Logica proposizionale e algebra Booleana

Vittorio Zaccaria

October 26, 2018

Outline

Introduzione alla logica proposizionale

Motivazione

- Le istruzioni impartite al calcolatore spesso sono condizionali ovvero valide solo se alcune premesse sono valide.
- Ad esempio:

  Se l'utente ha inserito un numero di carta di credito corretto ed un codice di controllo di 3 cifre allora autorizza la transazione

  Se l'anno corrente è bisestile allora stampa "anno bisestile" al terminale altrimenti stampa "anno non bisestile"

Motivazione

Sappiamo che tali istruzioni possono essere date attraverso l'uso di un costrutto `if-then-else` che è costituito da tre parti:

Premessa ("if")	Conclusione ("then")	Alternativa ("else")
carta corretta e CCV inserito	autorizza la transazione	
anno corrente è bisestile	stampa anno bisestile	stampa anno non bisestile

La premessa è anche detta proposizione o condizione, ovvero un'affermazione sulla realtà che può essere vera oppure falsa.

Logica

- La logica è una branca della scienza matematica che studia:
    - come è possibile definire in maniera non ambigua le condizioni (sintassi)
    - come è possibile prevederne/calcolarne il valore di verità (semantica)
- È praticamente un linguaggio

Sintassi: proposizioni semplici

- Una proposizione semplice è una affermazione sulla realtà che può essere vera oppure falsa e che non può essere suddivisa in condizioni più semplici, ad esempio:
  - "Sono miliardario"
  - "2+2=4"
  - "2+2=5"

Sintassi: proposizioni complesse

Chiamate formule, sono costruite a partire da connettivi logici vero-funzionali[1, 2] che combinano espressioni più semplici, ad esempio:

Esempio	Nome connettivo
2+2=4 $\wedge$ 3+3=6	DISGIUNZIONE
2+2=4 $\vee$ 3+3=6	CONGIUNZIONE
$\neg$ 2+2=5	NEGAZIONE
x>2 $\rightarrow$ 2x>4	IMPLICAZIONE
x>3 $\oplus$ x<1	CONGIUZIONE ESCLUSIVA

[1] La verità della proposizione dipende solo e soltanto dalla verità delle sotto-proposizioni.

[2] A livello di precedenza degli operatori, $\neg$ ha precedenza maggiore di tutti, seguito da $\wedge$, $\vee$ e $\rightarrow$ esattamente in questo ordine.

Sintassi: Proposizioni complesse

- Alcune proposizioni sono solo apparentemente semplici
- Ad esempio, la proposizione "L'anno X è bisestile" equivale in italiano a
  se X è divisibile per 100, deve essere divisibile anche per 400, altrimenti deve essere divisibile per 4
- Più formalmente:
  (X è divisibile per 100) e (X è divisibile per 400) o (X non è divisibile per 100) e (X è divisibile per 4)
- Usando i connettivi logici
  (X è divisibile per 100) $\wedge$ *(X è divisibile per 400)* $\vee$ *(X non è divisibile per 100)* $\wedge$ *(X è divisibile per 4)*

Semantica di una proposizione

Semantica di una proposizione semplice

La semantica di una proposizione semplice è il suo valore di verità, che può essere noto:

- “Sono miliardario” = F (falso)
- “2+2=4” = V (vero)
- “2+2=5” = F

oppure dipendente da altri parametri:

- “Oggi piove” (dipende da quando interpretiamo questa proposizione)
- X è divisibile per 100 (dipende dal valore della variabile X)

Nell’ultimo caso, le proposizioni sono chiamate predicati.

