Logica proposizionale

Sandro Zucchi

2009-10

Definire un linguaggio formale

- Definiamo un linguaggio formale LP (che appartiene a una classe di linguaggi detti *linguaggi della logica proposizionale*).
- Per definire un linguaggio, dobbiamo specificare
  1. quali sono i simboli del linguaggio (il lessico),
  2. quali sono le sequenze grammaticali di simboli del linguaggio (dette anche *frasi* o *formule ben formate*),
  3. qual è la semantica del linguaggio, ovvero quali circostanze rendono vere le frasi del linguaggio (se non ti è chiaro perché specificare un linguaggio comporta il passo 3, leggi la scheda su Linguaggio, significato e verità).

Il linguaggio LP

i simboli

- Un insieme infinito di lettere proposizionali: $p_1 \ \ p_2 \ \ p_3 \ \ p_4 \ \ \dots$
- I connettivi: $\wedge \ \ \vee \ \ \supset \ \ \equiv \ \ \sim$
- Le parentesi: ( )

- (Convenzione: per comodità, negli esempi, useremo p, q, r, s... per le lettere proposizionali).

Linguaggio LP

le formule ben formate

(a) Le lettere proposizionali sono formule ben formate di LP (dette *formule atomiche*).
Se φ e ψ sono formule ben formate di LP, allora:
(b) $\sim \varphi$ è una formula ben formata di LP,
(c) $(\varphi \wedge \psi)$ è una formula ben formata di LP,
(d) $(\varphi \vee \psi)$ è una formula ben formata di LP,
(e) $(\varphi \supset \psi)$ è una formula ben formata di LP,
(f) $(\varphi \equiv \psi)$ è una formula ben formata di LP.
(g) Nient'altro è una formula ben formata di LP.

- (Convenzione: è possibile tralasciare le parentesi, quando non crea ambiguità).

Linguaggio LP

valutazioni

- Una *valutazione* di LP è una funzione ν che assegna un valore di verità (cioè assegna 1, il vero, oppure 0, il falso) alle formule di LP e che soddisfa queste condizioni:
  - (a) se φ è una lettera proposizionale di LP, $\nu(\varphi) \in \{0, 1\}$;
    se φ e ψ sono formule ben formate di LP, allora:
  - (b) $\nu(\sim \varphi) = 1$ se $\nu(\varphi) = 0$, altrimenti $\nu(\sim \varphi) = 0$;
  - (c) $\nu(\varphi \wedge \psi) = 1$ se $\nu(\varphi) = 1$ e $\nu(\psi) = 1$, altrimenti $\nu(\varphi \wedge \psi) = 0$;
  - (d) $\nu(\varphi \vee \psi) = 1$ se non accade che $\nu(\varphi) = 0$ e $\nu(\psi) = 0$, altrimenti $\nu(\varphi \vee \psi) = 0$;
  - (e) $\nu(\varphi \supset \psi) = 1$ se non accade che $\nu(\varphi) = 1$ e $\nu(\psi) = 0$, altrimenti $\nu(\varphi \supset \psi) = 0$;
  - (f) $\nu(\varphi \equiv \psi) = 1$ se $\nu(\varphi) = \nu(\psi)$, altrimenti $\nu(\varphi \equiv \psi) = 0$.

Il linguaggio LP

tavole di verità

Una notazione alternativa per esprimere le condizioni (b)-(f) nella definizione di valutazione è la seguente:

φ	ψ	$(\varphi \wedge \psi)$
1	1	1
1	0	0
0	1	0
0	0	0

φ	ψ	$(\varphi \vee \psi)$
1	1	1
1	0	1
0	1	1
0	0	0

φ	ψ	$(\varphi \supset \psi)$
1	1	1
1	0	0
0	1	1
0	0	1

φ	ψ	$(\varphi \equiv \psi)$
1	1	1
1	0	0
0	1	0
0	0	1

φ	$\sim \varphi$
1	0
0	1

Commenti sulla definizione di valutazione

- La definizione di valutazione ci dice quali sono le condizioni di verità di formule della forma $\ulcorner \sim \varphi \urcorner$, $\ulcorner (\varphi \wedge \psi) \urcorner$, $\ulcorner (\varphi \vee \psi) \urcorner$, $\ulcorner (\varphi \supset \psi) \urcorner$ e $\ulcorner (\varphi \equiv \psi) \urcorner$. Per esempio, $\ulcorner (\sim \varphi) \urcorner$ è vera esattamente in quei casi in cui φ è falsa; $\ulcorner (\varphi \wedge \psi) \urcorner$ è vera esattamente in quei casi in cui φ è vera e ψ è vera; ecc.
- Mentre il valore di verità delle formule non atomiche in una valutazione dipende dal valore di verità di altre formule in quella valutazione, il valore di verità di una lettera proposizionale in una valutazione non dipende in alcun modo dal valore di verità di altre lettere proposizionali in quella valutazione (in quanto ogni assegnamento arbitrario di valori di verità alle lettere proposizionali determina una valutazione).
- (Questa caratteristica delle valutazioni in relazione alle lettere proposizionali sarà importante quando cercheremo di rappresentare gli argomenti formulati in italiano nel linguaggio LP).

Validità in LP

- Un argomento in LP consiste in un insieme di formule (le *premesse*) e una formula (la *conclusione*).
- Un argomento con premesse $\{\varphi_1, \ldots, \varphi_n\}$ e conclusione ψ è *valido* in LP se e solo se non esiste una valutazione che rende $\varphi_1, \ldots, \varphi_n$ vere e ψ falsa in LP.
- Se un argomento è valido in LP diremo anche che le sue premesse *implicano* la sua conclusione in LP.
- In simboli, quando un argomento è valido in LP, scriveremo:

$$\{\varphi_1, \ldots, \varphi_n\} \models_{LP} \psi$$

- Una formula ben formata φ di LP è valida ($\models_{LP} \varphi$) se e solo se ogni valutazione la rende vera.

Deduzione naturale per la logica proposizionale

- Introdurremo ora un *sistema di deduzione naturale* per la logica proposizionale (che chiameremo LP(NAT) per brevità): vale a dire introdurremo delle regole per il linguaggio LP che permettono di *derivare* una conclusione a partire da certe premesse.
- Queste regole sono puramente *sintattiche*, vale a dire non fanno alcun riferimento al significato delle formule che manipolano.
- Tuttavia, la *giustificazione* di queste regole è semantica, in quanto le regole che introdurremo ci consentiranno di derivare la conclusione dalle premesse esattamente nei casi in cui le premesse implicano la conclusione.

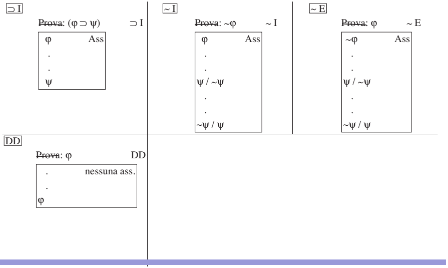
Il sistema LP(NAT)

- Nella formulazione che adottiamo, il sistema LP(NAT) consiste di due tipi di regole: regole di inferenza e regole di inscatolamento e cancellazione.
- Inoltre, per ciascun connettivo avremo due regole: una che ci permette di provare una formula che ha quel connettivo come connettivo principale (regola di introduzione) e l'altra che ci permette di utilizzare una formula che contiene quel connettivo come connettivo principale per provare una formula che non contiene quel connettivo come connettivo principale (regola di eliminazione).
- (Il sistema si basa su Kalish e Montague 1964 e Gettier 1984).

