazionale data dalla macchina astratta C (supporto run-time)

- In logica, formule ben formate costruite con parentesi, connettivi, quantificatori su variabili, atomi e termini …
- Semantica a modelli (model-theoretic) e semantica operazionale (proof system)

SEMANTICA

- Occorre associare un significato ai simboli.
- Ogni sistema formale è la modellizzazione di una certa realtà (ad esempio la realtà matematica).
- Un'interpretazione è la costruzione di un rapporto fra i simboli del sistema formale e tale realtà (chiamata anche **dominio del discorso**).
- Ogni formula atomica o composta della logica dei predicati del primo ordine può assumere il valore vero o falso in base alla frase che rappresenta nel dominio del discorso.
- **Esempio:**
  - $\forall X\ \forall Y\ \forall Z\ \ (op(X, Y, Z) \rightarrow op(Y, X, Z))$
  - se X, Y, Z variano sull'insieme dei numeri reali tale formula è vera se il simbolo di predicato "op" ha il significato di un operatore commutativo (es: somma o moltiplicazione), falsa se l'operatore non è commutativo (es. sottrazione o divisione).

Un po' di intuizione …

- Vogliamo rappresentare la conoscenza sul dominio come ***teoria logica*** data da un insieme di fbf chiuse (le cui variabili sono tutte quantificate) , ad esempio:

$$\{ \quad uomo(socrate), \quad \forall X\ (uomo(X) \Rightarrow mortale(X)) \quad \}$$

- e meccanizzare il ragionamento (quali sono le formule che seguono logicamente?):

$$mortale(socrate)$$

Un po' di intuizione … *(cont.)*

- Ogni sistema formale è la modellazione di una certa realtà (ad esempio la realtà matematica).
- ***Teoria logica*** data da un insieme di fbf chiuse (le cui variabili sono tutte quantificate) , ad esempio:

$$\{ \quad p(0,0), \quad \forall X\ p(X,X), \quad \forall X\ \forall Y\ (p(X,Y) \Rightarrow p(X,s(Y))) \quad \}$$

- Le formule di questa teoria sono vere solo in alcune ***interpretazioni*** (dette ***modelli***)
- Un' interpretazione, rapporto fra i simboli del sistema formale e tale realtà (chiamata anche **dominio del discorso**).

Un po' di intuizione … *(cont.)*

- Intensionalmente, con una ***teoria logica*** sottointendo i suoi modelli (possono essere infiniti, ed in numero infinito …)

$$\{ \quad p(0,0),$$
$$\forall X\ p(X,X),$$
$$\forall X\ \forall Y\ (p(X,Y) \Rightarrow p(X,s(Y))) \quad \}$$

- Operando a livello di modelli, possiamo cercare le formule che seguono logicamente dalla teoria (***conseguenze logiche***)

- In programmazione logica, si usa il modello di Herbrand: {p(0,0), p(0,s(0)), p(0,s(s(0))), …, p(s(0),s(0)), p(s(0),s(s(0))), … }

Un po' di intuizione … *(cont.)*

{ $p(0,0)$,

$\forall X\ p(X,X)$,

$\forall X\ \forall Y\ (p(X,Y) \Rightarrow p(X,s(Y)))$ }

Modello di Herbrand: ***è infinito!!***

{$p(0,0)$,

$p(0,s(0))$,

$p(0,s(s(0)))$, …,

$p(s(0),s(0))$,

$p(s(0),s(s(0)))$, … }

INTERPRETAZIONE

- Dato un linguaggio del primo ordine L, un' **interpretazione** per L definisce un dominio non vuoto D e assegna:
  - a ogni simbolo di costante in C, una costante in D;
  - a ogni simbolo di funzione n-ario F, una funzione: $F: D^n \rightarrow D$;
  - a ogni simbolo di predicato n-ario in P, una relazione in D^n, cioè un sottoinsieme di D^n.

