Logică computațională

CURS 3

Logica Propozițiilor

1 Sintaxa logicii propozițiilor

1. alfabet: $\neg$ negația

$\wedge$ conjuncția

$\vee$ disjuncția

$\rightarrow$ implicația

$\leftrightarrow$ echivalența

2. reguli de formare

2 Semantica logicii prop

1. domeniul semantic: $\{F, T\}$

3 Semantica conectivelor:

p	q	$p \wedge q$	$p \vee q$	$p \rightarrow q$	$p \leftrightarrow q$	$p \uparrow q$	$p \downarrow q$	$p \oplus q$
T	T	T	T	T	T	F	F	F
T	F	F	T	F	F	T	F	T
F	T	F	T	T	F	T	F	T
F	F	F	F	T	T	T	T	F

$\uparrow$ - nand: $p \uparrow q = \neg(p \wedge q)$

$\downarrow$ - nor : $p \downarrow q = \neg (p \vee q)$

$\oplus$ - xor : $p \oplus q = \neg (p \leftrightarrow q)$

4 Interpretarea :

- e o fct : $i : \{p_1, p_2, \dots, p_n\}$

5 Concepte semantice : Fie formula $U(p_1, \dots, p_n) \in F_p$.

1. o int. care evaluează form. U ca adevărată, s.n. MODEL al formulei

2. o int. care evaluează form. U ca falsă, s.n. ANTI - MODEL al formulei

3. U s.n. CONSISTENTĂ dacă are cel puțin un model

4. U s.n. VALIDĂ (TAUTOLOGIE), notație $\models$, dacă U este evaluată ca adevărată în orice interpretare

5. U s.n. INCONSISTENTĂ dacă U nu are niciun model

6. U s.n. CONTINGENTĂ dacă U este consistentă, dar nu e tautologie

6 Metasimboluri : relații semantice între formule

1. Formula V este consecință logică a formulei U, notație: $U \models V$, dacă $\forall\, i : F_P \to \{T, F\}$ a.î. $i(U) = T$, are loc $i(V) = T$.

2. Formulele $U(p_1, \dots, p_n) \in F_P$ și $V(p_1, \dots, p_n) \in F_P$ sunt logic echivalente, notație: $U \equiv V$, dacă tabelele lor de adevăr sunt identice, adică $\forall\, i : F_P \to \{T, F\}$, $i(U) = i(V)$

$\boxed{7}$ Concepte semantice pt. mulțimi de formule:

1. O mulțime $\{U_1, \dots, U_n\}$ de formule s.n. CONSISTENTĂ dacă formula $U_1 \wedge U_2 \wedge \dots \wedge U_n$ este consistentă, adică $i(U_1 \wedge U_2 \wedge \dots \wedge U_n) = T$, i este model.

2. —"— s.n. INCONSISTENTĂ dacă form $U_1 \wedge \dots \wedge U_n$ e inconsistentă, adică $i(U_1 \wedge \dots \wedge U_n) = F$, i este anti-model.

3. Formula V este consecință logică a mulț. de formule: $\{U_1, \dots, U_n\}$, notație: $U_1, \dots, U_n \models V$, dacă $\forall\, i : F_P \to \{T, F\}$ a.î. $i(U_1 \wedge \dots \wedge U_n) = T$ are loc $i(V) = T$. Formulele $U_1, \dots, U_n$ s.n. premize, ipoteze, fapte, iar V s.n. concluzie

$\boxed{8}$ Teoremă

Fie $U_1, \dots, U_n, U, V$ formule propoziționale:

a) $\models U$ dacă $\neg U$ este inconsistentă

b) $U \models V$ dacă $\models U \to V$ ddacă mulț. $\{U, \neg V\}$ e inconsistentă (nu există nicio interpretare în care U și $\neg V$ să fie adevărate simultan)

c) $U \equiv V$ dacă $\models U \leftrightarrow V$

d) $U_1, \dots, U_n \models V$ dacă $\models U_1, \dots, U_n \to V$ ddacă mulțimea $\{U_1, \dots, U_n, \neg V\}$ este inconsistentă

$\boxed{9}$ Principiul dualității

$\boxed{10}$ Forme normale în logica propozițiilor:

