2.1 Logica Proposizionale

sabato 26 marzo 2022 11:54

•Una proposizione è un'affermazione riguardo qualcosa nel mondo (es. sta piovendo)
•Può essere vera o falsa

Esempi vari
•"not P" è vera SE P è falso, e viceversa
•"P implica Q" significa che Q è vera solo quando P è vera
•"P if and only Q" significa che P e Q sono entrambe vere o entrambe false

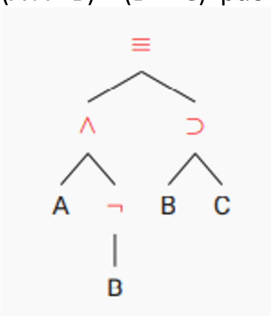
PRIORITA' DEI SIMBOLI

¬	1	negazione - not
∧	2	congiunzione - and
∨	3	disgiunzione - or
⊃	4	sottoinsieme
≡	5	equivalenza

Esempio
$P \land Q \supset R$
Giusto: $(P \land Q) \supset R$
Sbagliato: $P \land (Q \supset R)$

FORMULE COME ALBERI
Una formula può essere rappresentata da un albero
-una sottoformula è un sottoalbero della formula

$(A \land \neg B) \equiv (B \supset C)$ può essere rappresentata come



ESEMPI DI FORMULE
1)If it rains while the sun shines, a rainbow will appear
$(A \land B) \supset C$

2)Chales comes if Elsa does
$A \supset B$

3)Charles comes if Elsa does and the other way around
$A \supset B \lor B \supset A$ SOLUZIONE PROF: $A \equiv B$

4)Johan comes when Peter stays at home
$A \equiv B$

5)We are going, unless it's raining
$A \equiv \neg B$

6)Charles and Elsa are brother and sister or nephew and niece
A ∨ B

7)If I lose If I cannot make a move, then I have lost
(¬A ⊃ B) ⊃ B

FUNZIONE DI INTERPRETAZIONE
•Il dominio di una logica proposizionale è **D={True, False}**
•Un'interpretazione proposizionale è una funzione **I: PROP -> {True, False}** dove prendiamo solo i True
-PROP: formula atomica

	p	q	r	Set theoretic representation
I_1	True	True	False	{p, q}
I_2	False	False	False	{}

ENTAILMENT (IMPLICAZIONI |=)
•L'entailment è la conseguenza logica
•Diciamo che una funzione di interpretazione implica una formula A se valgono i seguenti:
-I |= A   if I(A) = True,   with A ∈ PROP
-I |= ¬A   if not I |= A
-I |= A ∧ B   if I |= A and I |= B
-I |= A ∨ B   if I |= A or I |= B
-I |= A ⊃ B   if, when I |= A, then I |= B
-I |= A ≡ B   if I |= A iff I |= B

How connectives operate (notion)

¬ True	False
¬ False	True
True ∧ True	True
True ∧ False	False
False ∧ True	False
False ∧ False	False
True ∨ True	True
True ∨ False	True
False ∨ True	True
False ∨ False	False

True ⊃ True	True
True ⊃ False	False
False ⊃ True	True
False ⊃ False	True
True ≡ True	True
True ≡ False	False
False ≡ True	False
False ≡ False	True

ERRORI COMUNI
Congiunzione: per indicare la congiunzione (and) usiamo anche altre parole come but, moreover, however, although e though
Esempio: I enjoyed the holiday, even though it rained a lot

Congiunzione 2: a volte l'and non indica una congiunzione
Esempio: Mario and Luigi are brothers
Esempio 2: The leech was long and wet and slimy

Disgiunzione inclusiva o esclusiva:

Implicazione: P ⊃ Q si traduce in tantissime espressioni inglesi. Come "if p, then q", "p implies q", p hence q", ecc.
Esempio: if Mario goes to the party, then I'll go too

Even if: "P even if Q" si può tradurre semplicemente in P, perché Q no ha importanza
Esempio 1: I'll go to the party even if Mario doesn't go
Esempio 2: I'll go to the party whether or not mario goes
Esempio 3: I'll go to the party regardless of whether Mario comes or not

Unless: a volte unless può essere tradotta in un or inclusivo che in un or esclusivo (dipende dal contesto)

Esempio: I'll go to the party unless I get another offer

Condizioni necessarie e sufficienti: P è una **condizione sufficiente** di Q quando il VERO di P garantisce il VERO di Q
A sua volta Q è una **condizione necessaria** di P quando il FALSO di Q garantisce il FALSO di P
Esempio: "se Gustavo è un gatto, allora Gustavo è un mammifero"
-essere un gatto è una **condizione sufficiente** per essere un mammifero
-essere un mammifero è una **condizione necessario** per essere un gatto

----

REASONING TASK

Model checking: dato T ed M, verifica che M |= T
Satisfiability: dato T, verifica se esiste un M tale che M |= T
Validity: dato T, verifica che per ogni M abbiamo M |= T
Unsatisfiability: dato T, verifica che non esista un M tale che M |= T
Logical conseguence: dato T1 e T2, verifica che T1 |= T2
-T1 |= T2 significa che, se M |= T1, allora M |= T2
Logical equivalence: dato T1 e T2, verifica che T1 |= T2 e T2 |= T1

FUNZIONI UTILIZZATE PER IL REASONING TASK
I: interpretazione
A: formula

MCHECK(I,A): ritorna "YES" se A è soddisfatto da I, "NO" altrimenti
SAT(A): se esiste, ritorna un'interpretazione I che soddisfa A, "NO" altrimenti
VALID(A): ritorna "YES" se tutte le I soddisfano A, "NO" altrimenti
UNSAT(A): ritorna "YES" se nessuna I soddisfa A, "NO" altrimenti
LOGCONS(T1, T2): ritorna "YES" se T2 consegue T1, "NO" altrimenti
LOGEQ(T1, T2): ritorna "YES" se T1 e T2 sono logicamente equivalenti, "NO" altrimenti

Validity ⊃ Satisfiability ⊃ not Unsatisfiability
Unsatisfiability ⊃ not Satisfiability ⊃ not Valid