Intelligenza Artificiale

Breve introduzione alla logica classica (Parte 2)

Marco Piastra

Introduzione alla logica formale

Parte 1. Preambolo: l'algebra di Boole e la logica

Parte 2. Logica proposizionale

Parte 3. Logica predicativa del primo ordine

Parte 2

Logica proposizionale

Logica proposizionale - Linguaggio

- Un linguaggio proposizionale $\mathscr{L}_P$ contiene:
  - un insieme non vuoto di **lettere proposizionali**: a, b, c, ...
  - due *connettivi* principali: $\neg$, $\rightarrow$
  - due simboli ausiliari: (,) (le parentesi)
  - tre *connettivi* derivati: $\wedge$, $\vee$, $\leftrightarrow$
  - un insieme di regole sintattiche o **regole di buona formazione**
    - (le formule sintatticamente corrette si dicono **formule ben formate** - ***fbf***)
- Regole di rappresentazione:
  - le lettere proposizionali rappresentano *proposizioni*, cioè frasi affermative in linguaggio naturale
  - i *connettivi* rappresentano relazioni tra *proposizioni*:
    - negazione: $\neg$ (non a)
    - implicazione: $\rightarrow$ (se a allora b)
    - congiunzione: $\wedge$ (a e b)
    - disgiunzione: $\vee$ (a o b)
    - equivalenza: $\leftrightarrow$ (a equivale a b)

LP - Regole semantiche

- Una *interpretazione* di $\mathscr{L}_P$ è una funzione
  $v : \mathrm{Fbf}(\mathscr{L}_P) \to \{0, 1\}$
  - $v(\neg\varphi) = 1$ sse $v(\varphi) = 0$
  - $v(\varphi \wedge \psi) = 1$ sse $v(\varphi) = 1$ e $v(\psi) = 1$
  - $v(\varphi \vee \psi) = 1$ sse $v(\varphi) = 1$ o $v(\psi) = 1$
  - $v(\varphi \to \psi) = 1$ sse non $v(\varphi) = 1$ e $v(\psi) = 0$
  - $v(\varphi \leftrightarrow \psi) = 1$ sse $v(\varphi) = v(\psi)$

- La funzione v rispetta le stesse regole algebriche viste in precedenza (le tavole di verità)
  – Notare i connettivi derivati $\to$ (implicazione) e $\leftrightarrow$ (equivalenza)

A	B	$A \to B$
1	1	1
0	1	1
1	0	0
0	0	1

E\` la stessa di $(\neg A \vee B)$

A	B	$A \leftrightarrow B$
1	1	1
0	1	0
1	0	0
0	0	1

E\` la stessa di $(\neg A \vee B) \wedge (\neg B \vee A)$

LP - Modelli e soddisfacibilità

- Data una fbf φ ed una interpretazione v tale per cui
  $v(\varphi) = 1$
- Si dice che:
  - v **soddisfa** φ
  - v è un **modello** di φ
- La definizione è facilmente estesa agli insiemi di fbf
  $\Gamma = \{\varphi_1, \dots \varphi_n\}$
- Una fbf è una tautologia (o una fbf **valida**) se è soddisfatta da qualsiasi interpretazione
- Una fbf ψ è una contraddizione se non ha un modello
- Una fbf ψ è una ***conseguenza logica*** di un insieme di fbf Γ sse qualisiasi modello di Γ è anche modello di ψ
  - si scrive anche:
    $\Gamma \models \psi$

LP – Derivazione

- Una regola di **derivazione** (anche regola di **inferenza**) permette di *derivare* fbf da altre fbf
- In logica proposizionale si ha una sola regola di derivazione
  - *modus ponens* (*mp*):
    $$\frac{\varphi \to \psi \quad \varphi}{\psi}$$
  - si può scrivere anche così:
    $\varphi \to \psi, \varphi \vdash \psi$ (da $\varphi \to \psi$ e φ è *derivabile* ψ)
- Esempio di applicazione:
  - dalle due formule
    $(\neg a \to b) \to (c \to \neg d)$
    $(\neg a \to b)$
  - si può derivare
    $(c \to \neg d)$