1. Un literal este o variabilă propozițională sau negația sa
2. O clauză este disjuncția mai multor literali: $p \vee q \vee r$
3. Un cub este conjuncția mai multor literali: $p \wedge q \wedge r$
4. Clauza vidă, simbolizată prin $\square$, este clauza fără literari fiind singura clauză inconsistentă
5. O formulă este în forma normală disjunctivă (FND), dacă aceasta este scrisă ca o disjuncție de cuburi:

$$\underbrace{(\neg p \wedge q \wedge r)}_{\text{cub}} \vee (p \wedge q \wedge r)$$

6. O formulă este în forma normală conjunctivă (FNC), dacă aceasta este scrisă ca o conjuncție de clauze:

$\underbrace{(\neg p \vee q \vee r)}_{\text{clauza}} \wedge (p \vee q \vee r)$

7. Exemple:

p — FNC, 1 clauză; FND, 1 cub

$\neg p \vee q \vee \neg r$ — FNC, 1 clauza; FND, 3 cuburi

$\neg q$ — FNC, clauze; FND, cuburi

$p \wedge \neg q \wedge \neg r$ — FNC, 3 clauze; FND, 1 cub

11 Proprietate:

Fie mult. de literali $\{l_1, \dots, l_n\}$. Urm. afirm. sunt ech.:

clauza $\vee_{i=1}^{n} l_i$ este validă

cubul $\wedge_{i=1}^{n} l_i$ este inconsistentă

în mulțimea $\{l_1, \dots, l_n\}$ există cel puțin o pereche de literali opuși: $\neg p \vee q \vee p = T$; $\neg p \wedge q \wedge p = F$

12 Teoremă

$\forall$ formulă admite o FNC și o FND, logic echivalente cu ea.

13 Algoritmul de normalizare:

PAS 1: Înlocuirea formulelor de tip $U \to V$ cu $\neg U \vee V$ și a formulelor de tip $U \leftrightarrow V$ cu $(\neg U \vee V) \wedge (U \vee \neg V)$

PAS 2: Aplicarea legilor lui De Morgan (se aplică dinspre exterior spre interior)

Eliminarea negațiilor multiple $\neg\neg U \equiv U$

PAS 3: Aplicarea legilor distributivității

- pt FND: $U \wedge (V \vee Z) \equiv (U \wedge V) \vee (U \wedge Z)$
- pt FNC: $U \vee (V \wedge Z) \equiv (U \vee V) \wedge (U \vee Z)$

PAS 4: Simplificarea formulelor obținute folosind alte echivalențe logice:
- legile de simplificare
- legile absorbției
- legile de idempotență

14 Teoremă:

1. O formulă FNC e tautologie ddacă toate clauzele sale sunt valide.
2. O formulă FND e inconsistentă ddacă toate cuburile sale sunt inconsistente.

CURS 4

1 Axiome și reguli de inferență:

1. $A_p = \{A_1, A_2, A_3\}$ scheme axiomatice.

* A_1: $U \to (V \to U)$
* A_2: $(U \to (V \to Z)) \to ((U \to V) \to (U \to Z))$
* A_3: $(U \to V) \to (\neg V \to \neg U)$

2. $R_p = \{mp\}$ - o singură regulă de inferență, modus pones

* $U, U \to V \vdash V$ „din faptele U și $U \to V$ se deduce (inferă) V."

2 Noțiunea de teoremă.

O formulă $U \in F_p$, a.î. $\emptyset \vdash U$ sau $\vdash U$ s.n. teoremă.

Teoremele sunt formule care sunt deductibile doar din axiome și folosind regula modus ponens.

3 Teorema de deducție și inversa ei

1. Dacă $U_1, \dots, U_{n-1}, U_n \vdash V$, atunci $U_1, \dots, U_{n-1} \vdash U_n \to V$ și invers.

4 Consecințele teoremei de deducție:

$\to$ este implicația ("dacă ... atunci ...")

$\vdash U \to ((U \to V) \to V)$

$\vdash (U \to V) \to ((V \to Z) \to (U \to Z))$ legea silogismului

$\vdash (U \to (V \to Z)) \to (V \to (U \to Z))$ legea permutării premizelor

$\vdash (U \to (U \to Z)) \to (U \wedge (V \to Z))$ legea reuniunii premizelor

$\vdash (U \wedge (V \to Z)) \to (U \to (V \to Z))$ legea separării premizelor

5 Proprietățile logicii propoziționale

1. Problemele decizionale în logica propozițiilor:

„Este o formulă propozițională o teoremă sau nu?"

„Este o formulă deductibilă dintr-o mulțime de form?"