Regole di inferenza di LP(NAT)

$\boxed{\wedge E}$ $\dfrac{(\varphi \wedge \psi)}{\varphi}$ $\dfrac{(\varphi \wedge \psi)}{\psi}$	$\boxed{\wedge I}$ $\dfrac{\begin{array}{c}\varphi\\ \psi\end{array}}{(\varphi \wedge \psi)}$
$\boxed{\vee E}$ $\dfrac{\begin{array}{c}(\varphi \vee \psi)\\ \sim\varphi\end{array}}{\psi}$ $\dfrac{\begin{array}{c}(\varphi \vee \psi)\\ \sim\psi\end{array}}{\varphi}$	$\boxed{\vee I}$ $\dfrac{\varphi}{(\varphi \vee \psi)}$ $\dfrac{\varphi}{(\psi \vee \varphi)}$
$\boxed{\equiv E}$ $\dfrac{(\varphi \equiv \psi)}{(\varphi \supset \psi)}$ $\dfrac{(\varphi \equiv \psi)}{(\psi \supset \varphi)}$	$\boxed{\equiv I}$ $\dfrac{\begin{array}{c}(\varphi \supset \psi)\\ (\psi \supset \varphi)\end{array}}{(\varphi \equiv \psi)}$
$\boxed{\supset E}$ $\dfrac{\begin{array}{c}(\varphi \supset \psi)\\ \varphi\end{array}}{\psi}$	$\boxed{R}$ $\dfrac{\varphi}{\varphi}$

Regole di inscatolamento e cancellazione di LP(NAT)



Derivazione in un sistema di deduzione naturale

Possiamo ora definire la nozione di *derivazione* in un sistema di deduzione naturale:

- una *derivazione* in un sistema di deduzione naturale è una serie consecutiva di righe numerate, costruite secondo le *regole generali per costruire derivazioni*, in cui ogni riga non inscatolata è una premessa (una riga annotata P) o una riga 'prova' cancellata.

Regole generali per costruire derivazioni

valide per tutti i sistemi formali di deduzione naturale

- *Regola delle righe 'Prova':* per qualsiasi formula ben formata φ, è sempre possibile introdurre ⌜Prova:φ⌝ in qualsiasi riga.
- *Regola delle premesse:* qualsiasi formula ben formata φ può essere introdotta come una riga annotata *P*, purché non sia già stata inserita una riga 'Prova'.
- *Regola delle assunzioni:* qualsiasi formula ben formata φ può essere introdotta come una riga annotata *Ass* su una riga che segue *immediatamente* una riga 'Prova'.
- *Regola delle inferenze:* ogni formula ben formata può essere introdotta come una riga se segue da righe *disponibili* attraverso una regola di inferenza, purché sia annotata con il nome della regola di inferenza e con i numeri delle righe da cui è inferita. (Una riga è *disponibile* purché non sia inscatolata e non contenga la parola 'Prova' non cancellata.)
- *Regola per applicare le regole di inscatolamento e cancellazione:* l'*ultima* riga 'Prova' non cancellata può essere cancellata da una regola di inscatolamento e cancellazione, purché la riga 'Prova' sia annotata con il nome della regola (e con il numero di una riga precedente quando sia appropriato) e purché tutte le righe menzionate nella annotazione della regola siano disponibili.

Notazione

valida per tutti i sistemi formali di deduzione naturale

- $\vdash_s \varphi =_{def.}$ Esiste una derivazione nel sistema *s* che non contiene righe annotate *P*, e ⌜~~Prova~~:φ⌝ è l'unica riga non inscatolata.
- $\Gamma \vdash_s \varphi =_{def.}$ Esiste una derivazione nel sistema *s* in cui ogni riga annotata *P* contiene una formula ben formata che è un membro di Γ, e ⌜~~Prova~~:φ⌝ è l'unica riga non inscatolata che non è una premessa.

Completezza

È possibile mostrare, anche se noi non lo faremo qui, che

- ψ è derivabile da $\{\varphi_1, \ldots, \varphi_n\}$ attraverso le regole di LP(NAT) se e solo se l'argomento che ha $\varphi_1, \ldots, \varphi_n$ come premesse e ψ come conclusione è valido in LP.
- In simboli, questo risultato può essere scritto così:

$$\{\varphi_1, \ldots, \varphi_n\} \vdash_{LP(NAT)} \psi \text{ sse } \{\varphi_1, \ldots, \varphi_n\} \models_{LP} \psi.$$

- (Come caso particolare, è possibile mostrare che $\vdash_{LP(NAT)} \psi$ sse $\models_{LP} \psi$).
- Le regole di LP(NAT) per derivare una conclusione da premesse date sono dunque per LP gli schemi di ragionamento validi di cui parlavamo nella nostra discussione iniziale.