  - Esempio: Linguaggio del primo ordine, L, nel quale si ha una costante "0", un simbolo di funzione unaria "s" e un simbolo di predicato binario "p"

INTERPRETAZIONE

- **Interpretazione I1** D: numeri naturali.
  - "0" rappresenta il numero zero.
  - "s" rappresenta il successore di un numero naturale
  - "p" rappresenta la relazione binaria "≤"

- **Interpretazione I2** D: numeri interi negativi.
  - "0" rappresenta il numero zero.
  - "s" rappresenta il predecessore di un numero naturale
  - "p" rappresenta la relazione binaria "≤"

$$\{ \quad p(0,0),$$

$$\forall X\ p(X,X),$$

$$\forall X\ \forall Y\ (p(X,Y) \Rightarrow p(X,s(Y))) \quad \} \quad \text{vera per I1, falsa per I2}$$

(I1 è un modello, I2 non lo è)

VALORE DI VERITÀ DI UNA fbf

- Data un'interpretazione il valore di verità di una fbf si definisce secondo le seguenti regole.
- **1) Formula atomica "ground"** ha valore vero sotto un'interpretazione quando il corrispondente predicato è soddisfatto (cioè quando la corrispondente relazione è vera nel dominio). La formula atomica ha valore falso quando il corrispondente predicato non è soddisfatto.
- Interpretazione I1:

p(0,s(0)) vero	"0 è minore o uguale a 1"
p(s(0), 0) falso	"1 è minore o uguale a 0"

- Interpretazione I2.

p(0,s(0)) falso	"0 è minore o uguale a -1"
p(s(0), 0) vero	"-1 è minore o uguale a 0"

VALORE DI VERITÀ DI UNA fbf (2)

- **2) Formula composta** il valore di verità di una formula composta rispetto a un'interpretazione si ottiene da quello delle sue componenti utilizzando le tavole di verità dei connettivi logici:

A	B	~A	A ∧ B	A∨ B	A ⇒B	A⇔B
T	T	F	T	T	T	T
T	F	F	F	T	F	F
F	T	T	F	T	T	F
F	F	T	F	F	T	T

Nota: *l'implicazione A ⇒ B è diversa rispetto al "se ..... allora" utilizzato nel linguaggio naturale.*

A: antecedente *B: conseguente*

VALORE DI VERITÀ DI UNA fbf (3)

- Data la formula F:

  volano(asini) $\Rightarrow$ ha_scritto(manzoni,promessi_sposi)

  assumendo l'interpretazione più intuitiva, F ha valore vero, poiché l'antecedente ha valore falso in tale interpretazione.

- La formula F:

  p(s(0),0) $\Rightarrow$ p(0,s(0))

  ha valore vero nell'interpretazione I1 poiché l'antecedente ha valore falso, mentre ha valore falso in I2 poiché a un antecedente vero corrisponde un conseguente falso.

VALORE DI VERITÀ DI UNA fbf (2)

- **3) Formula quantificata esistenzialmente:** una formula del tipo ∃X F è vera in un'interpretazione I se esiste almeno un elemento d del dominio D tale che la formula F', ottenuta assegnando d alla variabile X, è vera in I. In caso contrario F ha valore falso.
- Esempio: La formula ∃X p(X,s(0)) ha valore vero nell'interpretazione I1 in quanto esiste un numero naturale, zero, minore di uno, tale che la formula F'=p(0,s(0)) ha valore vero in I1.
- **4) Formula quantificata universalmente:** una formula del tipo ∀X F è vera in un'interpretazione I se per ogni elemento d del dominio D, la formula F', ottenuta da F sostituendo d alla variabile X, è vera in I. Altrimenti F ha valore falso.
- Esempio: La fbf ∀Y p(0,Y) ha valore vero rispetto alle interpretazioni I1 (dove viene interpretata come "0 è minore o uguale a ogni intero positivo Y"), mentre ha valore falso rispetto a I2 poiché esiste almeno un elemento del dominio che la falsifica (esempio non è vero che "0 è minore o uguale a –1").