Una regola di derivazione è di tipo *sintattico* in quanto opera sulla struttura delle fbf

LP - Assiomi

- Gli assiomi di un sistema logico esprimono *leggi logiche* di validità generale (nel sistema stesso)
- In logica proposizionale si usano degli *schemi di assioma* :

  Ax1 $\varphi \rightarrow (\psi \rightarrow \varphi)$

  Ax2 $(\varphi \rightarrow (\psi \rightarrow \chi)) \rightarrow ((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \chi))$

  Ax3 $(\neg\varphi \rightarrow \neg\psi) \rightarrow (\psi \rightarrow \varphi)$

  – ogni istanziazione è un assioma

- Esempi

  $a \rightarrow (a \rightarrow a)$ [Ax1, φ/a, ψ/a]

  $(\neg(b \vee c) \rightarrow \neg d) \rightarrow (d \rightarrow (b \vee c))$ [Ax3, φ/(b $\vee$ c), ψ/d]

  – Notare che ogni istanziazione è anche una fbf *valida* (o *tautologia*)

LP - Derivazioni

- Una *dimostrazione* (o *derivazione*) di una fbf ψ a partire da un insieme di fbf $\Gamma = \{\varphi_1, \dots \varphi_n\}$ è una successione *finita* di passi $<\alpha_1, \dots, \alpha_n>$
  - per ogni passo α_i si ha che:
    $\alpha_i \in$ istanza($\mathrm{Ax}n$) oppure
    $\alpha_i \in \Gamma$ oppure
    α_i è ottenibile dalle fbf precedenti tramite *modus ponens*
  - $\alpha_n = \psi$
  - in tal caso si scrive $\Gamma \vdash \psi$

- Vale il *teorema di deduzione* (*ded*)

  $\Gamma \cup \{\varphi\} \vdash \psi$ sse $\Gamma \vdash (\varphi \rightarrow \psi)$

LP - Derivazioni, esempio 1

- "Ex absurdo sequitur quodlibet":

  $\vdash \varphi \rightarrow (\neg\varphi \rightarrow \psi)$ (ovvero $\varphi, \neg\varphi \vdash \psi$)

  1: $\varphi, \neg\varphi \vdash \neg\varphi \rightarrow (\neg\psi \rightarrow \neg\varphi)$ (Ax1)
  2: $\varphi, \neg\varphi \vdash \neg\varphi$
  3: $\varphi, \neg\varphi \vdash \neg\psi \rightarrow \neg\varphi$ (*mp* 1,2)
  4: $\varphi, \neg\varphi \vdash (\neg\psi \rightarrow \neg\varphi) \rightarrow (\varphi \rightarrow \psi)$ (Ax3)
  5: $\varphi, \neg\varphi \vdash \varphi \rightarrow \psi$ (*mp* 4,3)
  6: $\varphi, \neg\varphi \vdash \varphi$
  7: $\varphi, \neg\varphi \vdash \psi$ (*mp* 5,6)
  8: $\vdash \varphi \rightarrow (\neg\varphi \rightarrow \psi)$

LP - Derivazioni, esempio 2

- Affermazione implica doppia negazione

  $\vdash \varphi \rightarrow \neg\neg\varphi$

1: $\neg\neg\varphi \vdash \neg\neg\neg\neg\varphi \rightarrow \neg\neg\varphi$	(Ax1, *ded*)
2: $\neg\neg\varphi \vdash (\neg\neg\neg\neg\varphi \rightarrow \neg\neg\varphi) \rightarrow (\neg\varphi \rightarrow \neg\neg\neg\varphi)$	(Ax3)
3: $\neg\neg\varphi \vdash \neg\varphi \rightarrow \neg\neg\neg\varphi$	(*mp* 2,1)
4: $\neg\neg\varphi \vdash (\neg\varphi \rightarrow \neg\neg\neg\varphi) \rightarrow (\neg\neg\varphi \rightarrow \varphi)$	(Ax3)
5: $\neg\neg\varphi \vdash \neg\neg\varphi \rightarrow \varphi$	(*mp* 4,3)
6: $\neg\neg\varphi \vdash \neg\neg\varphi$	
7: $\neg\neg\varphi \vdash \varphi$	(*mp* 5,6)
8: $\vdash \varphi \rightarrow \neg\neg\varphi$	

