Intelligenza Artificiale

Logica proposizionale classica (Parte 1)

Marco Piastra

Introduzione alla logica formale

Parte 1. Preambolo: algebra di Boole, proposizioni, conseguenza logica

Parte 2. Logica proposizionale

Parte 3. Calcolo dei predicati (in forma ridotta)

Testi consigliati

- Magnani, L., Gennari, R.
  Manuale di Logica
  Guerini Scientifica, 1997

- Lolli, G.
  Introduzione alla logica formale
  il Mulino, 1988
- Asperti, A., Ciabattoni, A.
  Logica a informatica
  McGraw-Hill, 1997
- Crossley et al.
  Che cos'è la logica matematica?
  Boringhieri, 1972

Parte 1

Preambolo:

Algebra di Boole
Proposizioni
Conseguenza logica

Concetti essenziali nella logica moderna

- **Astrazione**
  - Rappresentazione delle regole *generali* del ragionamento, a prescindere dalle specifiche situazioni in cui queste sono applicate
- **Esattezza**
  - Le regole del ragionamento devono essere definite in modo esatto, affinché la corretta applicazione possa essere verificata in modo oggettivo
- **Formalismo simbolico**
  - Adozione di un formalismo rappresentativo rigoroso basato su *simboli* e *formule*, composte secondo regole prestabilite
- **Estensionalità**
  - Ai *simboli* ed alle *formule* viene attribuito un significato quantitativo, in relazione a strutture di valori ben definite
  - Opposto, in questo caso, di *intensionalità*: un significato definito in modo qualitativo o puramente descrittivo

Algebra di Boole

- Algebra
  - Un insieme di elementi o *valori*, che chiamiamo V, su cui sono definite delle *operazioni* che soddisfano determinate condizioni.

- Un'algebra di Boole è formata da:
  - Due operazioni binarie $\vee$ e $\wedge$: ($\forall\ x,y,z \in V$)
    - commutatività: $x \vee y = y \vee x$ $\quad x \wedge y = y \wedge x$
    - associatività: $(x \vee y) \vee z = x \vee (y \vee z)$ $\quad (x \wedge y) \wedge z = x \wedge (y \wedge z)$
    - assorbimento: $x \vee (x \wedge y) = x$ $\quad x \wedge (x \vee y) = x$
    - elementi identità $\bot$ e $\top$:

      $x \vee \bot = x$

      $x \wedge \top = x$
  - Un'operazione unaria $\neg$ tale per cui:

    $x \vee \neg x = \top$

    $x \wedge \neg x = \bot$

Caso bivalente: connettivi

- L'insieme V è costituito dai **'valori di verità'**

  $V = \{0, 1\}$ = {FALSO, VERO}

- Le operazioni binarie sono OR ($\vee$) e AND ($\wedge$)

x	y	$x \vee y$
1	1	1
0	1	1
1	0	1
0	0	0

x	y	$x \wedge y$
1	1	1
0	1	0
1	0	0
0	0	0

Le tavole di verità

- L'operazione unaria è il NOT ($\neg$)

x	$\neg x$
1	0
0	1

Algebra proposizionale

- Elementi di base dell'algebra proposizionale (bivalente):
  - un insieme di **proposizioni atomiche** $P = \{a, b, c, d, ...\}$
  - a ciascuna proposizione atomica viene attribuito un **significato**, inteso come **valore di verità**:

    $v : P \rightarrow V$ *cioè* $v : P \rightarrow \{0, 1\}$
- Le **formule** sono espressioni costruite per composizione di proposizioni, connettivi ($\wedge$, $\vee$, $\neg$) e parentesi

  Esempio: $(a \vee b) \wedge \neg c$

  Si usano anche i simboli speciali: T, $\perp$
- Il **significato** delle **formule composite** viene determinato componendo algebricamente il significato delle proposizioni atomiche

vero-funzionalità

a	b	$a \vee b$
$v(a) = 1$	$v(b) = 1$	$v(a \vee b) = 1$
$v(a) = 0$	$v(b) = 1$	$v(a \vee b) = 1$
$v(a) = 1$	$v(b) = 0$	$v(a \vee b) = 1$
$v(a) = 0$	$v(b) = 0$	$v(a \vee b) = 0$

Esempio: AND, ($\forall$ a, b $\in P$)

Interpretazioni e soddisfacimento

- Esempio:
  φ: (a ∨ b) ∧ c
    - ("Giorgio è umano" OR "Silvia è madre di Giorgio") AND "Giorgio è un bipede senza piume"

a	b	c	a ∨ b	(a ∨ b) ∧ c
1	1	1	1	1
0	1	1	1	1
1	0	1	1	1
0	0	1	0	0
1	1	0	1	0
0	1	0	1	0
1	0	0	1	0
0	0	0	0	0