2. Teorema de corectitudine:

Dacă $\vdash U$, atunci $\vDash U$

3. Teorema de completitudine:

Dacă $\vDash U$, atunci $\vdash U$

4. Teorema de corectitudine și completitudine:

$\vdash U$ ddacă $\vDash U$

6 Consecințele teoremei de corect. și complet.:

1. Logica prop este <u>necontradictorie</u>:

nu pot avea loc simultan $\vdash U$ și $\vdash \neg U$

2. Logica prop este coerentă:

nu orice formulă prop este teoremă

3. Logica prop este decidabilă:

se poate deduce dacă o formulă prop este sau nu teoremă

CURS 5: METODA TABELELOR SEMANTICE

Ideea: - descompunerea formulei inițiale în subformule
- până la nivel de literali

1 Clase de formule:

* clasa α: form de tip conjunctiv
  - $A \wedge B$
  - $\neg(A \vee B)$
  - $\neg(A \to B)$

* clasa β: form de tip disjunctiv
  - $A \vee B$
  - $\neg(A \wedge B)$
  - $A \to B$

2 Reguli de descompunere a formulelor:

* regula α: $A \wedge B$; $\neg(A \vee B)$; $\neg(A \to B)$

$$\begin{array}{ccccc} A \wedge B & & \neg(A \vee B) & & \neg(A \to B) \\ | & & | & & | \\ A & & \neg A & & \neg A \\ | & & | & & | \\ B & & \neg B & & B \end{array}$$

* regula β:

$$\begin{array}{ccc} & A \vee B & \\ A & & B \end{array} \quad ; \quad \begin{array}{ccc} & \neg(A \wedge B) & \\ \neg A & & \neg B \end{array} \quad ; \quad \begin{array}{ccc} & A \to B & \\ \neg A & & B \end{array}$$

$\boxed{3}$ Arborele liniar de descompunere a unei formule

- se descompune până când pe ramură apare o formulă și negația sa.
- sau până când au fost descompuse toate formulele de pe ramura respectivă.

$\boxed{4}$ Tipuri de ramuri:

1. O ramură a tabelei s.n. ÎNCHISĂ „⊗", dacă ea conține o formulă și negația, în caz contrar ramura s.n. DESCHISĂ „O"
2. O ramură a tabelei s.n. COMPLETĂ dacă ea e fie închisă fie toate formulele de pe acea ramură au fost descompuse

$\boxed{5}$ Tipuri de tabele semantice

1. O tabelă s.n. ÎNCHISĂ dacă toate ramurile sale sunt închise. Dacă o tabelă are cel puțin o ramură deschisă atunci ea s.n. DESCHISĂ.

2. O tabelă s.n. COMPLETĂ dacă toate ramurile ei sunt complete.

6 Observație:

1. Se recomandă utilizarea regulilor de tip α înaintea regulilor de tip β care realizează o ramificare.
2. Formulele de pe aceeași ramură a unei tabele semantice sunt legate între ele prin conectiva logică $\wedge$, iar ramificarea corespunde conectivei logice $\vee$.
3. Fiecare ramură reprezintă un cub, iar arborele reprezintă disjuncția tuturor ramurilor
4. Unei formule consistente i se asociază o tabelă completă deschisă, iar fiecare ramură deschisă furnizează cel puțin un model pt. forma respectivă.
5. O tabelă semantică închisă asociată unei formule indică faptul că formula e inconsistentă, adică $\nexists$ nicio interpretare în care formula să fie adevărată.

7 Teorema de corect. și complet. a tabelelor semantice:

O formulă U, este teoremă (tautologie) ddacă $\exists$ o tabelă semantică închisă pt. formula $\neg U$.

8 Teoremă:

$U_1, \dots, U_n \vdash Y$ (echivalent $U_1, \dots, U_n \vDash Y$) ddacă $\exists$ o tabelă semantică închisă pt. formula: $U_1 \wedge U_2 \wedge \dots \wedge U_n \wedge \neg Y$

CURS 6

Metoda rezoluției

* are ca scop: verificarea consistenței / inconsistenței unei mulțimi de clauze.

1 Terminologie

1. clauzele $C_1 = A \vee l$, $C_2 = B \vee \neg l$ rezolvă pt. că conțin 2 lit opuși
2. notație: $C_3 = Res_l(C_1, C_2)$
3. clauzele C_1, C_2: clauze părinte
4. caz particular: $C_1 = l$, $C_2 = \neg l$, $Res_l(C_1, C_2) = \square$ - inconsistență

2 Alg rezoluției propoziționale

Notație: $S \vdash_{Res} \square$ „din mulțimea S de clauze s-a derivat clauza vidă prin aplicarea alg. rez. prop."

