Logica in generale

Le logiche sono linguaggi formali per rappresentare l'informazione
in modo tale che si possano trarre delle conclusioni

La sintassi definisce le sentenze che appartengono al linguaggio

La semantica definisce il "significato" delle sentenze;
cioè, definisce il valore di verità di una sentenza in un dato mondo

Per esempio, nel linguaggio dell'aritmetica

$x+2 \geq y$ è una sentenza; $x2+y>$ non è una sentenza

$x+2 \geq y$ è vera se e solo se il numero $x+2$ non è minore del numero y

$x+2 \geq y$ è vera in un mondo dove $x=7,\ y=1$
$x+2 \geq y$ è falsa in un mondo dove $x=0,\ y=6$

Entailment (Implicazione)

Entailment significa che una sentenza *segue da* un' altra (consegue):

$$KB \models \alpha$$

La base di conoscenza KB implica la sentenza α
se e solo se
α è vera in tutti i mondi dove KB è vera

Per esempio, la KB contenente "l'INTER ha vinto" e "il MILAN ha vinto" ne consegue che "ha vinto l'INTER o ha vinto il MILAN"

Per esempio, $x + y = 4$ implica $4 = x + y$

L'implicazione è una relazione fra sentenze (cioè, *sintassi*) che è basata sulla *semantica*

Modelli

I logici tipicamente pensano in termini di modelli, che formalmente sono mondi strutturati rispetto ai quali si può valutare la verità

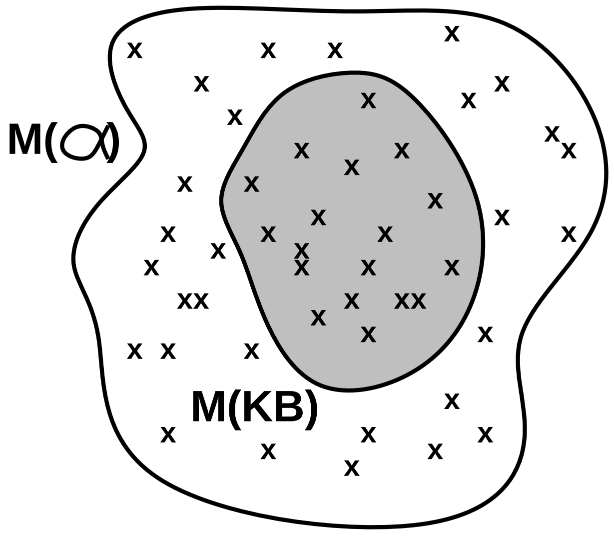
Diciamo che m è un modello di una sentenza α se α è vera in m

$M(\alpha)$ è l'insieme di tutti i modelli di α

Allora $KB \models \alpha$ se e solo se $M(KB) \subseteq M(\alpha)$

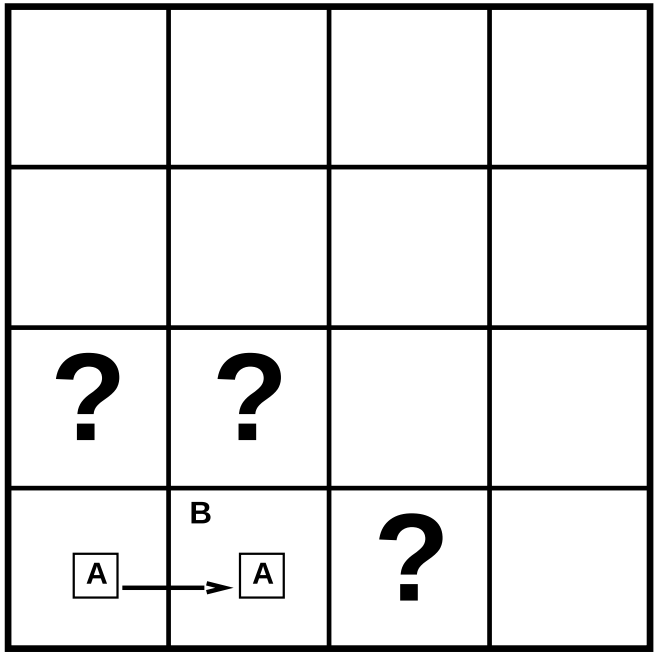
Per esempio, $KB =$ l'INTER ha vinto e il MILAN ha vinto

