Universitatea de Vest din Timisoara, Facultatea de Matematica si Informatica
ARHITECTURA CALCULATOARELOR, Informatica I, sem. I 2021-2022

Curs 1, partea 2

Sisteme si baze de numeratie

Reprezentarea numerelor in memoria calculatorului

Elemente de algebra booleana

Dr. Mafteiu-Scai Liviu Octavian

Universitatea de Vest din Timisoara, Facultatea de Matematica si Informatica
ARHITECTURA CALCULATOARELOR, Informatica I, sem. I 2021-2022

Curs 1, partea 2

Sisteme si baze de numeratie

Dr. Mafteiu-Scai Liviu Octavian

Comunicare

Def: limbaj…..

Definitie: Un sistem de numeratie este o multime finita de simboluri numite cifre si o multime de reguli folosite pentru a reprezenta o valoare numerica (numar).

Exemplu: in antichitate, pentru numărul 10 egiptenii foloseau simbolul „ ∩ „, babilonienii simbolul „ < „, iar romanii simbolul „ X”

Sisteme de numeratie:
- Nepozitionale (roman)
- Pozitionale (arab)

Sistem de numerație pozitional:

aportul unei cifre la valoarea totală a unui numar depinde de :

- valoarea cifrei
- locul ocupat de cifră în reprezentarea numărului respectiv.

valoarea indicată de o cifră este valoarea cifrei înmulțită cu o putere a unei constante numită bază.

ordinea puterilor bazei este succesivă, de la dreapta la stânga, cu puterea 0 la ultima cifră în cazul întregilor.

pentru valori fracționare, puterile negative ale bazei se extind la dreapta, începând de la un separator zecimal.

$C_n\, C_{n-1}\, C_{n-2}\, \ldots\, C_2 C_1\, C_0\, ,\, D_1 D_2 D_2 D_3 \ldots\, ,$
$Nr_{(10)} = C_n * b^n + C_{n-1} * b^{n-1} + \ldots + C_2 * b^2 + C_1 * b^1 + C_0 * b^0 + D_1 * b^{-1} + D_2 * b^{-2} + D_3 * b^{-3} + \ldots$

- Ex: $123_{10} = 1 \cdot 10^2 + 2 \cdot 10^1 + 3 \cdot 10^0 = 100 + 20 + 3$
- Tema scurta: 123_7=?

Numărul cifrelor defineşte **baza** sistemului de numeraţie.

Cifrele folosite intr-o baza de numeratie: un set de simboluri care formează o mulțime al cărei număr cardinal este egal cu baza de numerație.

2 cifre: baza 2

10 cifre: baza 10

….

Sisteme si baze de numeratie uzuale:

- **sistemul binar** este un sistem de numaraţie în baza 2. Cifrele utilizate de acest sistem: 0,1.
- **sistemul ternar** este un sistem de numeraţie în baza 4.Cifrele utilizate de acest sistem: 0,1,2,3.
- **sistemul octal** este un sistem de numeraţie în baza 8. Cifrele utilizate de acest sistem: 0,1,2,3,4,5,6,7.
- **sistemul zecimal** este un sistem de numeraţie în baza 10. Cifrele utilizate de acest sistem: 0,1,2,3...,9.
- **sistemul hexazecimal** este un sistem de numeraţie în baza 16. Cifrele utilizate: 0,1,2,...,9,A,B,C,D,E,F.

Exemple reprezentare valori in diferite baze:

Zecimal(10)	Binar(2)	Ternar(4)	Octal(8)	Hexazecimal(16)
0	0	0	0	0
1	1	1	1	1
2	10	2	2	2
3	11	3	3	3
4	100	10	4	4
5	101	11	5	5
6	110	12	6	6
7	111	13	7	7
8	1000	20	10	8
9	1001	21	11	9
10	1010	22	12	A
11	1011	23	13	B
12	1100	30	14	C
13	1101	31	15	D
14	1110	32	16	E
15	1111	33	17	F

Pentru a face distincţie între numerele scrise într-o anumită bază:

- la sfârşitul numărului se mai scrie o literă ce simbolizează baza, de exemplu:
B pentru numerele scrise în binar (baza 2)
Q pentru numerele scrise în octal (baza 8)
D pentru numerele scrise în zecimal (baza 10)
H pentru numerele scrise în hexazecimal (baza 16)
sau
- la sfarsit numar, ca si indice, (cu sau fara paranteze), baza
123_7 $123_{(7)}$

Sistemul binar = mod reprezentare a informatiei in calculator

Codificarea: operația de transformare a informației în secvențe de cifre binare.

