logica

- logica sillogistica
  - sintassi
  - dimostrazioni
    - dirette
      - leggi di conversione:
      - sillogismi perfetti:
    - indirette
      - contradditori:
- logica proposizionale
- risoluzione proposizionale

logica sillogistica

sintassi

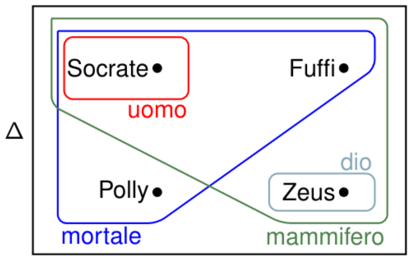
- **A**(x, y): Tutti gli x sono y.
- **E**(x, y): Nessun x è y.
- **I**(x, y): Qualche x è y.
- **O**(x, y): Qualche x non è y.

Figure 1: termini logici

termini non logici:
- Abbiamo un insieme finito (vocabolario) V di termini non logici (e.g. "uomo", "mortale", eccetera) e tale che A, E, I, O non sono in V.

Un modello M = (Δ, ι) per un vocabolario V è dato da:
- Un insieme non vuoto Δ di individui ("dominio del discorso");
- Una funzione ι che associa ogni termine non logico $x \in V$ a un insieme non vuoto $\iota(x) \subseteq \Delta$, $\iota(x) \neq \emptyset$.

- Sia $V = \{\text{uomo}, \text{mortale}, \text{mammifero}, \text{dio}\}$.
- Un possibile modello $\mathfrak{M} = (\Delta, \iota)$ per V può essere costruito come:
  - $\Delta = \{\text{Socrate}, \text{Fuffi}, \text{Polly}, \text{Zeus}\}$;
  - $\iota(\text{uomo}) = \{\text{Socrate}\}$;
  - $\iota(\text{mortale}) = \{\text{Socrate}, \text{Fuffi}, \text{Polly}\}$;
  - $\iota(\text{mammifero}) = \{\text{Socrate}, \text{Fuffi}, \text{Zeus}\}$;
  - $\iota(\text{dio}) = \{\text{Zeus}\}$.



Dato un modello $\mathfrak{M} = (\Delta, \iota)$ e una formula ϕ della nostra logica, diciamo che $\mathfrak{M}$ soddisfa ϕ (e scriviamo $\mathfrak{M} \models \phi$) se ϕ è vera in $\mathfrak{M}$. Più precisamente, per $x, y \in V$, $x \neq y$:

- $\mathfrak{M} \models$ **A**(x, y) se e solo se $\iota(x) \subseteq \iota(y)$ (tutti gli x sono y);
- $\mathfrak{M} \models$ **E**(x, y) se e solo se $\iota(x) \cap \iota(y) = \emptyset$ (nessun x è y);
- $\mathfrak{M} \models$ **I**(x, y) se e solo se $\iota(x) \cap \iota(y) \neq \emptyset$ (qualche x è y);
- $\mathfrak{M} \models$ **O**(x, y) se e solo se $\iota(x) \not\subseteq \iota(y)$ (qualche x non è y).

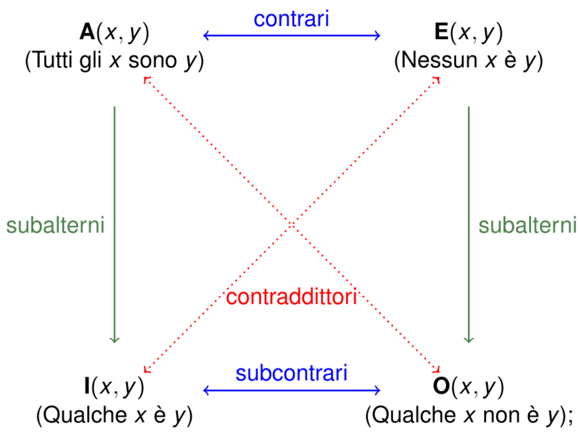
Se Σ è un insieme di formule, scriviamo $\mathfrak{M} \models \Sigma$ se $\mathfrak{M} \models \phi$ per tutti gli $\phi \in \Sigma$.