Semantica di una proposizione complessa

Vi sono 3 modi per ricavare il valore di verità di una proposizione complessa

- Tramite tabelle della verità
- Tramite l'algebra Booleana (rappresenta una scorciatoia alle tabelle della verità)
- (*Opzionale*) Tramite deduzione naturale (metodo molto più potente delle tabelle della verità)

Tabelle della verità

La semantica di una proposizione complessa è data interamente dai valori di verità delle proposizioni semplici che la compongono e dalla tabelle della verità connettivi utilizzati. Se A e B sono due sotto-proposizioni:

A	B	$\neg A$	$A \wedge B$	$A \vee B$	$A \rightarrow B$	$A \leftrightarrow B$
F	F	V	F	F	V	V
F	V	V	F	V	V	F
V	F	F	F	V	F	F
V	V	F	V	V	V	V

Supponiamo di avere la seguente proposizione complessa:

$$(P \wedge Q) \rightarrow \neg R \tag{1}$$

Non conoscendo i valori delle tre proposizioni più semplici, siamo costretti a considerare tutti i possibili casi:

			①	③	①	④	②	①
P	Q	R	(P	$\wedge$	Q)	$\rightarrow$	$\neg$	R
F	F	F	F	F	F	V	V	F
F	F	V	F	F	F	V	F	V
F	V	F	F	F	V	V	V	F
F	V	V	F	F	V	V	F	V
V	F	F	V	F	F	V	V	F
V	F	V	V	F	F	V	F	V
V	V	F	V	V	V	V	V	F
V	V	V	V	V	V	F	F	V

Tabelle della verità

Riprendiamo la proposizione complessa:

(X è divisibile per 100) $\wedge$ (X è divisibile per 400) $\vee$ (X non è divisibile per 100) $\wedge$ (X è divisibile per 4)

e riscriviamola in maniera compatta con:

- sotto-proposizione d100 = X è divisibile per 100
- sotto-proposizione d400 = X è divisibile per 400
- sotto-proposizione d4 = X è divisibile per 4

Tabelle della verità

Otteniamo:

$$(d100 \land d400) \lor (\neg d100 \land d4)$$

Non conoscendo i valori di X, siamo costretti a considerare tutti i possibili casi[3]:

			①	③	①	④	②	①	③	①
d100	d400	d4	(d100	$\land$	d400)	$\lor$	($\neg$	d100	$\land$	d4)
F	F	F	F	F	F	F	V	F	F	F
F	F	V	F	F	F	V	V	F	V	V
F**	V	F**	F	F	V	F	V	F	F	F
F**	V	V	F	F	V	V	V	F	V	V
V	F	F**	V	F	F	F	F	V	F	F
V	F	V	V	F	F	F	F	V	F	V
V	V	F**	V	V	V	V	F	V	F	F
V	V	V	V	V	V	V	F	V	F	V

[3] Alcune delle combinazioni non potranno mai verificarsi nella realtà (e.g., se un anno è divisibile per 400 lo sarà anche per 100). Tuttavia, le tabelle della verità considerano le sotto-proposizioni come "scatole chiuse" senza proprietà matematiche.

Algebra Booleana

A. DeMorgan (1806-1871) e G. Boole (1815-1864) osservarono per primi una corrispondenza fra logica proposizionale classica e un particolare esempio di algebra tradizionale dove si restringono i valori delle variabili nell'intervallo $\{0, 1\}$.

$$
\begin{aligned}
x \wedge y &= xy \\
x \vee y &= x + y - (xy) \\
\neg x &= 1 - x \\
x \rightarrow y &= (1 - x) + y - (1 - x)y
\end{aligned}
\tag{2}
$$

Algebra Booleana

Questo ci permette di scoprire alcune interessanti proprietà:

Nome proprietà	Esempio
Assorbimento	$a \lor (a \land b) = a$ oppure $a \land (a \lor b) = a$
Distributiva	$a \lor (b \land c) = (a \lor b) \land (a \lor c)$
Idempotenza	$a \lor a = a$ oppure $a \land a = a$
DeMorgan (1)	$\neg(\neg x \land \neg y) = x \lor y$
DeMorgan (2)	$\neg(\neg x \lor \neg y) = x \land y$
Complemento	$x \land \neg x = F$
Complemento	$x \lor \neg x = T$
Complemento	$¬ ¬ x = x
Implicazione materiale	$a \rightarrow b = \neg a \lor b$

Se riprendiamo l'esempio di prima:

$$(P \wedge Q) \to \neg R \tag{3}$$

possiamo applicare le regole dell'algebra per arrivare ad una espressione più semplice:

$$\begin{aligned} &= (P \wedge Q) \to \neg R \quad [\text{IMPL. MATER.}] \\ &= \neg(P \wedge Q) \vee \neg R \quad [\text{DE MORGAN 1}] \\ &= \neg P \vee \neg Q \vee \neg R \end{aligned} \tag{4}$$

che vale F proprio quando tutte le variabili P, Q ed R sono T.