- realizata cu ajutorul dispozitivelor de intrare

Decodificarea: operația inversă codificarii.

- realizata de catre dispozitivele de iesire

Conversii intre baze de numeratie

Conversia numerelor dintr-o bază oarecare in baza 10

Pentru un număr scris într-o bază oarecare “b” sub forma parte întreagă şi parte zecimală:

$$\mathbf{Nr_{(b)} = C_n\, C_{n-1}\, C_{n-2} \ldots C_2 C_1\, C_0\, ,\, D_1 D_2 D_2 D_3 \ldots\, ,}$$

valoarea sa în baza 10 este:

$$\mathbf{Nr_{(10)} = C_n * b^n + C_{n-1} * b^{n-1} + \ldots + C_2 * b^2 + C_1 * b^1 + C_0 * b^0 + D_1 * b^{-1} + D_2 * b^{-2} + D_3 * b^{-3} + \ldots}$$

Exemplu:...

Conversii intre baze de numeratie

Conversia numerelor din baza 10 într-o bază oarecare

! trebuie convertite separat partea întreagă şi cea zecimală

a) Conversia părţii întregi

impartirea din clasa I, adica cea cu *cat* si *rest* !!!!!!!!!!!!!!!

7: 2 = ?

- impartiri succesive ale numărului scris în baza 10 la baza in care se doreşte conversia (se împarte numărul la bază, iar în continuare se împarte câtul obţinut la bază ş.a.m.d. până când câtul devine 0), după care se iau resturile obţinute în <u>ordine inversă</u>, care constituie valoarea numărului în baza cerută.

b) Conversia părţii zecimale

- înmulţiri succesive ale părţilor fracţionare până când se ajunge la parte fracţionară nulă, sau se ajunge la perioadă sau se depăşeşte capacitatea de reprezentare (cifre suficiente). Ceea ce depăşeşte partea zecimală la fiecare înmulţire reprezintă o cifră a numărului în baza spre care se face conversia.

Exemplu conversie intre baze de numeratie pentru un numar real

Conversia numerelor din baza 10 într-o bază oarecare

<u>Exemplu:</u> sa se converteasca nr. 2,47 din zecimal in binar, octal, hexazecimal

Pt. partea intreaga este evident 2(D)= 10(B)=**………(O)…………………(H)**

iar pentru partea zecimala trecerea la binar:

0, 47*2

094
188
176
152
104
008
016
032

…….

2,47(D) = 10, 0111 1000 (B)

Operatii aritmetice in baze de numeratie

Adunarea si scaderea: aceleasi reguli ca si la zecimal, adica folosind "transport" respectiv "imprumut"

Exercitii rezolvate:

- 2(10) =10(2) ; 62(10) = 111110(2) ; 1995(10) = 11111001011(2) ; 1024(10) = 10000000000(2)

- 1010011(2) = 83(10) ; 11100011(2) = 227(10) ; 1000000000(2) = 512(10) ; 11001(2) =25(10)

- 86C(16) = 8 x 16^2 + 6 x 16 + C = 8 x 16^2 + 6 x 16 + 12 = 2156(10)

Exercitii propuse:

1) Să se transforme din baza 10 în baza 2 numerele: 23, 57, 101, 567.

2) Să se transforme din baza 2 în baza 10 numerele: 101(2), 1101(2), 111101(2) , 10011(2), 11011(2).

3) Să se transforme din baza 10 în baza 16 numerele: 1221, 1581, 52021, 5638, 4298, 44782

4) Să se transforme din baza 16 în baza 10 numerele: 1001A(16), 110F(16) , A25F(16) , FA0B(16),

CA01F(16).

5) Determinaţi baza de numeraţie x, astfel încât să aibă loc egalităţile:

102(x) = 27

10100(x) = 90

131(x) = 71

Reprezentarea numerelor in memoria calculatorului

Generalitati

- memoria calculatorului = un sir de celule (biti): fiecare celula contine la un moment dat un 0 sau un 1 (nu poate fi goala!).