- Un'**interpretazione** v è una assegnazione di significato a tutte le proposizioni atomiche nell'ambito discorsivo *P*
- Una interpretazione **soddisfa** una formula φ sse $v(\varphi) = 1$

Tautologie e contraddizioni

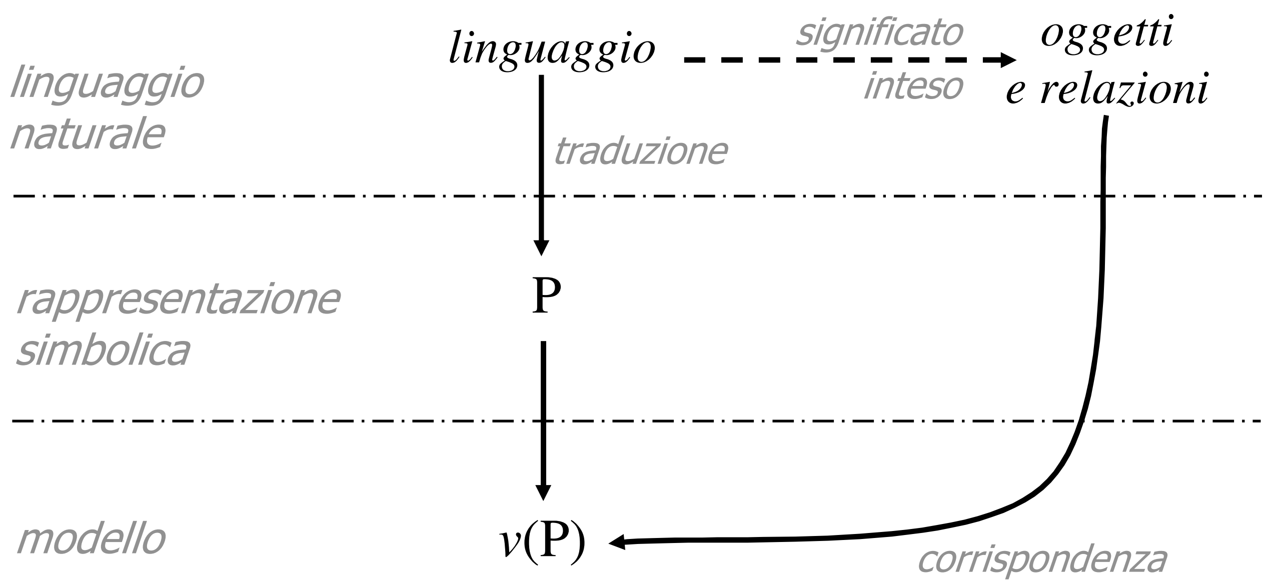
- Una **tautologia** è una formula φ tale per cui $v(\varphi) = 1$ per qualsiasi interpretazione v
  - Esempio: $(\neg a \lor b) \lor (\neg b \lor a)$

a	b	¬a ∨ b	¬b ∨ a	(¬a ∨ b) ∨ (¬b ∨ a)
1	1	1	1	1
0	1	1	0	1
1	0	0	1	1
0	0	1	0	1

- Una **contraddizione** è una formula φ tale per cui $v(\varphi) = 0$ per qualsiasi interpretazione v
  - Esempio: $(a \land \neg a)$