MODELLI

- Data una interpretazione I e una fbf chiusa F, I è un **modello** per F se e solo se F è vera in I.
  - Esempio: Per la fbf $\forall Y\ p(0,Y)$ l'interpretazione I1 è un modello, mentre I2 non lo è.
- Una fbf è **soddisfacibile** se e solo se è vera almeno in una interpretazione, ovvero se esiste almeno un modello per essa.
- Una fbf che ha valore vero per tutte le possibili interpretazioni, cioè per cui ogni possibile interpretazione è un modello, è detta **logicamente valida**.
  - Esempio: La fbf $\forall X\ p(X) \vee \sim(\forall Y\ p(Y))$ è logicamente valida. Infatti, le formule $\forall X\ p(X)$ e $\forall Y\ p(Y)$ sono semplici varianti della stessa formula F e quindi hanno i medesimi valori di verità per qualunque interpretazione. In generale, $F \vee \sim F$ ha sempre valore vero, in modo indipendente dall'interpretazione.
- F logicamente valida $\Leftrightarrow$ ~F è non soddisfacibile.
- F è soddisfacibile $\Leftrightarrow$ ~F non è logicamente valida.

INSIEMI DI FORMULE (1)

- Un **insieme** di formule chiuse del primo ordine S è **soddisfacibile** se esiste una interpretazione I che soddisfa **tutte le formule** di S (cioè che è un modello per ciascuna formula di S). Tale interpretazione è detta modello di S.

Esempio: si consideri il seguente insieme di formule S:

- $S=\{\forall Y\ p(Y,Y),\ \ p(s(0),0) \Rightarrow p(0,s(0))\}$.
- L'interpretazione I1 è modello di S, mentre I2 non lo è. In I2 è infatti soddisfatta la prima formula dell'insieme, ma non la seconda.
- **Un insieme di formule S che non può essere soddisfatto da alcuna interpretazione, è detto insoddisfacibile (o inconsistente). Ad esempio l'insieme di formule {A,~A} è insoddisfacibile.**

INSIEMI DI FORMULE (2)

- Un **insieme** di formule chiuse del primo ordine S è **soddisfacibile** se esiste una interpretazione I che soddisfa **tutte le formule** di S (cioè che è un modello per ciascuna formula di S). Tale interpretazione è detta modello di S.
- Esempi di insiemi di formule insoddisfacibili sono:
  - $S1=\{\sim (\exists X\ \forall Y\ p(X,Y)),\ \exists X\ \forall Y\ p(X,Y)\}$
  - $S2=\{p(s(0),0) \Rightarrow p(0,s(0)),\ p(s(0),0),\ \sim p(0,s(0))\}$
- In S1, infatti, compaiono una formula e la sua negazione. In S2, per ogni interpretazione in cui $p(s(0),0)$ e $\sim p(0,s(0))$ sono vere, la formula $p(s(0),0) \Rightarrow p(0,s(0))$ non è vera, per la tabella di verità della negazione e dell'implicazione.

CONSEGUENZA LOGICA (1)

- Una formula F <u>**segue logicamente**</u> (o è conseguenza logica) da un insieme di formule S (e si scrive **S |= F**), se e solo se ogni interpretazione I che è un modello per S, è un modello per F.

Esempio: si consideri l'insieme di fbf S:

- $\{p(0,0), \forall X\ p(X,X), \forall X\ \forall Y\ (p(X,Y) \Rightarrow p(X,s(Y)))\}$
- Da S segue logicamente la formula F=p(0,s(0)) poiché ogni interpretazione I che soddisfa S soddisfa anche F.
- Dall'insieme S, invece, non segue logicamente la formula F1: p(s(0), 0) in quanto esiste un'interpretazione (I1) che soddisfa S, ma non F1.