3 Teorema de corect. și complet.

1. Teorema de corectitudine

Dacă $S \vdash_{Res} \square$, atunci S e inconsistentă.

2. Teorema de completitudine

Dacă S e inconsistentă, atunci $S \vdash_{Res} \square$.

3. T.C.C.:

Mulțimea S e inconsistentă ddacă $S \vdash_{Res} \square$.

4 Teoreme:

* U e tautologie ddacă $FNC(\neg U) \vdash_{Res} \square$
* $U_1, \ldots, U_n \vDash V$ ddacă $U_1, \ldots, U_n \vdash V$ ddacă

$FNC(U_1 \wedge U_2 \wedge \ldots \wedge U_n \wedge \neg V) \vdash_{Res} \square$

CURS 7

Rafinările rezoluției

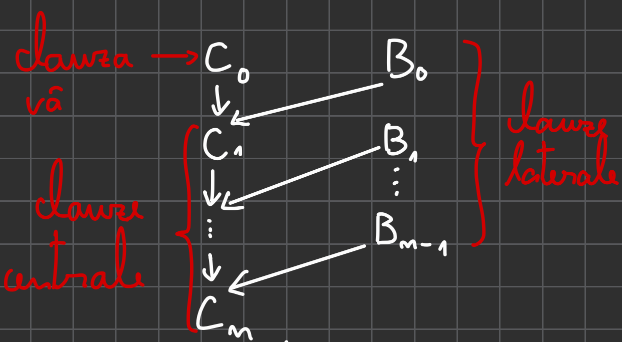
1 Notație:

$S \vdash_{Res}^{st} \square$ „din mulțimea S de clauze s-a derivat clauza vidă prin aplicarea strategiei ST a rezoluției propoziționale"

2 Rezoluția blocării:

- se recomandă combinarea ei cu strategia saturării pe nivele.

3 Rezoluția liniară:



- T.C.C. Mulțimea S de clauze este inconsistentă, ddacă: $S \vdash_{Res}^{lin} \square$

4 Cazuri particulare ale rez. liniare:

1. Rezoluția unitară (unit): clauzele centrale au cel puțin o clauză părinte unitară (conține un singur literal).
2. Rezoluție de intrare (input): clauzele laterale sunt clauze inițiale.

5 Tipuri de metode.

metoda
- semantică (alfa-beta (tabela semantică), tabela adevăr FNC)
- sintactică (rezoluție)
- directă (tabela de adevăr FNC)

<u>CURS 8</u>

Logica predicatelor de ordinul I

- $A1: U \to (V \to U)$
- $A2: (U \to (V \to Z)) \to (U \to V) \to (U \to Z)$
- $A3: (U \to V) \to (\neg V \to \neg U)$
- $A4: (\forall x)\, U(x) \to U(t)$, unde t este un termen arbitrar
- $A5: (U \to V(y)) \to (U \to (\forall x)\, V(x))$, unde y este o variabilă liberă în V care nu apare în U, iar x nu este variabilă liberă nici în U nici în V.
- Modus Ponens: $U, U \to V \vdash_{mp} V$
- Regula generalizării: $U(x) \vdash_{gen} (\forall x)\, U(x)$

Variabilele din formulele predicative care se află sub incidența unui cuantificator se numesc variabile legate, în caz contrar ele se numesc variabile libere.

Legile distributivității:

- $\exists$ față de $\vee$

$$(\exists x)(A(x) \vee B(x)) \equiv (\exists x) A(x) \vee (\exists x) B(x)$$

- $\forall$ față de $\wedge$

$$(\forall x)(A(x) \wedge B(x)) \equiv (\forall x) A(x) \wedge (\forall x) B(x)$$

Legile semidistributivității:

- $\exists$ față de $\wedge$

$$\models (\exists x)(A(x) \wedge B(x)) \to (\exists x) A(x) \wedge (\exists x) B(x)$$

- $\forall$ față de $\vee$

$$\models (\forall x)(A(x) \vee B(x)) \to (\forall x) A(x) \vee (\forall x) B(x)$$

Algoritmul de aducere la forma normală clauzală.