- Pentru ca o informatie din memorie sa fie utilizabila, trebuie sa stim unde anume se afla (la ce adresa).

- varianta simpla: fiecare informatie sa fie la o adresa fixa (fixata in momentul scrierii programului). => ca si dimensiunea pentru reprezentarea informatiei se fixeaza la scrierea programului.

- In reprezentarea numerelor se stabileste la inceput numarul *n* de biti pe care se face reprezentarea: prin urmare reprezentarea oricarui numar va fi un <u>sir finit</u> de *n* cifre 0 sau 1,

=> nu putem reprezenta orice numar intreg, ci doar numerele dintr-un anumit interval, fixat o data cu alegerea lui *n*.

Reprezentarea numerelor intregi pozitive (multimea N)

- Cea mai naturala reprezentare a numerelor intregi pozitive este scrierea pozitionala in baza 2.
- De exemplu, numarul 20 se reprezinta pe 8 biti ca $00010100_{(2)}$.
- Cel mai mare numar reprezentabil pe 8 biti este 11111111, adica 255, adica 256 de valori distincte (de ce 256?).
- In general, cel mai mare numar reprezentabil pe *n* biti este 2^n-1 (de ce?).
- Cel mai mic numar reprezentabil in acest fel, indiferent de numarul de biti ales, este 0.

- Se mai numeste si <u>reprezentare cu virgula fixa</u>

- **Exercitiu**

Care este intervalul maxim de reprezentare in baza 2 pe *n* biti pentru n numere intregi pozitive? Justificati.

Reprezentarea numerelor intregi pozitive

ADUNAREA

- Aritmetica pentru numere reprezentate in baza 2 se face intocmai ca in baza 10, dar cu tabla adunarii adecvata.

Obs. : noi nu stim sa adunam direct doua numere; stim doar sa aplicam algoritmul de adunare asupra reprezentarilor lor, de obicei in baza 10.

$$C_i = A_i + B_i + T_{i-1}, \text{ unde } T_{-1} = 0$$

Exemplu 1:

20 = 00010100 +
 7 = 00000111
      00011011

Exemplu 2:

144 = 10010000 +
116 = 01110100
260 = 00000100

Obs. transportul 1 de la rangul cel mai semnificativ ? => rezultatul corect al adunarii (260) nu este reprezentabil pe 8 biti, iar rezultatul obtinut este 4 (incorect).

In general, urmatoarele afirmatii sunt echivalente:

- suma este reprezentabila pe lungimea aleasa
- transportul de la rangul cel mai semnificativ este 0
- rezultatul calculului este suma (corecta).

Reprezentarea numerelor intregi pozitive

Conversii de lungime

Pentru a adunare/scadere doua numere reprezentate in baza 2, trebuie sa dispunem de reprezentari pe lungimi egale. In caz contrar, trebuie aplicate conversii de lungime asupra unuia sau ambelor numere:

R1) trecerea de la o reprezentare pe mai putini biti la reprezentare pe mai multi biti se face adaugand cifre 0 in fata.

R2) trecerea de la mai multi biti la mai putini se face ignorand bitii mai semnificativi.

In ce conditii conversiile de lungime sunt corecte (rezultatul este o reprezentare a aceluiasi numar ca si cel de plecare) ?

Ce riscuri implica R2 ? Exista vreo solutie?

Reprezentarea numerelor intregi pozitive si negative (multimea Z)

- Se foloseste un bit pentru reprezentarea semnului (de ex., 0 inseamna plus si 1 inseamna minus) si ceilalti (n-1) biti pentru reprezentarea valorii absolute.

De ce 0 pt. pozitive si nu invers?

! Dar reprezentarea precum la nr. Intregi pozitive creaza probleme!

De ce? (zp si zn)

Solutia:

- reprezentare *in complement fata de 2* sau *cod complementar*:

- numerele pozitive se reprezinta ca si in baza 2; primul bit trebuie sa fie 0, motiv pentru care cel mai mare numar reprezentabil este 2^(n-1)
- numerele negative se reprezinta astfel: plecam de la reprezentarea in baza 2 a opusului (care este pozitiv), dupa care inversam toate cifrele pana la ultimul 1 (exclusiv).