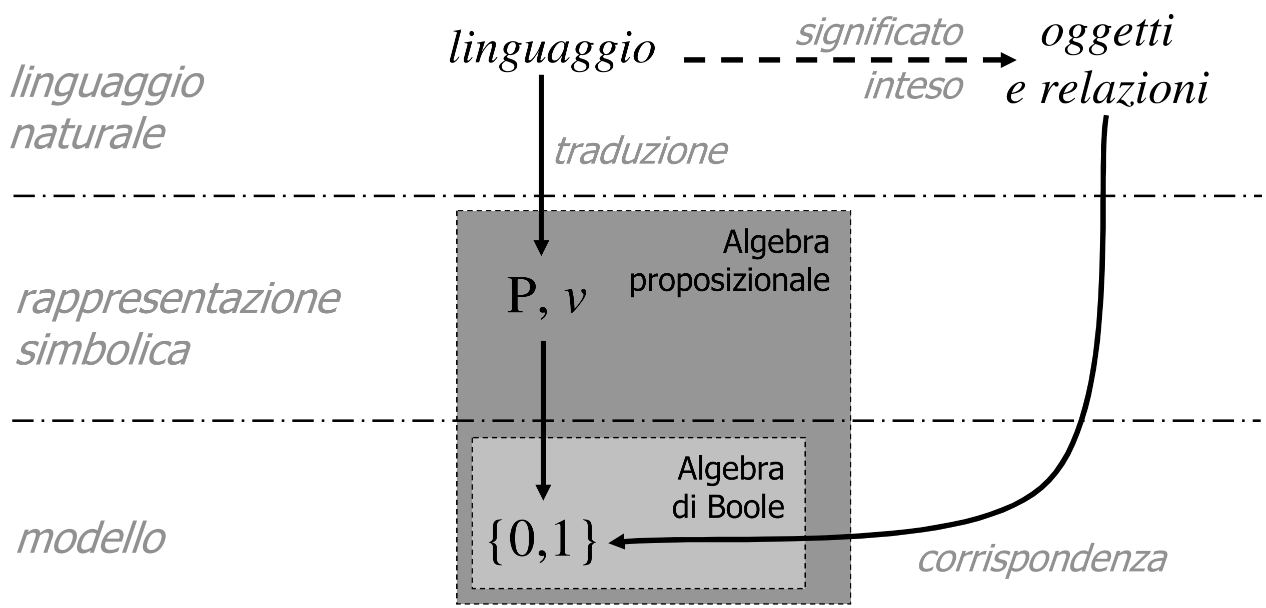
a	¬a	a ∧ ¬a
1	0	0
0	1	0

Rappresentazione formale



- La logica moderna si pone il compito di rappresentare in modo formale la corrispondenza tra *espressioni linguistiche* e *significato inteso*
- La *formalizzazione* è una specie di compressione informativa, con ampia perdita di dettagli (sfumature) e guadagno di precisione

Rappresentazione formale (2)



- L’algebra di Boole (a due valori) definisce operazioni su un insieme di valori semantici
- L’algebra proposizionale stabilisce una corrispondenza (biunivoca) tra formule di un linguaggio ed espressioni algebriche

Formalizzazione del linguaggio (naturale)

- Il processo di traduzione (formalizzazione, astrazione) dal linguaggio naturale alla rappresentazione proposizionale deve seguire delle regole
- Non tutte le espressioni linguisitiche sono 'buone candidate' per questa traduzione
- Nel caso dell'algebra proposizionale, solo le **proposizioni** possono essere rappresentate in *P*
  - *Frasi affermative, di valore compiuto*
  - *Di cui si possa dire esser vere o false*
- La traduzione sacrifica informazione
  - *Ciascuna proposizione è considerata 'atomica', cioè indivisibile*
  - *Il significato dei connettivi è solo quello inteso*
- La rappresentazione non è esente da guai (paradossi)
  - 'Questa proposizione è falsa'

Tutto qui?
Ed il **ragionamento**?

Relazione tra formule

- Premesse:

  φ_1: $\neg(a \wedge \neg b) \vee c$

  NOT ("Giorgio è umano" AND NOT "Silvia è madre di Giorgio")
  OR "Giorgio è un bipede senza piume"

  φ_2: $\neg c \vee b \vee d$

  NOT "Giorgio è un bipede senza piume" OR "Silvia è madre di Giorgio"
  OR "Giorgio è contento"

  φ_3: $d \vee a$

  "Giorgio è contento" OR "Giorgio è umano"

  φ_4: $\neg b$

  NOT "Silvia è madre di Giorgio"

- Affermazione:

  ψ: d

  "Giorgio è contento"

Qual'è il **legame logico** tra le premesse?

E tra le premesse e l'affermazione finale?

Conseguenza logica

- Eseguendo il calcolo diretto:

φ1: ¬(a ∧ ¬b) ∨ c
φ2: ¬c ∨ b ∨ d
φ3: d ∨ a
φ4: ¬b

ψ: d

a	b	c	d	φ_1	φ_2	φ_3	φ_4	ψ
1	1	1	1	1	1	1	0	1
0	1	1	1	1	1	1	0	1
1	0	1	1	1	1	1	1	1
0	0	1	1	1	1	1	1	1
1	1	0	1	1	1	1	0	1
0	1	0	1	1	1	1	0	1
1	0	0	1	0	1	1	1	1
0	0	0	1	1	1	1	1	1
1	1	1	0	1	1	1	0	0
0	1	1	0	1	1	0	0	0
1	0	1	0	1	0	1	1	0
0	0	1	0	1	0	0	1	0
1	1	0	0	1	1	1	0	0
0	1	0	0	1	1	0	0	0
1	0	0	0	0	1	1	1	0
0	0	0	0	1	1	0	1	0

- Tutte le interpretazioni v che soddisfano $\{\varphi_1, \varphi_2, \varphi_3, \varphi_4\}$ soddisfano anche ψ

Attenti alla notazione!