Algebra Booleana

Supponiamo di avere la seguente proposizione complessa:

$$\neg(y > 7 \wedge \neg(x > 3)) \vee \neg(y > 7) \vee (x < 0)$$

e di voler capire se, quando $x = 2$, esistono valori di y tali per cui l'espressione è vera. Possiamo usare anche in questo caso l'algebra Booleana:

$$\begin{aligned} & \neg(y > 7 \wedge \neg(x > 3)) \vee \neg(y > 7) \vee (x < 0) \\ = \quad & \neg(y > 7 \wedge \neg F) \vee \neg(y > 7) \vee F \\ = \quad & \neg(y > 7) \vee \neg(y > 7) \\ = \quad & (y \leq 7) \end{aligned} \tag{5}$$

Sono uguali queste due regole di pagamento alla dogana?

1. Tariffa viene scontata se il camion pesa meno di una tonnellata (A) e proviene dalla Francia (B) oppure si muove all'interno di una zona di 200 km dal confine (C).

$$A \wedge B \vee C$$

2. La tariffa non e' scontata se il camion e' piu' pesante di una tonnellata e non si muove all'interno di una zona di 200km dal confine oppure non proviene dalla Francia e va a piu' di 200km dal confine.

$$\neg(\neg A \wedge \neg C \vee \neg B \wedge \neg C)$$

Applicazioni all’informatica

Introduzione

Vi sono almeno due applicazioni importanti della logica nell'informatica degne di nota:

- Scrittura ed analisi delle condizioni nei linguaggi di programamazione
- Sviluppo di circuiti logici

Condizioni nei linguaggi di programmazione

Riprendiamo il nostro esempio iniziale:

> *Se l'anno corrente è bisestile allora stampa "anno bisestile" al terminale altrimenti stampa "anno non bisestile"*

e supponiamo di voler scrivere un programma in linguaggio C che faccia proprio questo.

Condizioni nei linguaggi di programmazione

La condizione alla quale eravamo arrivati era la seguente:

$$(d100 \land d400) \lor (\neg d100 \land d4)$$

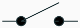
L'espressione diventa la condizione dell'istruzione di controllo `if`:

```c
int anno_corrente = .... ;             /* Richiesto all'utente, oppure una costante */
int d100 = (anno_corrente % 100) == 0; /* '%'  ~  resto della divisione */
int d400 = (anno_corrente % 400) == 0; /* '==' ~  verifica uguaglianza */
int d4   = (anno_corrente % 4)   == 0;
if((d100 && d400) || (!d100 && d4)) {  /* '&&' ~ AND logico */
   printf("Anno bisestile");           /* '!'  ~ NOT logico */
} else {                               /* '||' ~ OR logico */
   printf("Anno non bisestile");
}
```

Algebra dei circuiti

Si fonda su due osservazioni:

1. Data una qualsiasi funzione di n variabili logiche $f : \mathbb{B}^n \to \mathbb{B}$ è possibile descriverla con una espressione logica.
2. Claude Shannon (teoria dell'informazione) dimostrò che gli interruttori elettrici possono essere interpretati come variabili logiche:

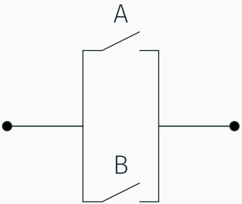
$A = F$ 

$A = V$ 

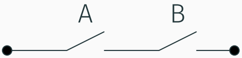
Algebra dei circuiti

Shannon dimostrò anche che alcune configurazioni corrispondono ad operazioni logiche su tali interruttori:

$A \vee B$



$A \wedge B$



Algebra dei circuiti

Al giorno d'oggi si usano "interruttori" speciali chiamate porte logiche. Questo circuito effettua il calcolo di `a + b` con un carry in entrata `cin` producendo la somma `s` ed un carry in uscita `cout`.