- $\mathfrak{M} \models$ **A**$(\text{uomo}, \text{mammifero})$, perchè $\iota(\text{uomo}) \subseteq \iota(\text{mammifero})$;
- $\mathfrak{M} \not\models$ **A**$(\text{mortale}, \text{mammifero})$, perchè $\iota(\text{mortale}) \not\subseteq \iota(\text{mammifero})$;

- $\mathfrak{M} \models$ **E**$(\text{dio}, \text{mortale})$, perchè $\iota(\text{mortale}) \cap \iota(\text{dio}) = \emptyset$;
- $\mathfrak{M} \not\models$ **E**$(\text{mortale}, \text{mammifero})$, perchè Fuffi $\in \iota(\text{mortale}) \cap \iota(\text{mammifero})$;

- $\mathfrak{M} \models$ **I**$(\text{mortale}, \text{mammifero})$, perchè Socrate $\in \iota(\text{mortale}) \cap \iota(\text{mammifero})$;
- $\mathfrak{M} \not\models$ **I**$(\text{mortale}, \text{dio})$, perchè $\iota(\text{mortale}) \cap \iota(\text{dio}) = \emptyset$;

- $\mathfrak{M} \models$ **O**$(\text{mammifero}, \text{mortale})$, perchè Zeus $\in \iota(\text{mammifero})$, Zeus $\notin \iota(\text{mortale})$;
- $\mathfrak{M} \not\models$ **O**$(\text{dio}, \text{mammifero})$, perchè $\iota(\text{dio}) \subseteq \iota(\text{mammifero})$.



inferenze quadrato delle opposizioni:
- If A is true, then E is false, I is true, O is false;
- If E is true, then A is false, I is false, O is true;
- If I is true, then E is false, A and O are indeterminate;
- If O is true, then A is false, E and I are indeterminate;
- If A is false, then O is true, E and I are indeterminate;
- If E is false, then I is true, A and O are indeterminate;
- If I is false, then A is false, E is true, O is true;
- If O is false, then A is true, E is false, I is true

dimostrazioni

dirette

leggi di conversione:

- **C1**: $E(x, y) \Rightarrow E(y, x)$
- **C2**: $A(x, y) \Rightarrow I(x, y)$
- **C3**: $I(x, y) \Rightarrow I(y, x)$

sillogismi perfetti:

- **PS1**: $A(y, z) \wedge A(x, y) \Rightarrow A(x, z)$
- **PS2**: $E(y, z) \wedge A(x, y) \Rightarrow E(x, z)$
- **PS3**: $A(y, z) \wedge I(x, y) \Rightarrow I(x, z)$
- **PS4**: $E(y, z) \wedge I(x, y) \Rightarrow O(x, z)$

indirette

contradditori:

- $\overline{A(x, y)} = O(x, y)$
- $\overline{E(x, y)} = I(x, y)$
- $\overline{I(x, y)} = E(x, y)$
- $\overline{O(x, y)} = A(x, y)$
- $\overline{\overline{\phi}} = \phi$

logica proposizionale

Definizione

Una formula P è **soddisfacibile** *se esiste una valutazione della variabili v tale che $v(P) = 1$, cioè se esiste una riga della sua tavola di verità nella quale la formula ha valore 1. In questo caso si dice che la valutazione v soddisfa la formula P e si scrive anche $v \vDash P$.*

Una formula è una **tautologia** *se per ogni valutazione delle variabili v si ha $v(P) = 1$, cioè se in ogni riga della tavola di verità di P la formula ha valore 1. In questo caso si scrive anche $\vDash P$.*

Una formula è una **contraddizione** *o insoddisfacibile se per ogni valutazione delle variabili v si ha $v(P) = 0$, cioè se in ogni riga della tavola di verità di P la formula ha valore 0.*

Forma normale disgiuntiva

Definizione

Un **letterale** è una variabile o la negazione di una variabile. Lo indicheremo in generale con ℓ.

Una formula è in **forma normale disgiuntiva (DNF)** se è della forma

$$\bigvee_{i=1}^{n}\left(\bigwedge_{j=1}^{m_i}\ell_{ij}\right)$$

dove per ogni $i = 1, \dots, n$ e $j = 1, \dots, m_i$ (con $n \geq 1$ e $m_i \geq 1$) gli ℓ_{ij} sono letterali.

Esempio

$(X \wedge \neg Y) \vee (\neg Z \wedge X \wedge Y)$ è una formula in DNF (dove $n = 2$, $m_1 = 2$ e $m_2 = 3$) .

CNF

Definizione

Analogamente diciamo che una formula è in **forma normale congiuntiva** se è una congiunzione di disgiunzioni di letterali.

Esempio

La formula $(X \vee \neg Y) \wedge (\neg Y \vee Z)$ è in CNF.

La formula $\neg Y \wedge (X \vee Z)$ è in CNF.