$\alpha =$ l'INTER ha vinto



Entailment nel mondo dei wumpus

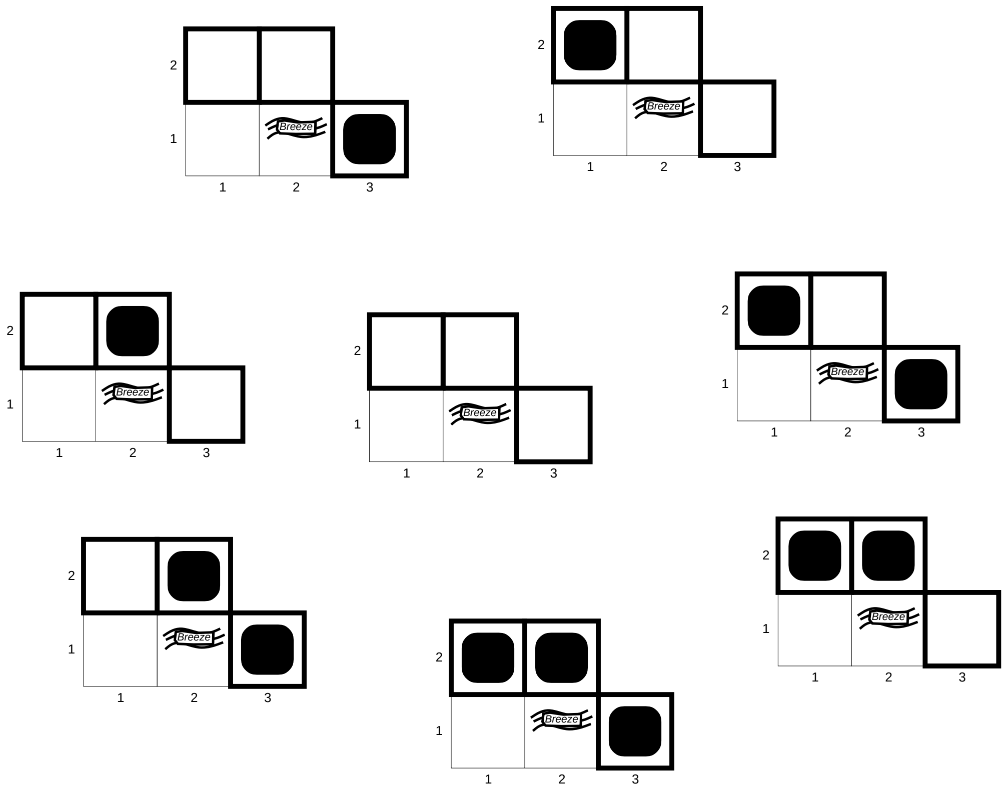
Situazione dopo aver riconosciuto che non c'è nulla in [1,1],
spostamento a destra, brezza in [2,1]