LP - Correttezza e completezza

- *Correttezza*
  - le fbf φ *derivabili* da un insieme di fbf Γ sono una *conseguenza logica* di Γ (sono soddisfatte dai modelli di Γ)

    $\Gamma \vdash \varphi \;\Rightarrow\; \Gamma \models \varphi$
  - tutte le fbf *derivabili* dagli assiomi $\mathrm{Ax}n$ assiomi sono *valide* (cioè sono *tautologie*)

    $\vdash \varphi \;\Rightarrow\; \models \varphi$

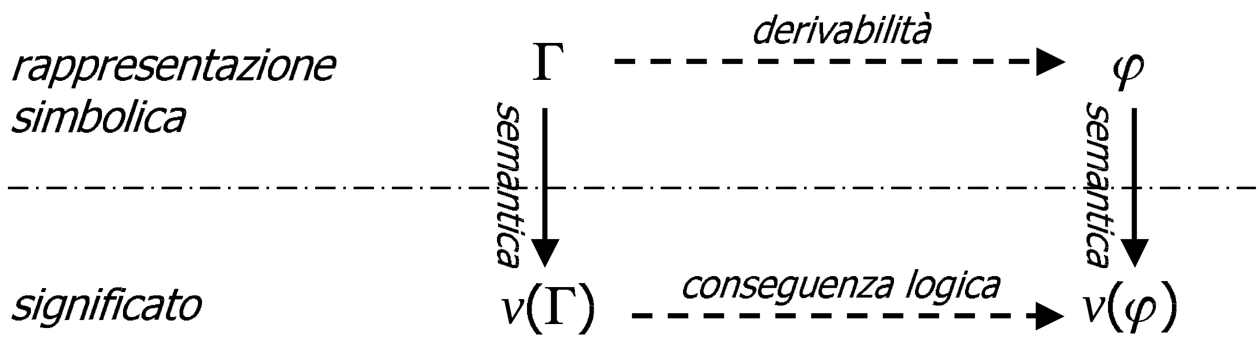
- *Completezza*
  - le *conseguenze logiche* di Γ sono le fbf φ *derivabili*

    $\Gamma \vdash \varphi \;\Leftrightarrow\; \Gamma \models \varphi$
  - le fbf *valide* sono le fbf *derivabili* dagli assiomi $\mathrm{Ax}n$

    $\vdash \varphi \;\Leftrightarrow\; \models \varphi$

LP - Calcolo simbolico

- Per la proprietà di *completezza*, la *derivazione* simbolica è rappresentativa delle relazioni tra i significati



- Nel caso della logica proposizionale, la relazione di *conseguenza logica* può essere determinata in modo diretto
- In molti altri casi, la derivazione simbolica è l'unica possibilità

LP - Decidibilità

- Un sistema logico è detto *decidibile* se esiste un algoritmo di validità generale per stabilire se

  $\Gamma \vdash \varphi$

- La logica proposizionale è senz'altro decidibile
  - alla peggio, si provano tutte le 2^n possibili interpretazioni per stabilire se $\Gamma \models \varphi$

- Il procedimento di *derivazione* non è un algoritmo deterministico
  - ad ogni passo occorre scegliere 'la mossa giusta'
  - si tratta di una tecnica per la prova manuale

LP - Altre forme di derivazione

- Il metodo assiomatico per la derivazione di formule
  - prevede un insieme di operazioni assai complesso
    - modus ponens, sostituzione, introduzione di assiomi, uso di teoremi dimostrati in precedenza, etc.
    - è più adatto al calcolo manuale che non a quello automatico
- Inferenza per **risoluzione**
  - a partire da due formule $\varphi \vee A$ e $\neg A \vee \psi$ si può derivare $\varphi \vee \psi$ *risolvente*
    - A è una proposizione qualsiasi, φ e ψ sono formule qualsiasi
  - in quanto $\varphi \vee A, \neg A \vee \psi \models \varphi \vee \psi$