Le formule $\neg Y \wedge X \wedge Z$ e $\neg Y \vee Z \vee \neg Z$ sono in CNF (e anche in DNF).

Esempio

L'implicazione invece non è commutativa $A \to B \not\equiv B \to A$ e neanche associativa $A \to (B \to C) \not\equiv (A \to B) \to C$.

Contronominale: $A \to B \equiv \neg B \to \neg A$. Questa equivalenza si usa spesso nelle dimostrazioni: se voglio dimostrare che da A segue B posso provare a ipotizzare la negazione di B e concludere che da tale ipotesi segue la negazione di A. Se poi aggiungo che $A \wedge \neg A \equiv \bot$ ottengo le dimostrazioni per assurdo.

Implicazione materiale: le formule $A \to B$ e $\neg A \vee B$ sono logicamente equivalenti:

A	B	$A \to B$	$\neg A \vee B$
0	0	1	1
0	1	1	1
1	0	0	0
1	1	1	1

Esempio

Doppia negazione: $\neg\neg A \equiv A$.

Leggi di De Morgan: Le formule $\neg(A \vee B)$ e $\neg A \wedge \neg B$ sono logicamente equivalenti.

A	B	$A \vee B$	$\neg(A \vee B)$	$\neg A$	$\neg B$	$\neg A \wedge \neg B$
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

Analogamente si ha che $\neg(A \wedge B) \equiv \neg A \vee \neg B$. Inoltre vale

$$\begin{aligned} A \wedge B &\equiv \neg(\neg A \vee \neg B) \\ A \vee B &\equiv \neg(\neg A \wedge \neg B)\,. \end{aligned}$$

Esempio

Assorbimento:

$$\begin{aligned} X \vee (X \wedge Y) &\equiv X \\ X \wedge (X \vee Y) &\equiv X \end{aligned}$$

Legge distributiva: Vale la distributività di $\wedge$ rispetto a $\vee$ e anche il viceversa.

$$\begin{aligned} X \vee (Y \wedge Z) &\equiv (X \vee Y) \wedge (X \vee Z) \\ X \wedge (Y \vee Z) &\equiv (X \wedge Y) \vee (X \wedge Z) \end{aligned}$$

Generalizzando si ha che vale:

$$\begin{aligned} (X_1 \vee X_2) \wedge (Y_1 \vee Y_2) &\equiv (X_1 \wedge Y_1) \vee (X_1 \wedge Y_2) \vee (X_2 \wedge Y_1) \vee (X_2 \wedge Y_2) \\ (X_1 \wedge X_2) \vee (Y_1 \wedge Y_2) &\equiv (X_1 \vee Y_1) \wedge (X_1 \vee Y_2) \wedge (X_2 \vee Y_1) \wedge (X_2 \vee Y_2) \end{aligned}$$

Definizione

Una formula P è una α-**formula** se ha la forma $A \wedge B$ oppure $\neg(A \vee B)$ oppure $\neg(A \to B)$. I ridotti di una α-formula sono definiti dalla seguente tabella:

	ridotti	
$A \wedge B$	A	B
$\neg(A \vee B)$	$\neg A$	$\neg B$
$\neg(A \to B)$	A	$\neg B$

Proposizione

Ogni α-formula è equivalente alla congiunzione dei suoi ridotti.

Definizione

Una formula P è una β-**formula** se ha la forma $A \vee B$ oppure $\neg(A \wedge B)$ oppure $A \to B$. I ridotti di una α-formula sono definiti dalla seguente tabella:

	ridotti	
$A \vee B$	A	B
$\neg(A \wedge B)$	$\neg A$	$\neg B$
$A \to B$	$\neg A$	B

Proposizione

Ogni β-formula è equivalente alla disgiunzione dei suoi ridotti.

Proposizione

Ogni formula P è di uno dei seguenti tipi:

- *P è un letterale;*
- *P è una doppia negazione, cioè $P = \neg\neg Q$;*
- *P è una α-formula;*
- *P è una β-formula.*

Definizione

Una coppia di letterali $X, \neg X$ si dice **complementare**.

Chiaramente una coppia complementare di letterali non è soddisfacibile. In generale vale che:

Proposizione

Un insieme di letterali è soddisfacibile se e solo se non contiene coppie complementari.

Definizione

Un ramo di un tableau è **chiuso** se la foglia contiene una coppia complementare. Un tableau è **chiuso** se ogni ramo è chiuso.

