# Curs 8

## Consistența

* Dacă v (niu) este o evaluare booleană și f (fi) este o formula propozițională, v îl satisface pe f dacă f este A
* Daca sigma este o multime de formule propoziționale, v îl satisface pe sigma daca valoarea de adevar a lui sigma este A, pentru orice f din sigma
* Spunem că f (sau sigma) este satisfiabilă dacă există o evaluare booleană care să o satisfacă
* O formulă f este validă sau tautologie dacă este satisfăcută de toate evaluările booleene
* O formulă f este insatisfiabilă sau contradicție dacă nu există o evaluare booleană care să o satisfacă

O mulțime de formule sigma spunem că va deduce o formulă f dacă f este consecință logică pentru sigma, notată cu Σ Π ϕ, dacă fiecare evaluare booleana v care satisface sigma îl satisface și pe f

## Evaluarea definită formal

O evaluare booleană este o funcție v al cărui domeniu este mulțimea tuturor formulelor din logica propozițiilor, iar codomeniul este mulțimea de valori de adevăr, astfel încât valoarea de adevăr a lui p este definită pentru orice formulă atomică p.

Pentru orice formulă alfa avem:

* Dacă v(non alfa)=A, atunci v(alfa)=F
* Dacă v(non alfa)=F, atunci v(alfa)=A

Pentru orice formule alfa si beta avem:

* v(alfa ∧ beta)=A, dacă v(alfa)=A si v(beta)=A
* v(alfa ∧ beta)=F altfel

Pentru orice formule alfa si beta avem:

* v(alfa ∨ beta)=F daca v(alfa)=F si v(beta)=F
* v(alfa ∨ beta)=A altfel

Pentru orice formule alfa si beta avem:

* v(alfa ⊕beta)=F, daca v(alfa)=v(beta)
* v(alfa ⊕beta)=A, altfel

Pentru orice formule alfa si beta avem:

* v(alfa→ beta)=F, dacă v(alfa)=A si v(beta)=F
* v(alfa→ beta)=A, altfel

Pentru orice formule alfa si beta avem:

* v(alfa↔beta)=A, dacă v(alfa)=v(beta)
* v(alfa↔beta)=F, altfel

Pentru orice două formule propoziționale alfa (a) și beta (b) avem:

* v(¬a)→¬v(a)
* v(a∧b)=v(a)∧v(b)
* v(a∨b)=v(a)∨v(b)
* v(a⊕b)=v(a)⊕v(b)
* v(a→b)=v(a)→v(b)
* v(a↔b)=v(a)↔v(b)

Ex:

Știind că valoarea de adevăr a lui p este adevărul, v(q)=F, ce valoare de adevăr va avea următoarea formulă?

(non p sau q) implică (p si q)

v(p si q) = F

v(non p sau q)=F

v((non p sau q) implică (p si q))=A

## Validitate, nesatisfiabilitate, contingență

O propoziție este validă dacă și numai dacă este adevărată în toate interpretările posibile.

Propoziția este validă dacă în ultima coloană a tabelei de adevăr am doar valori adevărate.

Ex:

Merg sau nu merg la concert

* p=”merg la concert”
* p∨¬p
* propoziția compusă este adevărată în orice situație
* Pentru a fi validă o formulă propozițională trebuie să aibă adevăr pe toate liniile, deci trebuie sa construim întreaga tabelă de adevăr.

Ex:

Arătați că (p∧q)→p este validă

|  |  |  |  |
| --- | --- | --- | --- |
| p | q | p∧q | (p∧q)→p |
| 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 |

* formula este validă

O formulă propozițională este satisfiabilă dacă și numai dacă există cel puțin o interpretare în care ea este adevărată

Propoziția este satisfiabilă dacă în ultima coloană a tabelei de adevăr există cel puțin un A.

* nu trebuie să construim întreaga tabelă de adevăr, ci doar să găsim o linie care are ca rezultat valoarea lui A

Ex:

Arătați că p∧(q∨r) este satisfiabilă.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| p | q | r | q∨r | p∧(q∨r) |
| 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 |

* formula este satisfiabilă

O propoziție nu este satisfiabilă, atunci ea se va numi nesatisfiabilă sau contradicție.

O propoziție este nesatisfiabilă dacă pe ultima coloană vom avea valoarea F pe toate liniile.

În limbajul natural, propozițiile care se contrazic, sunt propoziții nesatisfiabile.ț

* pentru a demonstra nesatisfiabilitatea, trebuie să construim întreaga tabelă de adevăr

Ex:

Voi trece și voi pica examenul de logică

p=voi trece examenul de logică

non p = voi pica examenul de logică

p si non p = fals

* propoziția este o contradicție (nesatisfiabilă)

Ex:

Arătați că [(p↔p)→p]∧⎤(p→p) este nesatisfiabilă

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| p | p↔p | (p↔p)→p | p→p | ⎤(p→p) | [(p↔p)→p]∧⎤(p→p) |
| 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 |

* propoziția este nesatisfiabilă