- Relazione di **conseguenza logica** : $\varphi_1, \varphi_2, \varphi_3, \varphi_4 \models \psi$

(Due altri connettivi)

- Implicazione $\rightarrow$ ed equivalenza $\leftrightarrow$

a	b	$a \rightarrow b$
1	1	1
0	1	1
1	0	0
0	0	1

E\` la stessa di $(\neg a \vee b)$

Si legge anche "a implica b"

a	b	$a \leftrightarrow b$
1	1	1
0	1	0
1	0	0
0	0	1

E\` la stessa di $(\neg a \vee b) \wedge (\neg b \vee a)$

Si legge anche "a equivale a b"

- Il problema di prima potrebbe essere riscritto così:

φ_1: $(a \wedge \neg b) \rightarrow c$

φ_2: $c \rightarrow (b \vee d)$

φ_3: $\neg d \rightarrow a$

Regole

φ_4: $\neg b$

ψ: d

Fatti

Conseguenza logica e ragionamento

- La **conseguenza logica** è una *relazione* tra *formule* (o insiemi di formule)
- In generale, in logica si studia la relazione tra le formule di un *sistema logico-simbolico* in cui:
  - il *linguaggio* delle formule è definito con precisione
  - il *significato* delle formule è stabilito in modo non ambiguo
- Le relazioni studiate riguardano la *struttura* dei ragionamenti e non il '*senso*' comune delle formule nell'ambito discorsivo di riferimento (logica ***formale***)
- Quindi, il significato delle formule viene stabilito in riferimento ad una struttura astratta (p. es. {0, 1})
  e non ad una situazione effettiva (p.es. Giorgio e Silvia, bipedi)

La logica in intelligenza artificiale

- Rappresentazione esatta della conoscenza
  - in un *sistema logico-simbolico* :
    - il linguaggio è definito con precisione
    - la semantica è chiara e non ambigua
    - la relazione tra formule descrive il legame logico
  - si possono quindi distinguere le relazioni corrette da quelli fallaci
    - (ammesso di riuscire a formalizzarle)
- Calcolo della relazione logica tra formule
  - il calcolo diretto della conseguenza logica tramite le tavole è scomodo (e non è sempre possibile)
  - occorre trovare tecniche pratiche più efficaci
- Automazione del calcolo
  - se esistessero algoritmi deterministici … (cioè implementabili su macchine reali)
  - si potrebbe pensare di far 'ragionare' le macchine

Complessità

- Il problema della soddisfacibilità di una formula proposizionale qualsiasi è una 'pietra miliare' della teoria della complessità computazionale
  - Indicato anche come SAT, in letteratura
- E\` un problema NP
  - Vale a dire, è risolvibile in tempo polinomiale da una macchina di Turing non deterministica. (Cioè?)
    - Si consideri una formula φ ed un'interpretazione v (che assegna un valore definito a ciascuna lettera proposizionale)
    - Il fatto $v(\varphi) = 1$ può essere accertato in tempo polinomiale (basta applicare le regole per i connettivi)
    - Una macchina di Turing che esplora **simultaneamente** tutte le possibili interpretazioni v di φ terminerebbe in tempo polinomiale
    - Si può far meglio? Si pensa di no (NP $\neq$ P), ma non esiste una prova.
- E\` NP-completo
  - Esiste un'intera classe di problemi NP che possono essere trasformati (in tempo polinomiale) nel problema SAT

Complessità (2)

- Verosimilmente (se NP $\neq$ P)
  - Il problema SAT ha complessità esponenziale, $O(2^n)$
  - (In caso contrario, l'intera classe di problemi NP-completi sarebbe risolvibile in tempo polinomiale)
- Il problema SAT ammette i 'colpi di fortuna'
  - Basterebbe trovare **una** interpretazione che soddisfa la formula ...
  - Algoritmi effettivi per la logica proposizionale (e non)
    - Si basano sulla riduzione del potere espressivo del formalismo, trasformando SAT in una versione ridotta
    - Cercano di essere 'fortunati', basandosi sulla struttura della rappresentazione simbolica per identificare la soluzione in tempi rapidi "nei casi di interesse pratico"
- Attenzione all'impostazione del problema: $\neg$SAT
    - Provare che una formula φ non è soddisfacibile non ammette 'colpi di fortuna'
    - Occorre provare **tutte** le interpretazioni v