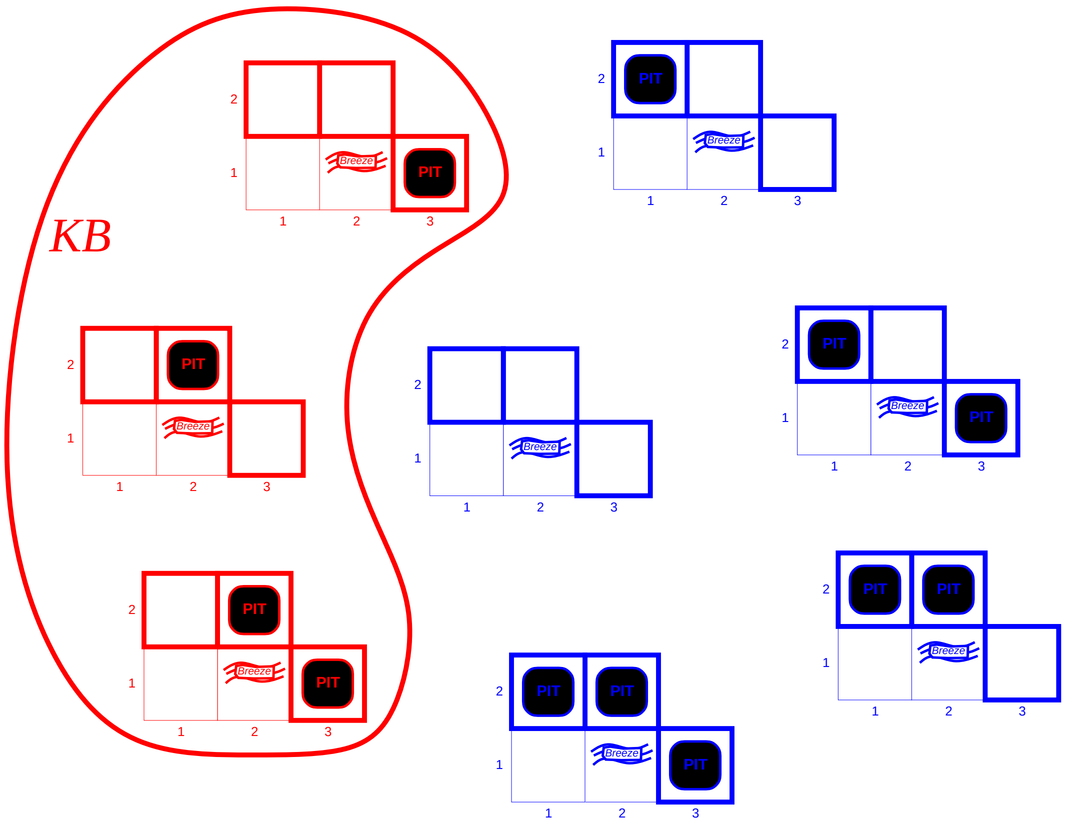
Consideriamo modelli possibili per i "?" assumendo solo trappole