Un po' di intuizione … *(cont.)*

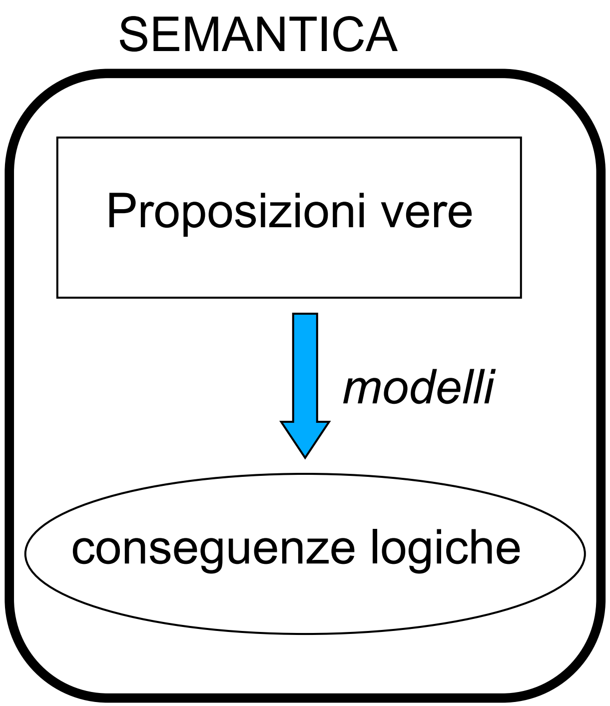
- Una formula F **segue logicamente** da una teoria logica S (e si scrive **S |= F**), se e solo se ogni modello per S, è un modello per F:

$$\{ \quad p(0,0), \quad \forall X\, p(X,X), \quad \forall X\, \forall Y\, (p(X,Y) \Rightarrow p(X,s(Y))) \quad \}$$

- F=p(0,s(0)) segue logicamente
- F1=p(s(0),0) non segue logicamente

(I1 è modello di S, ma non rende vera F1)