Reprezentarea numerelor intregi pozitive si negative (multimea Z)

Astfel (tot pe 8 biti), -20 se reprezinta ca 11101100. Cel mai mic numar reprezentabil este -2^*n* (care are reprezentarea un 1 urmat de *n*-1 zerouri.

- primul bit se numeste *bit de semn* deoarece el indica semnul numarului (este 0 daca numarul este pozitiv).
- avantajul acestei reprezentari: a*lgoritmii de adunare si de scadere a doua numere reprezentate in cod complementar sunt identici cu algoritmii de adunare si de scadere pentru numere pozitive reprezentate in baza 2.*

Elemente de algebra booleana

Algebra Booleană (logica booleana)

- **George Boole** matematician englez:„O investigare a legilor gândirii" („An investigation of the Laws of Thought") - 1854;

- Aplicatii majore ale algebrei boolene: logica matematică, logica digitală, programarea calculatoarelor, statistica;

- **Shannon Claude Elwood** a descoperit aplicablitatea algebrei booleene în electronică. A demonstrat ca orice problemă de logică se poate rezolva utilizând relee electrice.

Algebra Booleană (logica booleana)

Definim o **funcție logică** (funcție booleană) astfel:

$$f:\{0,1\}\rightarrow\{0,1\}$$

Operatorii logici de bază sunt:

- <u>Conjunctie</u>, produs logic, notat cu SI, AND, *, Λ
- <u>Disjunctie</u>, suma logica, notat cu SAU, OR, +, V
- <u>Negatie</u>,notat cu NOT, $\bar{x}$, ¬

Nota: combinand cei trei operatori de baza se obtin altii derivati precum:

- SAU EXCLUSIV (XOR) (True, True => False)
- SAU NEGAT (NOR)
- SI NEGAT (NAND)

Proprietățile algebrei booleene

- **idempotența**

$$a + a = a$$
$$0 + 0 = 0$$
$$1 + 1 = 1$$
$$a \cdot a = a$$
$$0 \cdot 0 = 0$$
$$1 \cdot 1 = 1$$

- **comutativitatea**

$$a + b = b + a$$
$$a \cdot b = b \cdot a$$

- **asociativitatea**

$$(a + b) + c = a + (b + c) = a + b + c$$
$$(a \cdot b) \cdot c = a \cdot (b \cdot c) = a \cdot b \cdot c$$

Proprietățile algebrei booleene

- **distributivitatea**

$$a \cdot (b + c) = a \cdot b + a \cdot c$$

- **simetria**

$$a = b \rightarrow b = a$$

- **reflexivitatea**

$$a = a$$

- **tranzitivitatea**

$$\left.\begin{array}{l} a = c \\ b = c \end{array}\right\} \rightarrow a = c$$

Regulile lui De Morgan

$$\overline{a\ OR\ b} = \overline{a}\ AND\ \overline{b}$$

$$\overline{a\ OR\ b\ OR\ c} = \overline{a}\ AND\ \overline{b}\ \mathrm{AND}\ \overline{\mathrm{c}}$$

$$\overline{a\ AND\ b} = \overline{a}\ OR\ \overline{b}$$

$$\overline{a\ \mathrm{AND}\ b\ \mathrm{AND}\ c} = \overline{a}\ OR\ \overline{b}\ OR\ \overline{c}$$

Reguli de absorbție/simplificare

a OR a AND b = a

$a\ AND(a\ \mathrm{OR}\ b) = a$

a OR $\overline{a}$ AND b = a OR b

$a\ AND\ (\overline{a}\ OR\ b\ )$ = $a\ AND\ b$

$a\ AND\ b$ OR $\overline{a}\ AND$ c OR $b\ AND\ c$ = $a\ AND\ b$ OR $\overline{a}\ AND\ c$

$(a\ OR\ b)\ AND\ (\overline{a}\ OR\ c)\ AND\ (b\ OR\ c) = (a\ OR\ b)\ AND\ (\overline{a}\ OR\ c)$

Alte relații utile

a OR 0 = a

a OR 1 = 1

a AND 0 = 0

a AND 1 = a

a OR $\overline{a}$ = 1

a AND $\overline{a}$ = 0

$\overline{\overline{a}}$ = a

Reprezentarea funcțiilor logice

Se poate face prin:

- tabele de adevar;
- in forma algebrica (FND sau FNC):
    - Forma normală disjunctivă (FND) (sumă/disjuncție de produse/conjuncții)
    - Forma normală conjunctivă (FNC) (produse/conjuncții de suma/disjuncții)
- notatia sigma

Nota: prima pare a fi mai des si mai usor de folosit

Compararea a doua functii logice:

- aducerea lor la forma canonică, adica operarea cu termeni canonici.
- termen canonic = un termen în care sunt prezente toate variabilele independente, luate sub formă directă sau negată.