Tableaux

- Infine, introduciamo un altro sistema per derivare una conclusione a partire da premesse, il sistema dei *tableaux* (o LP(TAB)).
- Di nuovo, benché si tratti di regole che non fanno alcun riferimento al significato delle formule che manipolano, la giustificazione di queste regole è semantica.
- Vale a dire, come nel caso di LP(NAT) è possibile dimostrare che le regole del sistema dei tableaux ci permettono di derivare una conclusione da un insieme di premesse esattamente nei casi in cui le premesse implicano la conclusione.
- In simboli:

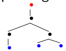
$$\{\varphi_1, \dots, \varphi_n\} \vdash_{LP(TAB)} \psi \text{ sse } \{\varphi_1, \dots, \varphi_n\} \models_{LP} \psi.$$

- (Come caso particolare, è possibile mostrare che $\vdash_{LP(TAB)} \varphi$ sse $\models_{LP} \varphi$).

Definizioni per i tableaux

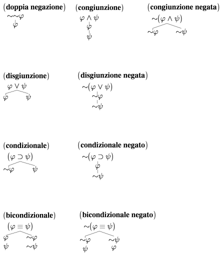
alberi, rami e nodi

- Un tableaux è una struttura ad albero, cioè una struttura di questo genere:



- Le posizioni occupate dai circoletti sono i *nodi* (in un tableau i nodi saranno occupati da formule).
- La posizione occupata dal circoletto rosso è la *radice*, quelle occupate dai circoletti blu sono le *punte*.
- Ogni cammino verso il basso dalla radice a una delle punte è un *ramo*. Per esempio, il cammino contrassegnato in rosso è un ramo:

Regole per costruire tableaux



Convenzioni per costruire i tableaux

- Quando, applicando una regola, si introduce una nuova formula su un ramo, la si deve introdurre sempre *sotto* tutte le altre che compaiono su quel ramo.
- Quando si applica una regola a una formula, si devono introdurre tutti gli elementi richiesti dalla regola su *ogni ramo* su cui la formula compare.
- È meglio applicare le regole che non richiedono di introdurre alcuna diramazione sempre *prima* delle regole che richiedono una diramazione.
- Quando si applica una regola a una formula, si può contrassegnare la formula con $\surd$ (per ricordarsi che quella formula è già stata analizzata).
- Appena un ramo si chiude (nel senso specificato nella prossima slide), è possibile abbandonarlo e contrassegnarlo con il simbolo $\otimes$.
- Per comodità, quando costruiamo un tableau, tutti i segmenti dell'albero che non diramano vengono omessi (le formule vengono semplicemente messe una sopra l'altra).

Terminazione, chiusura, derivabilità

- Un tableau è *terminato* se e solo se ogni regola che poteva essere applicata è stata applicata.
- Un *ramo* di un tableau è *chiuso* se e solo se ci sono formule della forma φ e $\sim \varphi$ su due dei suoi nodi.
- Un *tableau* è *chiuso* se e solo se ogni suo ramo è chiuso; altrimenti è aperto.
- φ è *derivabile in LP(TAB)* da un insieme di formule Σ se e solo se c'è un tableau terminato e chiuso la cui radice consiste nei membri di Σ e della negazione di φ.
- $\{\varphi_1, \ldots, \varphi_n\} \vdash_{LP(TAB)} \psi =_{def.} \varphi$ è derivabile in LP(TAB) dall'insieme di formule $\{\varphi_1, \ldots, \varphi_n\}$.
- $\vdash_{LP(TAB)} \psi =_{def.} \varphi$ è derivabile in LP(TAB) dall'insieme di vuoto $\emptyset$.