Logica: apparato semantico

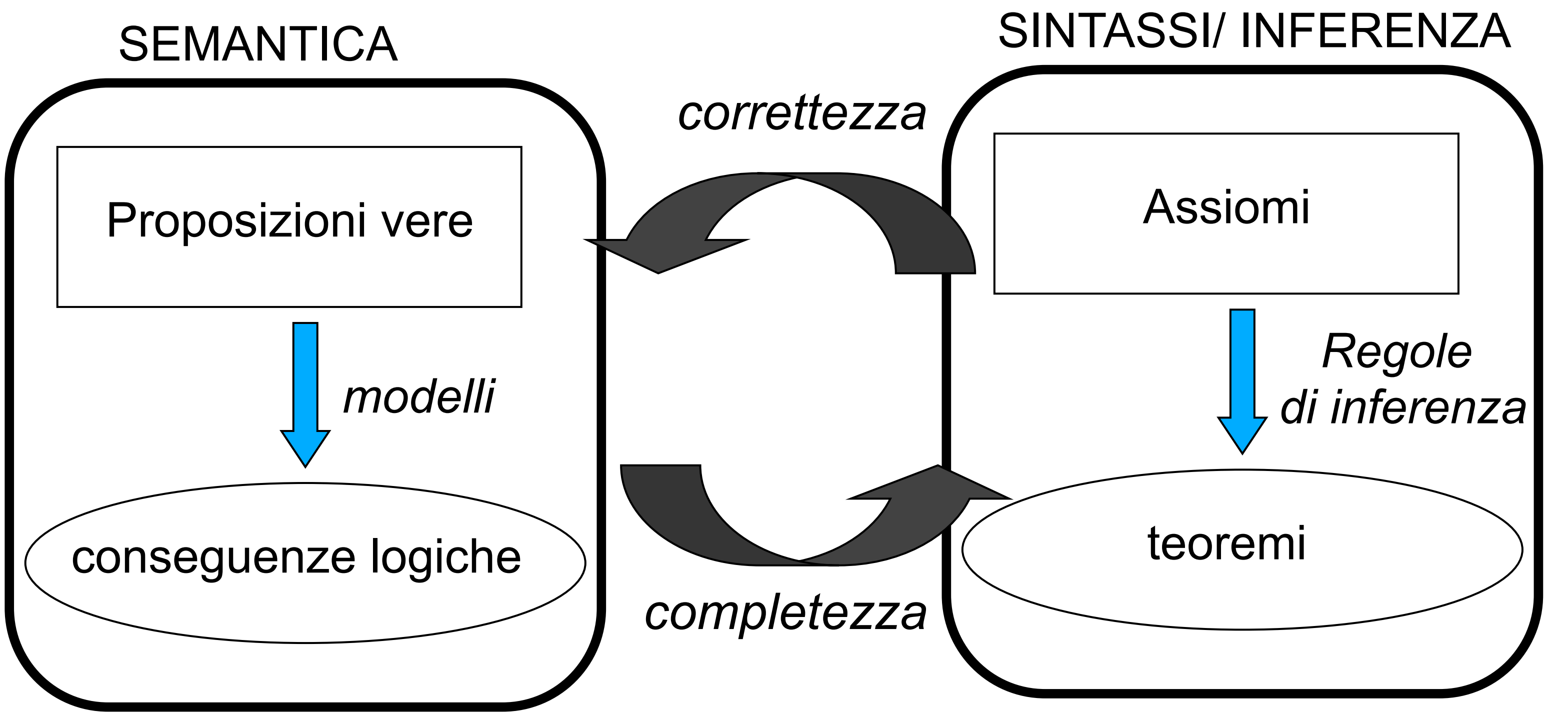


- Lavorare a livello semantico significa trattare con interpretazioni e **modelli spesso infiniti**
- Non è meccanizzabile la costruzione di un modello infinito (è un procedimento che **non termina!!**)
- Serve invece un apparato "meccanico" che lavori a livello sintattico (manipolazione di formule della teoria)

Riassumiamo

- Scrivendo un **insieme** di formule chiuse del primo ordine S **soddisfacibile** sottoindendiamo uno (o più) modelli di riferimento
- Quali formule F **seguono logicamente** (conseguenza logica, formule vere in tutte i modelli di S) dall'insieme di formule S (si scrive **S |= F**) e quali no?
- Ad esempio, S: {p(0,0), ∀X p(X,X), ∀X ∀Y (p(X,Y) ⇒p(X,s(Y)))} rappresenta in logica una relazione riflessiva e transitiva su un dominio infinito (da zero …)
- Da S segue logicamente la formula F=p(0,s(0)) poiché ogni interpretazione I che soddisfa S (è un modello di S) soddisfa anche F.
- Dall'insieme S, invece, non segue logicamente la formula F1: p(s(0), 0) in quanto esiste un modello (I1) di S, che non soddisfa F1.

Due approcci: apparato semantico e sintattico



Equivalenti?

Sì, se apparato di dimostrazione corretto e completo rispetto alla semantica

TEORIE DEL PRIMO ORDINE (1)

- **Calcolo proposizionale**: verifica di formula/e vera/e tramite le tavole di verità
- **Calcolo dei predicati del primo ordine**: tavole di verità troppo complesse. Dominio di interpretazione estremamente grande, se non infinito.
  - Si ricorre al metodo assiomatico (noto come **proof theory,** le regole di inferenza sono Modus Ponens e Specializzazione).
- La logica dei predicati proposizionale e del primo ordine può essere formulata come **sistema assiomatico-deduttivo**.

TEORIE DEL PRIMO ORDINE (2)

- **Teoria assiomatica:**
  - Assiomi, formule ben formate ritenute vere
  - Criteri di manipolazione sintattica:
    - Regole di inferenza derivano fbf da fbf
  - Scopo: produrre nuove formule sintatticamente corrette (teoremi).

TEORIE DEL PRIMO ORDINE (3)

- Semplificazioni:

$(A \wedge B)$	equivale a	$(\sim(A \rightarrow (\sim B)))$
$(A \vee B)$	equivale a	$((\sim A) \rightarrow B)$
$(A \equiv B)$	equivale a	$((A \rightarrow B) \wedge (B \rightarrow A))$

- Inoltre, per i quantificatori:

$\exists X\ A$	abbrevia	$\sim(\forall X \sim A)$
$\forall X\ A$	abbrevia	$\sim(\exists X \sim A)$

REGOLE DI INFERENZA

- **Modus Ponens (MP):**

$$\frac{A,\ A \rightarrow B}{B}$$

che deriva da due formule del tipo A e $A \rightarrow B$ la nuova formula B

- **Specializzazione (Spec):**

$$\frac{\forall X \quad A}{A(t)}$$

- Da una formula quantificata universalmente è possibile derivare una formula identica all'originale in cui la variabile X è sostituita da un elemento del dominio del discorso (costante e funzione).