Posibilităţi de a exprima forma canonică a unei funcţii:

- forma canonică conjunctivă (**fcc)** – expresia funcţiei este o sumă de produse
- forma canonică disjunctivă (**fcd)** – expresia funcţiei este un produs de sume

Ambele forme se deduc din tabelul de adevăr a funcţiei:

FCC: se însumează toţi termenii pentru care funcţia este egală cu 1

FCD: se scrie produsul sumelor de termeni pentru care funcţia este egală cu 0.

Funcţiile logice sunt funcţii de *n* variabile ce se caracterizează prin faptul că atât variabilele cât şi funcţia nu pot lua decât două valori distincte (0 sau 1).

Câte funcţii de *n* variabile binare există?

Teoremă:

- Numărul N al funcţiilor de n variabile binare este egal cu 2^m, unde $m=2^n$;
- Pentru n variabile binare (n biţi) există $m=2^n$ configuraţii distincte.

Un mod de reprezentare al funcţiilor logice este ***tabelul de adevăr*** – un tabel care în coloana/coloanele din stânga listează toate elementele mulţimii valorilor posibile de intrare, iar în coloana/coloanele din dreapta (coloanele de ieşire) sunt listate valorile corespunzătoare ieşirilor.

1. Funcţii de 1 variabilă

n=1 variabile de intrare (x)

$\Rightarrow$ $m=2^n=2^1=2$ configuraţii distincte şi

$N=2^m=2^2=4$ funcţii de o variabilă (f_0, f_1, f_2 şi f_3)

x	f_0	f_1	f_2	f_3
0	0	1	0	1
1	0	0	1	1

$f_0(x)=0$ funcţia ZERO

$f_1(x)=\overline{x}$ funcţia NOT

$f_2(x)=x$ funcţia DRIVER

$f_3(x)=1$ funcţia TAUTOLOGIE

Funcţii de 2 variabile

n=2 variabile de intrare (x, y)

$m=2^n=2^2=4$ configuraţii distincte ale variabilelor şi

$N=2^m=2^4=16$ funcţii de 2 variabile (f_0, f_1, f_2 … f_{15})

x	y	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Vom avea:

$f_0(x,y)=0$ funcţia ZERO

$f_3(x,y)=$ funcţia NOT

$f_5(x,y)=$ funcţia NOT

$f_{12}(x,y)=x$ funcţia DRIVER

$f_{10}(x,y)=y$ funcţia DRIVER

$f_{15}(x,y)=1$ funcţia TAUTOLOGIE

f_8 este funcţia **ŞI (AND) –** realizează produsul logic $x \cdot y$

x	y	$f_8(x,y)=x \cdot y$
0	0	0
0	1	0
1	0	0
1	1	1

f_8 – funcţia ŞI (AND) ia valoarea 1 daca variabilele de intrare iau valoarea 1

Analog vom avea:

f_7 funcţia **ŞI NEGAT (NAND) –** realizează produsul logic negat

f_{14} funcţia **SAU (OR) –** realizează suma logică

f_1 funcţia **SAU NEGAT (NOR) –** realizează suma logică negată

f_6 funcţia **SAU EXCLUSIV (XOR) –** realizează suma logică modulo2 $x \oplus y$

f_9 funcţia **SAU EXCLUSIV NEGAT (NXOR) –** realizează suma logică modulo2 negată $\overline{x \oplus y}$

Seminar, completare

Conversii directe intre baze de numeratie

Dr. Mafteiu-Scai Liviu Octavian

(8) -> (2)

Se scriu cifrele corespunzatoare bazei 2 pentru fiecare cifra componenta a numarului scris in baza 8

Baza 8	Baza 2
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

$$\mathbf{631_{(8)} = 110\ 011\ 001_{(2)} = 110011001_{(2)}}$$

(2) ->(8)

Grupam cifrele cate 3 si scriem cifra corespunzatoare grupei. Vom avea atatea cifre in baza 8 cate grupe de 3 cifre exista.