(slides tratte dal corso di B. Gerla)

Definizione

Un **tableau** per una formula P è un albero T i cui nodi sono etichettati con insiemi di sottoformule di P.
Denotiamo con $E(n)$ l'etichetta del nodo n.
L'albero si costruisce per passi successivi.
Al passo 0 abbiamo un albero T_0 formato da un solo nodo con etichetta $\{P\}$.
Se al passo $i-1$ abbiamo costruito un albero T_{i-1}, al passo i costruiamo l'albero T_i guardando le foglie dell'albero T_{i-1}:

- Se nelle foglie ci sono solo letterali, allora la costruzione termina e T_{i-1} sarà l'albero finale.
- supponiamo che nell'etichetta $E(n)$ della foglia n ci sia una formula G che non è un letterale. Allora si possono avere i seguenti casi:
  - Se G è una doppia negazione $G = \neg\neg G_1$, allora l'albero T_i si costruisce aggiungendo un nodo n_1 come successore di n e ponendo
  $$E(n_1) = (E(n) \setminus \{G\}) \cup \{G_1\}.$$
  - Se G è una α formula con ridotti G_1 e G_2, allora l'albero T_i si costruisce aggiungendo un nodo n_1 come successore di n e ponendo
  $$E(n_1) = (E(n) \setminus \{G\}) \cup \{G_1, G_2\}.$$
  - Se G è una β formula con ridotti G_1 e G_2, allora l'albero T_i si costruisce aggiungendo due nod n_1 e n_2 come successori di n e ponendo
  $$E(n_1) = (E(n) \setminus \{G\}) \cup \{G_1\},$$
  $$E(n_2) = (E(n) \setminus \{G\}) \cup \{G_2\}.$$

To convert a propositional formula to conjunctive normal form, perform the following two steps:

1. Push negations into the formula, repeatedly applying De Morgan's Law, until all negations only apply to atoms. You obtain a formula in negation normal form.
   - `¬(p ∨ q)` to `(¬p) ∧ (¬q)`
   - `¬(p ∧ q)` to `(¬p) ∨ (¬q)`
2. Repeatedly apply the distributive law where a disjunction occurs over a conjunction. Once this is not possible anymore, the formula is in CNF.
   - `p ∨ (q ∧ r)` to `(p ∨ q) ∧ (p ∨ r)`

To obtain a formula in disjunctive normal form, simply apply the distribution of `∧` over `∨` in step 2.

risoluzione proposizionale

Definizione

Una **clausola** è una disgiunzione di letterali.

Definizione

La **clausola vuota** (denotata con $\square$) è l'insieme vuoto di letterali.

Semantica delle clausole

Adattando la nozione di valutazione agli insiemi di clausole abbiamo:

Definizione

Sia S un insieme di clausole. Una valutazione è una funzione $v : Var \to \{0, 1\}$. Per definire quando v soddisfa S (in simboli $v \models S$) procediamo nel seguente modo:

- Se $X \in Var$ allora $v \models X$ se $v(X) = 1$ e $v \models \neg X$ se $v(X) = 0$;
- per ogni clausola $C \in S$, con $C = \{L_1, \dots, L_n\}$ si ha $v \models C$ se esiste $i \in \{1, \dots, n\}$ tale che $v \models L_i$;
- $v \models S$ se per ogni $C \in S$ si ha $v \models C$.

Nei casi particolari della clausola vuota e dell'insieme vuoto di clausole abbiamo che

La clausola vuota $\square$ è sempre insoddisfacibile.
Ogni insieme di clausole che contiene $\square$ è insoddisfacibile.
L'insieme vuoto di clausole $\emptyset$ è soddisfatto da ogni interpretazione.

Definizione

Due insiemi di clausole S e S' sono logicamente equivalenti ($S \equiv S'$) se sono soddisfatti dalle stesse valutazioni.
S' è una conseguenza logica di S se ogni valutazione che soddisfa S soddisfa anche S'.

Proposizione

Una clausola è una tautologia se e solo se contiene un letterale e la sua negazione.
Sia S' l'insieme ottenuto da S cancellando una tautologia. Allora $S \equiv S'$.

Esempio

$S = \{\{X, Y, \neg Z\}, \{X, \neg Y\}, \{X, \neg X, Y\}\}$ è logicamente equivalente a $S' = \{\{X, Y, \neg Z\}, \{X, \neg Y\}\}$. Controllare che le formule $(X \vee Y \vee \neg Z) \wedge (X \vee \neg Y) \wedge (X \vee \neg X \vee Y)$ e $(X \vee Y \vee \neg Z) \wedge (X \vee \neg Y)$ sono logicamente equivalenti.