3 scelte Booleane $\Rightarrow$ 8 possibili modelli

Modelli per il mondo dei wumpus

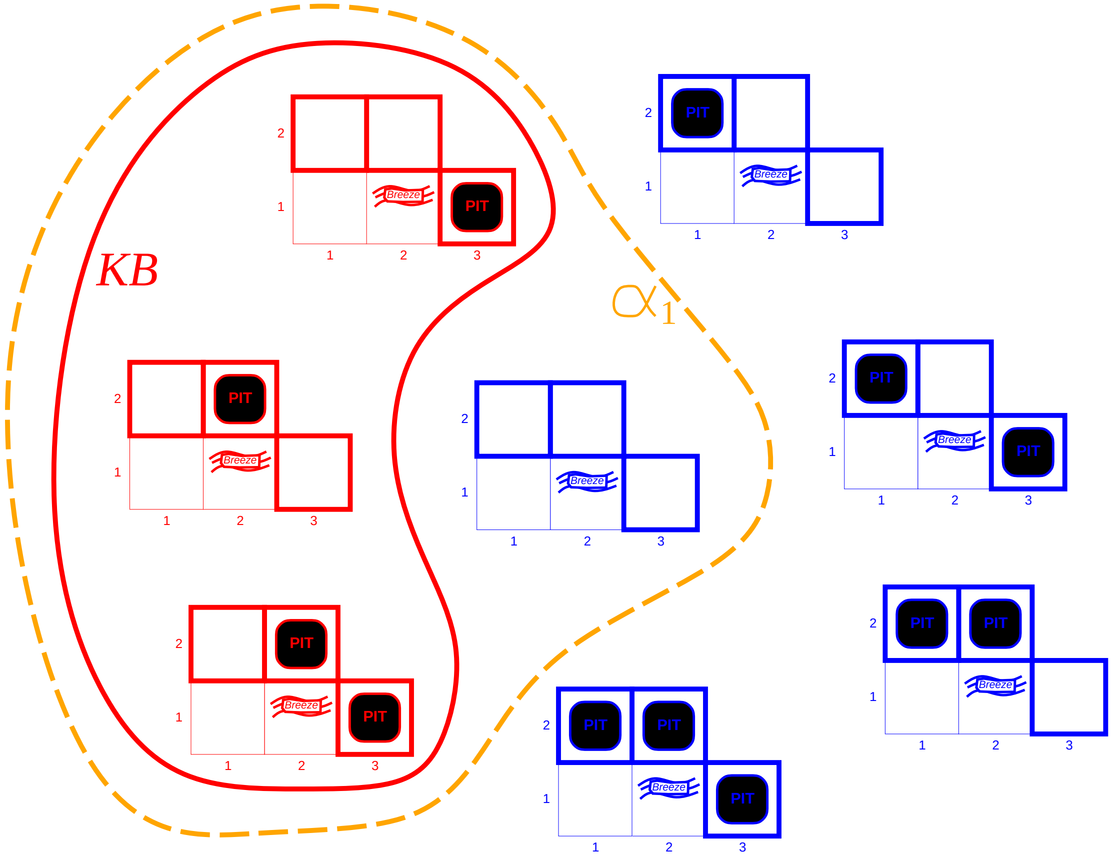


Modelli per il mondo dei wumpus



KB = regole del mondo dei wumpus + osservazioni

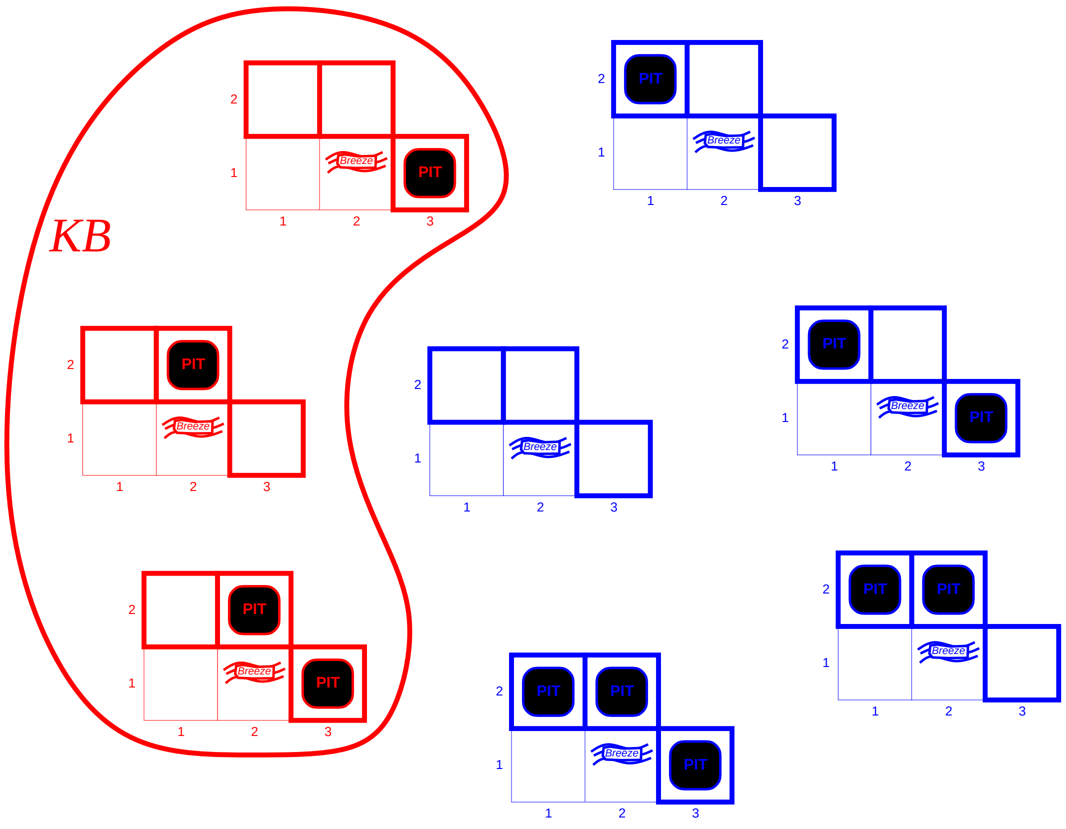
Modelli per il mondo dei wumpus



KB = regole del mondo dei wumpus + osservazioni

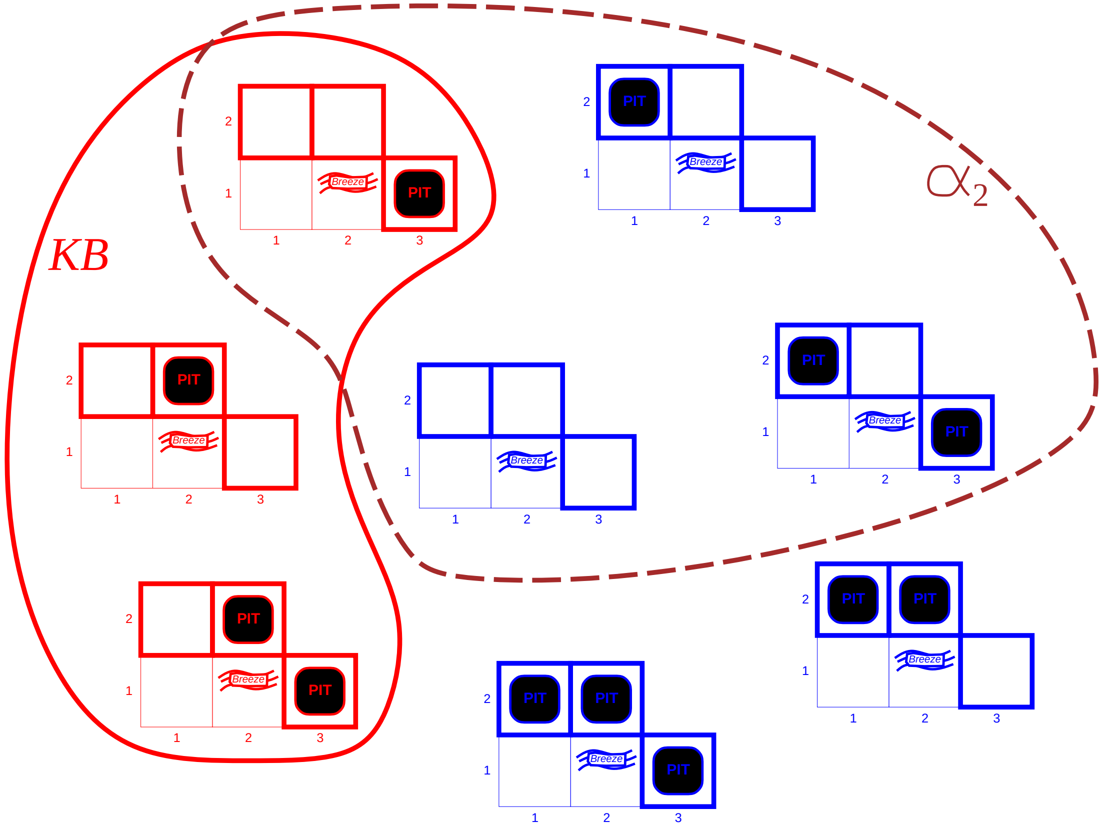
α_1 = "[1,2] è sicuro", $KB \models \alpha_1$, provato via model checking (test sui modelli)

Modelli per il mondo dei wumpus



KB = regole del mondo dei wumpus + osservazioni

Modelli per il mondo dei wumpus



KB = regole del mondo dei wumpus + osservazioni

α_2 = "[2,2] è sicuro", $KB \not\models \alpha_2$

Inferenza

$KB \vdash_i \alpha$ = la sentenza α può essere derivata da KB per mezzo della procedura i

