**Operațiile logice**

Logica matematică este de fapt o algebră formată pe o mulțime cu doar 2 valori: 0 (fals) și 1 (adevăr)

Operațiile de bază sunt următoarele, pe care le prezentăm de la cea mai prioritară la cea mai puțin prioritară (pe algebra clasică prioritățile sunt: ridicarea la putere, înmulțirea și împărțirea la egalitate, adunarea și scăderea tot la egalitate)

1. Cea prioritară este negația:

se notează cu NOT sau dacă e un bit se folosește și notația cu bară deasupra

|  |  |
| --- | --- |
| a | NOT a |
| 0 | 1 |
| 1 | 0 |

2. Următoarea este conjuncția (AND, și-ul logic). Se folosește notația \* sau &

|  |  |  |
| --- | --- | --- |
| a | b | a AND b |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

3. Următoarea este disjuncția exclusivă (XOR). Se folosește notația (un plus într-un cerc). E ceea ce pe română se formulează cu "sau .. sau ..".

|  |  |  |
| --- | --- | --- |
| a | b | a XOR b |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

4. Următoarea este disjuncția (OR). Se folosește notația + sau |.

|  |  |  |
| --- | --- | --- |
| a | b | a OR b |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

În afară de acestea, la acest tip de exerciții apar deplasări pe biți.

La o deplasare de genul ăsta se specifică întotdeauna pe câți biți lucrăm în total.

Deplasările sunt:

LSHIFT-poz val : deplasarea la stânga cu poz poziții - biții valorii val se deplasează la stânga cu poz poziții. Cifrele care ies prin stânga se pierd. Din dreapta intră zerouri.

De ex., dacă lucrăm pe 6 biți:

LSHIFT-2 101011 = 101100

LCIRC-poz val : deplasarea circulară la stânga cu poz poziții - biții valorii val se deplasează la stânga cu poz poziții. Cifrele care ies prin stânga se lipesc la loc prin dreapta.

De ex., dacă lucrăm pe 6 biți:

LCIRC-2 101011 = 101110

RSHIFT-poz val : deplasarea la drepata cu poz poziții - biții valorii val se deplasează la dreapta cu poz poziții. Cifrele care ies prin dreapta se pierd. Din stânga intră zerouri.

De ex., dacă lucrăm pe 6 biți:

RSHIFT-2 101011 = 001010

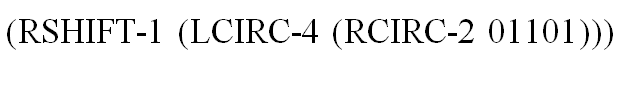
RCIRC-poz val : deplasarea circulară la dreapta cu poz poziții - biții valorii val se deplasează la dreapta cu poz poziții. Cifrele care ies prin dreapta se lipesc la loc prin stânga.

De ex., dacă lucrăm pe 6 biți:

RCIRC-2 101011 = 111010

Avem două tipuri de exerciții:

\* unele simple (de calculat). Exemplu:



(RSHIFT-1 (LCIRC-4 (RCIRC-2 01101))) =

(RSHIFT-1 (LCIRC-4 01011)) =

(RSHIFT-1 10101) = 01010

\* unele complicate (ecuații). Exemplu:

List all possible values of x (5 bits long) that solve the following equation:

(LSHIFT-1 (10110 XOR (RCIRC-3 x) AND 11011)) = 01100

se notează x cu abcde

Există câteva reguli care se verifică ușor cu tabel:

fie a o necunoscută care este în algebra booleană, deci doar a=0 sau a=1.

Reguli:

negația:

a AND 0 = 0

a AND 1 = a

a XOR 0 = a

a XOR 1 =

a OR 0 = a

a OR 1 = 1

(LSHIFT-1 (10110 XOR (RCIRC-3 abcde) AND 11011)) = 01100 ⇒

(LSHIFT-1 (10110 XOR cdeab AND 11011)) = 01100

(LSHIFT-1 (10110 XOR cd0ab)) = 01100

(LSHIFT-1 (10110 XOR cd0ab)) = 01100

(LSHIFT-1 d1b) = 01100

d1b0 = 01100

Și identificăm membru cu membru. Dacă am obține 1=0 sau 0=1 ar însemna că ecuația NU are soluții.

PE ecuația de mai sus obține:

d=0

1=1

=1 ⇒ a=0

b=0

0=0

Deci soluțiile abcde înseamnă: 00\*0\*

(adică avem 4 soluții: 00000, 00001, 00100, 00101)