DIMOSTRAZIONE DI TEOREMI (1)

- Dimostrazione: sequenza finita di fbf f_1, f_2, ..., f_n, tale che ciascuna f_i o è un assioma oppure è ricavabile dalle fbf precedenti mediante una regola di inferenza.
- Teorema: L'ultima fbf di ogni dimostrazione.
- Prova del teorema: sequenza di regole di inferenza applicate.
- Una fbf F è **derivabile** in una teoria T (**T |- F**) se esiste una sequenza di fbf f_1, f_2, ..., f_n, tale che f_n = F e, per ogni i, o f_i è un assioma di T, oppure è ricavabile dalle fbf precedenti mediante una regola di inferenza di T.

DIMOSTRAZIONE DI TEOREMI (2)

Esempio

- Teoria T: assiomi propri (relazione di minore uguale sui numeri naturali):

  $p(0,0)$ (A1)

  $\forall X\ \forall Y\ (p(X,Y) \Rightarrow p(X,s(Y)))$ (A2)

  $\forall X\ p(X,X)$ (A3)

- Teorema $p(0,s(0))$ (cioè T l- $p(0,s(0))$)
- Trasformazione da Spec e A2:

  $p(0,0) \Rightarrow p(0,s(0)))$

  applicando MP $p(0,s(0))$

DECIDIBILITÀ

- **Teoria decidibile**, teoria per la quale esiste un metodo meccanico per stabilire se una qualunque fbf è un teorema o non lo è.
- Il calcolo dei predicati del primo ordine non è decidibile, ma **semi-decidibile**: se una formula è un teorema, esiste un metodo meccanico che la deriva in un numero finito di passi. Se invece la formula non è un teorema, non è garantita, in generale, la terminazione del metodo meccanico (**Turing 1936**, Church 1936).

- Una teoria del primo ordine è un insieme di fbf chiuse (assiomi) e si può quindi parlare di modello di una teoria.
- Un modello per una teoria del primo ordine T è un'interpretazione che soddisfa tutti gli assiomi di T (assiomi logici e assiomi propri).
- Se T ha almeno un modello viene detta consistente (o soddisfacibile)

CORRETTEZZA E COMPLETEZZA (1)