Le conseguenze di KB costituiscono un "pagliaio"; α è un "ago".
Entailment = "ago nel pagliaio"; inferenza = trovarlo

Correttezza (Soundness): i è corretta se
ogni volta che $KB \vdash_i \alpha$, è anche vero che $KB \models \alpha$

Completezza (Completeness): i è completa se
ogni volta che $KB \models \alpha$, è anche vero che $KB \vdash_i \alpha$

Anticipazione: definiremo una logica (logica del primo ordine) che è abbastanza espressiva da essere in grado di esprimere quasi tutto ciò che interessa, e per cui esiste una procedura di inferenza corretta e completa.

In sintesi, la procedura risponde a tutte le domande la cui risposta segue da ciò che è conosciuto dalla KB.

Logica Proposizionale: Sintassi

La logica proposizionale è la più semplice—illustra idee di base

I simboli proposizionali P_1, P_2 etc costituiscono sentenze

Se S è una sentenza, $\neg S$ è una sentenza (negazione)

Se S_1 e S_2 sono sentenze, $S_1 \wedge S_2$ è una sentenza (congiunzione)

Se S_1 e S_2 sono sentenze, $S_1 \vee S_2$ è una sentenza (disgiunzione)

Se S_1 e S_2 sono sentenze, $S_1 \Rightarrow S_2$ è una sentenza (implicazione)

Se S_1 e S_2 sono sentenze, $S_1 \Leftrightarrow S_2$ è una sentenza (bicondizionale)

Logica Proposizionale: Semantica

Ogni modello specifica il valore di verità (vero/falso) per ogni simbolo proposizionale

P.e. $\begin{array}{ccc} P_{1,2} & P_{2,2} & P_{3,1} \\ vero & vero & falso \end{array}$

(Con questi simboli, 8 possibili modelli possono essere enumerti automaticamente.)

Regole per valutare la verità rispetto ad un modello m:

$\neg S$	è vero sse	S	è falso		
$S_1 \wedge S_2$	è vero sse	S_1	è vero and	S_2	è vero
$S_1 \vee S_2$	è vero sse	S_1	è vero or	S_2	è vero
$S_1 \Rightarrow S_2$	è vero sse	S_1	è falso or	S_2	è vero
cioè,	è falso sse	S_1	è vero and	S_2	è falso
$S_1 \Leftrightarrow S_2$	è vero sse	$S_1 \Rightarrow S_2$	è vero and	$S_2 \Rightarrow S_1$	è vero

Un semplice processo ricorsivo valuta una sentenza arbitraria, p.e.,

$\neg P_{1,2} \wedge (P_{2,2} \vee P_{3,1}) = falso \wedge (vero \vee falso) = falso \wedge vero = falso$

Tabelle di verità per i connettivi

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
$falso$	$falso$	$vero$	$falso$	$falso$	$vero$	$vero$
$falso$	$vero$	$vero$	$falso$	$vero$	$vero$	$falso$
$vero$	$falso$	$falso$	$falso$	$vero$	$falso$	$falso$
$vero$	$vero$	$falso$	$vero$	$vero$	$vero$	$vero$

Sentenze nel mondo dei wumpus

Poniamo $P_{i,j}$ essere vero se c'è una trappola in $[i, j]$.
Poniamo $B_{i,j}$ essere vero se c'è brezza in $[i, j]$.

$$\neg P_{1,1}$$
$$\neg B_{1,1}$$
$$B_{2,1}$$

"Le trappole causano brezze in quadrati adiacenti"

Sentenze nel mondo dei wumpus

Poniamo $P_{i,j}$ essere vero se c'è una trappola in $[i, j]$.
Poniamo $B_{i,j}$ essere vero se c'è brezza in $[i, j]$.

$$\begin{aligned} \neg P_{1,1} \\ \neg B_{1,1} \\ B_{2,1} \end{aligned}$$

"Le trappole causano brezze in quadrati adiacenti"

$$\begin{aligned} B_{1,1} &\Leftrightarrow (P_{1,2} \vee P_{2,1}) \\ B_{2,1} &\Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1}) \end{aligned}$$