Pas 1: Se înlocuiesc $\to$, $\leftrightarrow$

Pas 2: Se aplică legile lui DeMorgan

Pas 3: Se redenumesc variabilele legate

Pas 4: Se extrag cuantificatorii în fața formulei (forma prenexă)

Pas 5: Eliminarea cuantificatorilor existențiali (forma Skolem)

Pas 6: Eliminarea cuantificatorilor universali (forma Skolem)

Pas 7: Distributivitatea lui $\vee$ față de $\wedge$ (forma normală clauzală)

CURS 9

Metoda tabelelor semantice în calcul predicatelor

clasa γ - formule cuantificate universal

$$\begin{array}{ccc} (\forall x)A(x) & \qquad & \neg(\exists x)A(x) \\ | & & | \\ A(c_1) & & \neg A(c_1) \\ | & & | \\ A(c_2) & & \neg A(c_2) \\ | & & | \\ \dots & & \dots \end{array}$$

$c_1, c_2, \dots, c_m$ - toate constantele existente pe ramură

clasa δ - formule cuantificate existențial

$$\begin{array}{ccc} (\exists x)A(x) & \qquad & \neg(\forall x)A(x) \\ | & & | \\ A(a) & & \neg A(a) \end{array}$$

a - constantă nouă introdusă

CURS 11

Algebre Booleene

O conjuncție de variabile se numește monom.

Un monom care conține toate cele n variabile se numește monom canonic sau minterm de n variabile.

Disjuncția care conține toate cele n variabile se numește maxterm de n variabile.

Metoda diagramelor Veitch

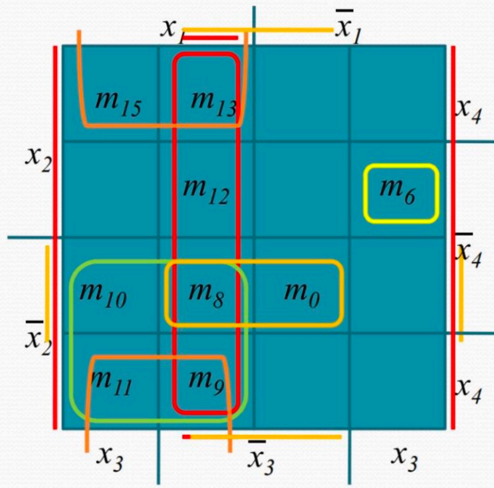
$$max_1 = m_{13} \vee m_{12} \vee m_8 \vee m_9 = x_1\overline{x}_3$$

$$max_2 = m_{10} \vee m_{11} \vee m_8 \vee m_9$$
$$= x_1\overline{x}_2$$

$$max_3 = m_{11} \vee m_9 \vee m_{15} \vee m_{13}$$
$$= x_1x_4$$

$$max_4 = m_8 \vee m_0$$
$$= \overline{x}_2\overline{x}_3\overline{x}_4$$

$$max_5 = m_6 = \overline{x}_1x_2x_3\overline{x}_4$$



Forma simplificată:

$$f^s(x_1, x_2, x_3, x_4) = x_1\overline{x}_3 \vee x_1\overline{x}_2 \vee x_1x_4 \vee \overline{x}_2\overline{x}_3\overline{x}_4 \vee \overline{x}_1x_2x_3\overline{x}_4$$

Metoda diagramelor Karnaugh

x_1 \ x_2x_3	00	01	11	10
0				
1				

Metoda Quine-Mc'Clusky

Metoda analitică a lui **Quine-Mc'Clusky**

Pas 1: Ordonarea mulțimii suport a funcției cu n variabile, după numărul de valori 1 conținut de fiecare n-uplu.

$$f(x_1,x_2,x_3,x_4)=m_4\vee m_6\vee m_7\vee m_8\vee m_9\vee m_{10}\vee m_{11}\vee m_{12}$$

$S_f = \{(0,1,0,0),(0,1,1,0),(0,1,1,1),(1,0,0,0),(1,0,0,1),(1,0,1,0),(1,0,1,1),(1,1,0,0)\}$

$S_f = \{(0,1,1,1),(1,0,1,1),(0,1,1,0),(1,0,0,1),(1,0,1,0),(1,1,0,0),(0,1,0,0),(1,0,0,0)\}$

Pas 2:

Grupul	x_1	x_2	x_3	x_4	
I	0	1	1	1	m_7
	1	0	1	1	m_{11}
II	0	1	1	0	m_6
	1	0	0	1	m_9
	1	0	1	0	m_{10}
	1	1	0	0	m_{12}
III	0	1	0	0	m_4
	1	0	0	0	m_8

Pas 3: Factorizare

Grupul		x_1	x_2	x_3	x_4	
I	√	0	1	1	1	m_7
	√	1	0	1	1	m_{11}
II	√	0	1	1	0	m_6
	√	1	0	0	1	m_9
	√	1	0	1	0	m_{10}
	√	1	1	0	0	m_{12}
III	√	0	1	0	0	m_4
	√	1	0	0	0	m_8
IV=I+II		0	1	1	-	$m_7\vee m_6$
Factorizare	√	1	0	-	1	$m_{11}\vee m_9$
simplă	√	1	0	1	-	$m_{11}\vee m_{10}$
V=II+III		0	1	-	0	$m_6\vee m_4$
	√	1	0	0	-	$m_9\vee m_8$
	√	1	0	-	0	$m_{10}\vee m_8$
		-	1	0	0	$m_{12}\vee m_4$
		1	-	0	0	$m_{12}\vee m_8$
VI=IV+V Factorizare dublă		1	0	-	-	$m_{11}\vee m_9\vee m_{10}\vee m_8$

Mulțimea monoamelor maximale este formată din toate monoamele corespunzătoare liniilor nebifate din tabel.

$max_1=m_7\vee m_6=\bar{x}_1x_2x_3$

$max_2=m_6\vee m_4=\bar{x}_1x_2\bar{x}_4$

$max_3=m_{12}\vee m_4=x_2\bar{x}_3\bar{x}_4$

$max_4=m_{12}\vee m_8=x_1\bar{x}_3\bar{x}_4$

$max_5=m_{11}\vee m_9\vee m_{10}\vee m_8=x_1\bar{x}_2$

$M(f)=\{\bar{x}_1x_2x_3,\bar{x}_1x_2\bar{x}_4,x_2\bar{x}_3\bar{x}_4,x_1\bar{x}_3\bar{x}_4,x_1\bar{x}_2\} = \{max_1,max_2,max_3,max_4,max_5\}$

Pas 4: Identificarea monoamelor centrale

	max_1	max_2	max_3	max_4	max_5
m_7	⊛				
m_{11}					⊛
m_6	*	*			
m_9					⊛
m_{10}					⊛
m_{12}			*	*	
m_4		*	*		
m_8				*	*

$C(f)=\{max_1,max_5\}$

Suntem în cazul II.

Cazuri:

- Cazul I: $M(f)=C(f)\Rightarrow f'(x_1,x_2,x_3,x_4)=g(x_1,x_2,x_3,x_4)$
- Cazul II: $M(f)\neq C(f), C(f)\neq\emptyset \Rightarrow$ $f'(x_1,x_2,x_3,x_4)=g(x_1,x_2,x_3,x_4)\vee h(x_1,x_2,x_3,x_4)$
  Cazul III: $M(f)\neq C(f), C(f)=\emptyset \Rightarrow$ $f'(x_1,x_2,x_3,x_4)=h(x_1,x_2,x_3,x_4)$

Pas 5: Indentificarea formelor simplificate

$g(x_1,x_2,x_3,x_4)=max_1\vee max_5=\bar{x}_1x_2x_3\vee x_1\bar{x}_2$

În tabelul de mai sus:

Se hașurează coloanele care conțin ⊛.

Se hașurează liniile ce conțin * hașurate anterior.

Se observă că cel mai simplu mod de a acoperi tot tabelul este:

$$h(x_1,x_2,x_3,x_4)=max_3$$

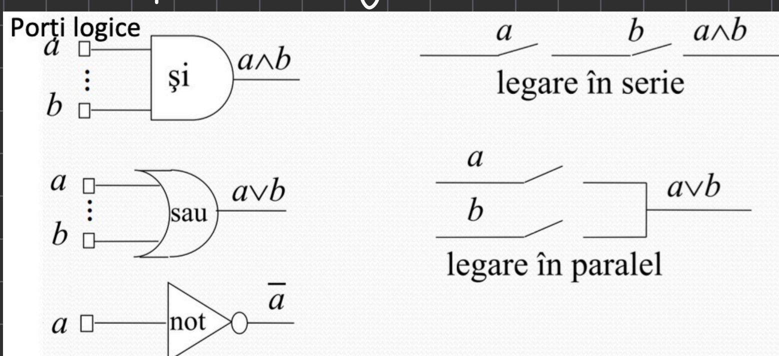
Rezultă că forma simplificată este:

$$f'(x_1,x_2,x_3,x_4)=g(x_1,x_2,x_3,x_4)\vee h(x_1,x_2,x_3,x_4) = \bar{x}_1x_2x_3\vee x_1\bar{x}_2\vee x_2\bar{x}_3\bar{x}_4$$