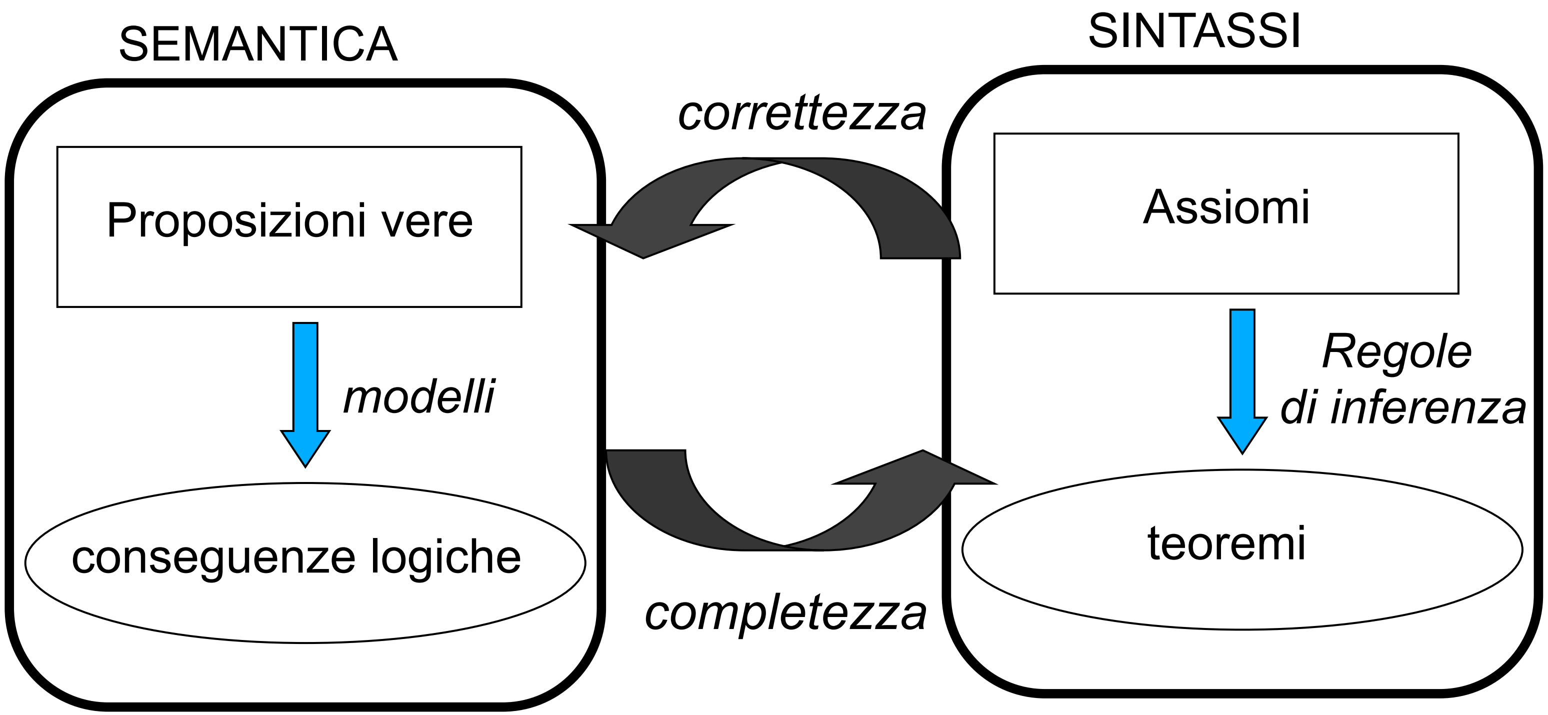
- Una teoria assiomatica è **corretta** se i teoremi dimostrati seguono logicamente dagli assiomi della teoria.

- Una teoria assiomatica è **completa** se tutte le fbf che seguono logicamente dalla teoria possono essere dimostrate come teoremi della teoria.

- Se T è corretta e completa è garantita l'equivalenza tra l'aspetto sintattico e semantico

$$T \vdash F \quad \Leftrightarrow \quad T \models F$$

CORRETTEZZA E COMPLETEZZA (2)



TEORIE ASSIOMATICHE NON CORRETTE

- Si consideri una teoria del primo ordine T, data dai seguenti assiomi propri che rappresentano la relazione di minore sui numeri naturali:

  $p(0,s(0))$ (A1)

  $\forall X \forall Y\ (p(X,Y) \rightarrow p(X,s(Y)))$ (A2)

  $\forall X\ p(X,s(X))$ (A3)

- Le regole di inferenza di T siano Modus Ponens, Specializzazione e la seguente regola:
- **Abduzione (ABD):**

$$\frac{B, A \rightarrow B}{A}$$

ESEMPIO

- In T si deriva come teorema la formula p(0,0) applicando le seguenti trasformazioni:
  - da Spec. e A2:
  - $\forall X \forall Y(p(X,Y) \rightarrow p(X,s(Y))) \Rightarrow \forall Y\ (p(0,Y) \rightarrow p(0,s(Y)))$ (T1)
- da Spec. e T1:
  - $p(0,0) \rightarrow p(0,s(0))$ (T2)
- applicando ABD a T2 e A6:
  - p(0,0) (T5)

ESEMPIO

- A causa dell'applicazione dell'abduzione, questa teoria **non è corretta**: un'interpretazione che ha come dominio l'insieme dei numeri naturali e associa al simbolo di funzione "s" la funzione successore e al simbolo di predicato "p" la relazione < (minore) è un modello per gli assiomi, ma non per la formula p(0,0).
- **Esempio**
  - sta-male(mario).
  - ∀ X (ha-epatite(X) → sta-male(X)).
- **si conclude**:
  - ha-epatite(mario).