"Un quadrato ha brezza *se e solo se* c'è una trappola adiacente"

Tabelle di verità per l'inferenza

$B_{1,1}$	$B_{2,1}$	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{2,2}$	$P_{3,1}$	KB	α_1
$falso$	$falso$	$falso$	$falso$	$falso$	$falso$	$falso$	$falso$	$vero$
$falso$	$falso$	$falso$	$falso$	$falso$	$falso$	$vero$	$falso$	$vero$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$falso$	$vero$	$falso$	$falso$	$falso$	$falso$	$falso$	$falso$	$vero$
$falso$	$vero$	$falso$	$falso$	$falso$	$falso$	$vero$	$\underline{vero}$	$\underline{vero}$
$falso$	$vero$	$falso$	$falso$	$falso$	$vero$	$falso$	$\underline{vero}$	$\underline{vero}$
$falso$	$vero$	$falso$	$falso$	$falso$	$vero$	$vero$	$\underline{vero}$	$\underline{vero}$
$falso$	$vero$	$falso$	$falso$	$vero$	$falso$	$falso$	$falso$	$vero$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$vero$	$vero$	$vero$	$vero$	$vero$	$vero$	$vero$	$falso$	$falso$

Inferenza per mezzo di enumerazione

Una enumerazione a scandaglio (Depth-first) di tutti i modelli è corretta e completa

```
function TV-IMPLICA?(KB, α) returns true oppure false

    s ← una lista di simboli proposizionali contenuti in KB e α
    return TV-VERIFICA-TUTTO(KB, α, s, [ ])
--------------------------------------------------------------------------
function TV-VERIFICA-TUTTO(KB, α, s, modello) returns true oppure false
    if VUOTO?(s) then
        if CP-VERO?(KB, modello) then return CP-VERO?(α, modello)
        else return true
    else do
        P ← PRIMO(s); resto ← RESTO(s)
       return TV-VERIFICA-TUTTO(KB, α, resto, ESTENDI(P, true, modello))
and
                TV-VERIFICA-TUTTO(KB, α, resto, ESTENDI(P, false, modello))
```

$O(2^n)$ per n simboli; il problema è co-NP-complete

Equivalenza Logica

Due sentenze sono logicamente equivalenti sse sono vere negli stessi modelli:

$\alpha \equiv \beta$ se e solo se $\alpha \models \beta$ and $\beta \models \alpha$

$$
\begin{aligned}
(\alpha \land \beta) &\equiv (\beta \land \alpha) \quad \text{commutativity of } \land \\
(\alpha \lor \beta) &\equiv (\beta \lor \alpha) \quad \text{commutativity of } \lor \\
((\alpha \land \beta) \land \gamma) &\equiv (\alpha \land (\beta \land \gamma)) \quad \text{associativity of } \land \\
((\alpha \lor \beta) \lor \gamma) &\equiv (\alpha \lor (\beta \lor \gamma)) \quad \text{associativity of } \lor \\
\neg(\neg\alpha) &\equiv \alpha \quad \text{double-negation elimination} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\beta \Rightarrow \neg\alpha) \quad \text{contraposition} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\alpha \lor \beta) \quad \text{implication elimination} \\
(\alpha \Leftrightarrow \beta) &\equiv ((\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)) \quad \text{biconditional elimination} \\
\neg(\alpha \land \beta) &\equiv (\neg\alpha \lor \neg\beta) \quad \text{de Morgan} \\
\neg(\alpha \lor \beta) &\equiv (\neg\alpha \land \neg\beta) \quad \text{de Morgan} \\
(\alpha \land (\beta \lor \gamma)) &\equiv ((\alpha \land \beta) \lor (\alpha \land \gamma)) \quad \text{distributivity of } \land \text{ over } \lor \\
(\alpha \lor (\beta \land \gamma)) &\equiv ((\alpha \lor \beta) \land (\alpha \lor \gamma)) \quad \text{distributivity of } \lor \text{ over } \land
\end{aligned}
$$