Baza 2	Baza 8
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

$$110001010101_{(2)}=110\ 001\ 010\ 101_{(2)}=6125_{(8)}$$

(2) -> (16) -> (2)

Grupam cifrele cate 4 si scriem cifra in baza 16 corespunzatoare grupei. Vom avea atatea cifre in baza 16 cate grupe de 4 cifre am format.

Baza 16	Baza 2	Baza 16	Baza 2
0	0000	8	1000
1	0001	9	1001
2	0010	A	1010
3	0011	B	1011
4	0100	C	1100
5	0101	D	1101
6	0110	E	1110
7	0111	F	1111

$$12A3_{(16)} = 0001\ 0010\ 1010\ 0011_{(2)} = 1001010100011_{(2)}$$

$$1001010100011_{(2)} = 0001\ 0010\ 1010\ 0011_{(2)} = 12A3_{(16)}$$

Exercitii seminar 2,3

Sisteme si baze de numeratie

Reprezentarea numerelor in memoria calculatorului

Elemente de algebra booleana

Exercitii:

1) Să se transforme din baza 10 în baza 2 numerele: 23, 57, 101, 567.

2) Să se transforme din baza 2 în baza 10 numerele: 101(2), 1101(2), 111101(2) , 10011(2), 11011(2).

3) Să se transforme din baza 10 în baza 16 numerele: 1221, 1581, 52021, 5638, 4298, 44782

4) Să se transforme din baza 16 în baza 10 numerele: 1001A(16), 110F(16) , A25F(16) , FA0B(16),

CA01F(16).

5) Determinaţi baza de numeraţie x, astfel încât să aibă loc egalităţile:

102(x) = 27

10100(x) = 90

131(x) = 71

6) Sa se adune 11010 si 10101 in bazele 2, 8,10 si 16

7) Sa se faca scaderea pentru numerele de la ex. 6.

Tema de casa :

8. Aflaţi cifra a ştiind că aaa + aa + a = 861 2

9. Să se determine numărul abc , ştiind că abbc - abb - ab - a = 1779

10. Determinaţi numărul natural de forma ab scris în baza zece pentru care : ab = 5a + 3b .

11. Aflaţi toate numerele naturale de forma ab astfel încât 5 se împarte exact prin numărul (a + b).

12. Determinaţi toate numerele naturale de forma ab57 astfel încât a + b = 11.

13. Determinaţi a şi b astfel încât pentru 17ab24 produsul cifrelor să fie 672.

14. Aflaţi numărul abcd care verifică egalitatea abcd + bcd + cd + d = 3102

15. Scrie numerele de forma abc , astfel încât cifra sutelor sa fie de 3 ori mai mare decât cifra zecilor, iar cifra zecilor de 3 ori mai mică decât cifra unităţilor.

16. Să se determine numerele ab scrise în baza 10 care verifică condiţia: ab=a + 22 · b

17. Să se determine numerele abc scrise în baza 10 care verifică condiţia : a a = bc

18. Determinaţi numărul natural abc cu proprietatea 7a + 5b + 4c = 175.

19. Determinaţi baza de numeraţie x, astfel încât să aibă loc egalităţile: 102(x) = 27 , 10100(x) = 90 , respectiv 131(x) = 71

20. Numărul A se scrie în baza patru A = 333...3 , iar în baza zece se scrie A = 24020 – 1. Aflaţi câte cifre are numărul A în baza patru

Exerciții

Completați următoarul tabel de adevăr:

x	y	$f(x,y) = \bar{x} + \overline{xy} + x\bar{y}$	x	y	z	$f = \bar{x} + \overline{xy}z + x\bar{y}\bar{z} + \bar{x}z$
0	0		0	0	0	
0	1		0	0	1	
1	0		0	1	0	
1	1		0	1	1	
			1	0	0	
			1	0	1	
			1	1	0	
			1	1	1	

- Completați tabelele de adevăr pentru următoarele funcții:

$F(x, y)=x+y+xy$

$F(x, y, z)=x+xy+xy+xz+yz$