ERRORE !!

ABDUZIONE: ESEMPI

- Assiomi:
  - $\forall$ X (person(X) $\rightarrow$ mortal(X)).
  - mortal(tweety).
- Allora deriviamo: person(tweety).
- Vincoli:
  - $\forall$ X not(person(X) and bird(X)).
- Se aggiungiamo:
  - bird(tweety)
- violiamo i vincoli.

Ragionamento abduttivo usato per diagnosi di guasti

ABDUZIONE: ESEMPI

- Teoria:
  - ruota_traballante:- raggi_rotti.
  - ruota_traballante:- gomma_sgonfia.
  - gomma_sgonfia:- valvola_difettosa.
  - gomma_sgonfia:- forata_camera_aria.
  - gomma_mantiene_aria.
- Vincoli:
  - :- gomma_sgonfia, gomma_mantiene_aria
- Goal
  - ?- ruota_traballante.
- Risposta: yes if raggi_rotti
- Mentre:
  - yes if valvola_difettosa
  - yes if forata_camera_aria
  - non sono accettabili in quanto violano i vincoli.

MONOTONICITÀ

- Un'altra proprietà fondamentale delle teorie del primo ordine è la monotonicità. Una teoria T è **monotona** se l'aggiunta di nuovi assiomi non invalida i teoremi trovati precedentemente.

Proprietà

- Sia Th(T) l'insieme dei teoremi derivabili da T. Allora T è monotona se Th(T) $\subseteq$ Th(T$\cup$H) per qualunque insieme aggiuntivo di assiomi H.
- Esistono regole di inferenza non monotone. Ad esempio la regola nota come Assunzione di Mondo Chiuso ("Closed World Assumption"):

Assunzione di Mondo Chiuso (CWA):

$$\frac{T \not\models A}{\sim A}$$

- se una formula atomica "ground" A non è conseguenza logica di una teoria T, ~A si può considerare un teorema di T. Se alla teoria T si aggiunge l'assioma A, non si può più derivare ~A, da cui segue la non monotonicità del sistema di inferenza.

Agenti logici

- Gli agenti logici applicano inferenze a una base di conoscenza per derivare nuove informazioni.

- Concetti base della logica:
  – sintassi: struttura formale delle sentenze
  – semantica: verità di sentenze rispetto ad interpretazioni/modelli
  – conseguenza logica (entailment): sentenza necessariamente vera data un’altra sentenza
  – inferenza: derivare (sintatticamente) sentenze da altre sentenze
  – correttezza (soundness): la derivazione produce solo sentenze che sono conseguenza logica.
  – completezza (completeness): la derivazione può produrre tutte le conseguenze logiche.

CONSEGUENZA LOGICA (2)

- Una formula F **segue logicamente** (o è conseguenza logica) da un insieme di formule S (e si scrive **S |= F**), se e solo se ogni interpretazione I che è un modello per S, è un modello per F.

Proprietà (*ci torneranno utili più avanti*):

- Se una fbf F segue logicamente da S (S |= F), allora l'insieme S ∪ {~F} è insoddisfacibile.
- Viceversa, se S ∪ {~F} è insoddisfacibile (e S era soddisfacibile), allora F segue logicamente da S.
- Poiché è difficile lavorare a livello semantico (interpretazioni, modelli sono spesso infiniti e/o in numero infinito), allora …
- **… si lavora a livello sintattico**, operando per refutazione

SISTEMI DI REFUTAZIONE

- I sistemi di refutazione si basano su questa proprietà: per dimostrare S |= F supposto S soddisfacibile è sufficiente dimostrare che S∪{~F} è insoddisfacibile.
- Problema interessante:

  Determinare se una formula F segue logicamente da S (ovvero che S ∪ {~F} è insoddisfacibile) **utilizzando solo semplici trasformazioni sintattiche (regole di inferenza)**, possibilmente ripetitive e quindi automatizzabili, e non introducendo concetti quali significato o interpretazione o modello.

→ *Principio di risoluzione (regola di inferenza per logica a clausole)*