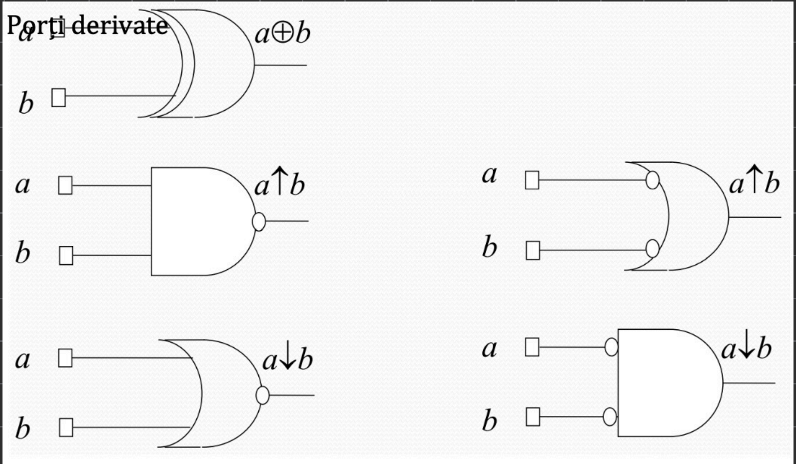
CURS 13

Circuite logice

O poartă este un minicircuit logic care realizează una dintre operațiile logice de bază $\wedge, \vee, \bar{\ }$



Porţi derivate



Pasul 1: Identificarea intrărilor (variabilelor) și ieșirilor (funcțiilor)

- intrare: 4 cifre binare: x_1, x_2, x_3, x_4.
- ieșire: $f_i(x_1, x_2, x_3, x_4) = 1$ pentru $x_1x_2x_3x_{4(2)} = i_{(10)}, i = \overline{0,9}$

Pasul 2: Construirea tabelei de valori asociate

x_1	x_2	x_3	x_4	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	FCD (cu un singur element)
0	0	0	0	1	0	0	0	0	0	0	0	0	0	$f_0(x_1, x_2, x_3, x_4) = \bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4$
0	0	0	1	0	1	0	0	0	0	0	0	0	0	$f_1(x_1, x_2, x_3, x_4) = \bar{x}_1\bar{x}_2\bar{x}_3x_4$
0	0	1	0	0	0	1	0	0	0	0	0	0	0	$f_2(x_1, x_2, x_3, x_4) = \bar{x}_1\bar{x}_2x_3\bar{x}_4$
0	0	1	1	0	0	0	1	0	0	0	0	0	0	$f_3(x_1, x_2, x_3, x_4) = \bar{x}_1\bar{x}_2x_3x_4$
0	1	0	0	0	0	0	0	1	0	0	0	0	0	$f_4(x_1, x_2, x_3, x_4) = \bar{x}_1x_2\bar{x}_3\bar{x}_4$
0	1	0	1	0	0	0	0	0	1	0	0	0	0	$f_5(x_1, x_2, x_3, x_4) = \bar{x}_1x_2\bar{x}_3x_4$
0	1	1	0	0	0	0	0	0	0	1	0	0	0	$f_6(x_1, x_2, x_3, x_4) = \bar{x}_1x_2x_3\bar{x}_4$
0	1	1	1	0	0	0	0	0	0	0	1	0	0	$f_7(x_1, x_2, x_3, x_4) = \bar{x}_1x_2x_3x_4$
1	0	0	0	0	0	0	0	0	0	0	0	1	0	$f_8(x_1, x_2, x_3, x_4) = x_1\bar{x}_2\bar{x}_3\bar{x}_4$
1	0	0	1	0	0	0	0	0	0	0	0	0	1	$f_9(x_1, x_2, x_3, x_4) = x_1\bar{x}_2\bar{x}_3x_4$

Pasul 3: Obținerea expresiilor funcțiilor (FCD de mai sus)

Pasul 4: Simplificarea funcțiilor

Pasul 5: Desenarea circuitului