Si può verificare direttamente

φ	ψ	A	$\varphi \vee A$	$\neg A \vee \psi$	$\varphi \vee \psi$
1	1	1	1	1	1
0	1	1	1	1	1
1	0	1	1	0	1
0	0	1	1	0	0
1	1	0	1	1	1
0	1	0	0	1	1
1	0	0	1	1	1
0	0	0	0	1	0

LP - Forma a clausole

- Una **clausola** (*clause*) è una formula in cui si usano solo ¬ e ∨
  - Esempio: a ∨ ¬b ∨ c ∨ ¬d
  - Un singolo letterale in forma positiva (a) o negativa (¬a) è un **atomo**
- Tutte le formule di $\mathscr{L}_P$ possono essere tradotte in **forma normale congiuntiva**
  - cioè possono essere espresse come *congiunzione di clausole*
  - (a ∧ ¬b) ∨ (¬c ∨ d) equivale a (a ∨¬c) ∧ (¬b ∨ ¬c ∨ d)
  - si scrive anche {(a,¬c), (¬b,¬c, d)}
- Il *modus ponens* è un caso particolare di *risoluzione*
  - a → b, a ⊢ b può essere riscritto come ¬a ∨ b, a ⊢ b
  - cioè {(¬a, b), (a)} ⊢ {(b)}

LP - Derivazione senza assiomi

- L'idea di base è
  - tradurre le premesse nella forma a *clausole*
  - applicare tutte le *risoluzioni* possibili
  - derivando così nuove formule
- Vantaggi:
  - esiste un'unica operazione di derivazione (la risoluzione)
  - può essere applicata in modo esaustivo
  - non necessita di *assiomi logici*
- Purtroppo:
  - questo metodo è *corretto*
  - ma non è *completo*
    - non è possibile derivare tutte le conseguenze logiche di un insieme di premesse

LP – Refutazione e risoluzione

- Conseguenza logica e insoddisfacibilità
  - se $\Gamma \models \varphi$ allora $\Gamma \cup \{\neg\varphi\}$ è **insoddisfacibile**

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

- Clausola vuota come testimone di insoddisfacibilità
  - da $\{(\neg a), (a)\}$ si deriva per risoluzione { } (i.e. una contraddizione)
- Risoluzione per refutazione
  - dovendo dimostrare $\Gamma \models \varphi$
  - si parte da $\{\Gamma, \neg\varphi\}$
  - e si cerca di derivare { } (insoddisfacibilità)

LP - Risoluzione come algoritmo

- Il metodo della refutazione e risoluzione
  - è corretto
  - è anche completo (si può derivare qualsiasi conseguenza logica)
- Come algoritmo
  - è sempre terminante (nel caso proposizionale)
  - infatti, dato un generico problema $\{\Gamma, \neg\varphi\}$
    - si deriva la clausola vuota { }
    - oppure l'algoritmo si arresta quando non sono possibili ulteriori risoluzioni
  - ha una complessità esponenziale $O(2^n)$ (dove *n* è il numero degli *atomi*)

LP - Clausole di Horn

- In una clausola di Horn si ha al massimo <u>un</u> atomo in forma positiva
- Tre tipi di clausole di Horn:
  - atomi singoli (o **fatti**): a, ¬b, c
  - implicazioni (o **regole**): $(a \wedge b) \rightarrow c$ cioè $\neg a \vee \neg b \vee c$
  - obiettivi o **goal**: $(c \wedge d)$, la cui forma negata è $\neg c \vee \neg d$
- Tecnica generale
  - si esprimono le premesse come *fatti* e *regole*
  - si definisce il risultato atteso ("Giorgio è contento"?) come *goal*
  - si applica la tecnica della *risoluzione*
- Limitazioni e vantaggi
  - non tutte le derivazioni possibili sono esprimibili come clausole di Horn
    - ma molti problemi pratici lo sono
  - esiste un metodo di risoluzione a complessità lineare O(*n*)