Definizione

Siano C_1 e C_2 due clausole tali che esista un letterale $L \in C_1$ e $\neg L \in C_2$.
Allora il **risolvente** R di C_1 e C_2 (rispetto al letterale L) è la clausola

$$R = (C_1 \setminus \{L\}) \cup (C_2 \setminus \{\neg L\}).$$

Diciamo anche che R si ottiene per **risoluzione** da C_1 e C_2.

Esempio

Se $C_1 = \{\neg X, \neg Y, Z\}$ e $C_2 = \{Y, H, Z\}$ allora $R = \{\neg X, Z, H\}$ è il risolvente di C_1 e C_2 rispetto a Y.

Proposizione: correttezza della risoluzione

Il risolvente R è conseguenza logica della congiunzione $\{C_1, C_2\}$.

Dimostrazione.

Sia v una valutazione tale che $v \models C_1$ e $v \models C_2$. Questo vuol dire che esistono $M \in C_1$ e $N \in C_2$ tali che $v(M) = v(N) = 1$.
Se fosse $M = L$ e $N = \overline{L}$ non potrebbe essere $v(M) = v(N) = 1$, quindi almeno uno tra M e N appartiene a R e quindi R è soddisfacibile. $\square$

Si ha quindi che

$$\{C_1, C_2\} \equiv \{C_1, C_2, R\}.$$

Nota che se $R = \square$ allora si ha $\{C_1, C_2\} \equiv \{C_1, C_2, \square\}$ che è insoddisfacibile e quindi:

Se da C_1 e C_2 ottengo $\square$ tramite risoluzione, allora l'insieme $\{C_1, C_2\}$ è insoddisfacibile.

Definizione

Una clausola C è derivabile **per risoluzione** da un insieme di clausole S se esiste una sequenza $C_1, \dots, C_n$ di clausole tale che $C_n = C$ e per ogni $i = 1, \dots, n-1$ si ha che $C_i \in S$ oppure C_i si ottiene per risoluzione da clausole di S e da qualche C_j con $j < i$.
In questo caso scriviamo

$$S \vdash_R C.$$

Definizione

Una **refutazione** di S è una derivazione della clausola vuota $\square$ da S.
S è refutabile se $S \vdash_R \square$.

Teorema

$S \vdash_R \square$ se e solo se S è insoddisfacibile.

Definizione

Se C e G sono due clausole e $C \subseteq G$ (ma $C \neq G$) allora diciamo che C **sussume** G (o che G **è sussunta** da C).

Proposizione

Sia S' l'insieme ottenuto cancellando da S tutte le clausole G sussunte da altre clausole $C \in S$. Allora $S' \equiv S$.

Procedura di Davis-Putnam

E' un algoritmo che semplifica un insieme finito di clausole al fine di determinare se è soddisfacibile oppure no.

Definizione

Se X è una variabile, si dice che una clausola è X-*esonerata* se non contiene né X né $\neg X$.
Dato un insieme di clausole S, gli X-*risolventi* di S sono tutte le clausole che si ottengono da S facendo la risoluzione rispetto a X e $\neg X$.

Sia S l'insieme di clausole considerato.
Iniziamo con il togliere da S tutte le tautologie e le clausole sussunte.
Poi trasformiamo S con una sequenza di passi.

Procedura di Davis-Putnam

Da S otteniamo un insieme S_1 nel seguente modo:

- Si eliminano da S tutte le tautologie e tutte le clausole sussunte.
- Si sceglie una variabile X (detta il **pivot**) che occorre nella clausola più corta. Nel caso di parità di lunghezza si applica l'ordine alfabetico.
- Si aggiungono a S_1 tutte le clausole X-*esonerate* di S.
- Si aggiungono a S_1 tutti gli X-risolventi di S fatti su clausole che non sono X-esonerate.
- Si rimuovono da S_1 tutte le eventuali tautologie e le clausole sussunte.

Dopo questo primo passo la variabile X non sarà presente in S_1.

Nota che se in S ci sono solo clausole che contengono X o solo clausole che contengono $\neg X$, allora in S_1 tali clausole non saranno presenti.

Per quanto detto finora, S_1 è soddisfacibile se e solo se S è soddisfacibile.

Teorema

Sia S un insieme di clausole nelle variabili $X_1, \dots, X_n$. Allora dopo t passi (con $t \leq n$) l'insieme S_t è costituito solo dalla clausola vuota, oppure è vuoto. Nel primo caso S è insoddisfacibile, nel secondo caso è soddisfacibile.