

## Cuprins

<b>Metoda Grafului Cauză-Efect (Cause-Effect Graphing) .....</b>	<b>1</b>
<b>Crearea grafului cauză-efect (exemplu) .....</b>	<b>4</b>
<b>Crearea tabelului de decizie din graful cauză-efect.....</b>	<b>5</b>

## Metoda Grafului Cauză-Efect (Cause-Effect Graphing)

- Partiționarea în categorii poate produce un număr mare de combinații de intrări, dintre care o mare parte poate fi nefezabilă.
- Metoda grafului cauză-efect (cunoscută și ca modelarea dependențelor) se concentrează pe modelarea relațiilor de dependență între condițiile de intrare ale programului (cauze) și condițiile de ieșire (efecte).
- Relația dintre acestea este exprimată sub forma unui graf cauză-efect.
- Graful cauză-efect = reprezentare vizuală a relației logice dintre cauze și efecte, exprimabilă ca o expresie Booleană.

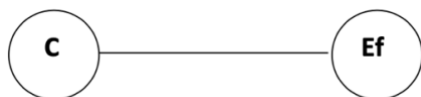
Cauză = orice condiție în specificație (cerințe) care poate afecta răspunsul programului.

Efect = răspunsul programului la o combinație de condiții de intrare. Efectul nu este în mod necesar o ieșire (poate fi un mesaj de eroare, un display, o modificare a unei baze de date sau chiar un punct de testare intern)

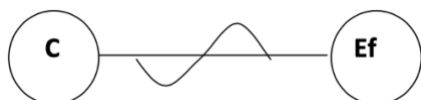
### Notăție:

#### Relații Cauză-Efect

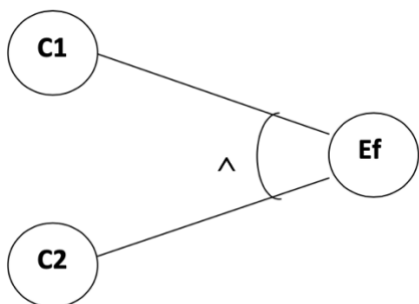
Implies : if C then Ef



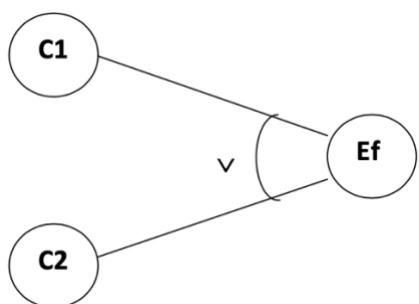
Not – Implies : if ( $\neg$ C) then Ef



And – Implies: if (C1 && C2) then Ef

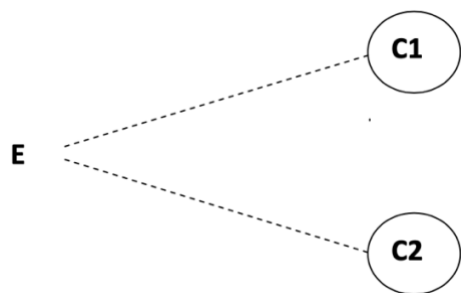


Or – Implies: if (C1 || C2) then Ef

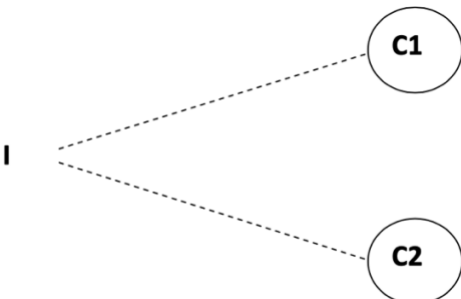


### Constrângeri între cauze

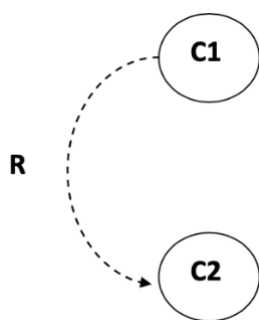
E (exclusive): fie C1 sau C2 (cel mult unul dintre ele)



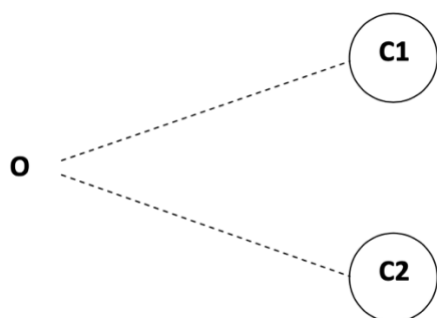
I (Inclusive): cel puțin C1 sau C2



R (Requires) C1 cere C2 (dacă C1 atunci C2)

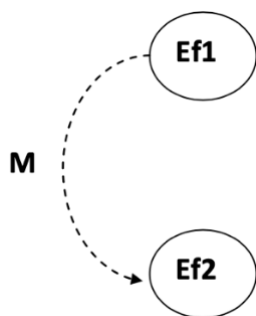


O (one and only one): unul și numai unul dintre C1 și C2



### Constrângeri între efecte

M (Masks): Ef1 maschează Ef2 (dacă Ef1 atunci  $\neg$ Ef2)



## Crearea grafului cauză-efect (exemplu)

după Aditya P. Mathur, Foundations of Software Testing, Pearson Education 2008.

**Cerințe:** O companie vinde pe web calculatoare (CPU1, CPU2, CPU3), imprimante (PR1, PR2), monitoare (M20, M23, M30) și memorie adițională (RAM256, RAM512, RAM1G). O comandă cuprinde între 1 și 4 articole, cel mult câte unul dintre cele 4 categorii amintite. Interfața grafică constă în 4 ferestre (pentru cele 4 categorii de produse) și o fereastră în care sunt afișate articolele primite cadou.

Monitoarele M20 și M23 pot fi cumpărate cu oricare CPU sau singure. M30 poate fi cumpărat doar împreună cu CPU3. PR1 este oferită cadou la cumpărarea lui CPU2 sau CPU3. Monitoare și imprimantele, în afară de M30, pot fi cumpărate separat, fără a cumpăra și CPU. La cumpărarea unui CPU1 se primește RAM256 upgrade, iar la cumpărarea unui CPU2 sau CPU3 se primește RAM512 upgrade. La cumpărarea unui CPU3 și a unui M30 se primește RAM1G upgrade și PR2 cadou.

### Construirea grafului cauză-efect

C1: Cumpărare CPU1

C2: Cumpărare CPU2

C3: Cumpărare CPU3

C4: Cumpărare PR1

C5: Cumpărare PR2

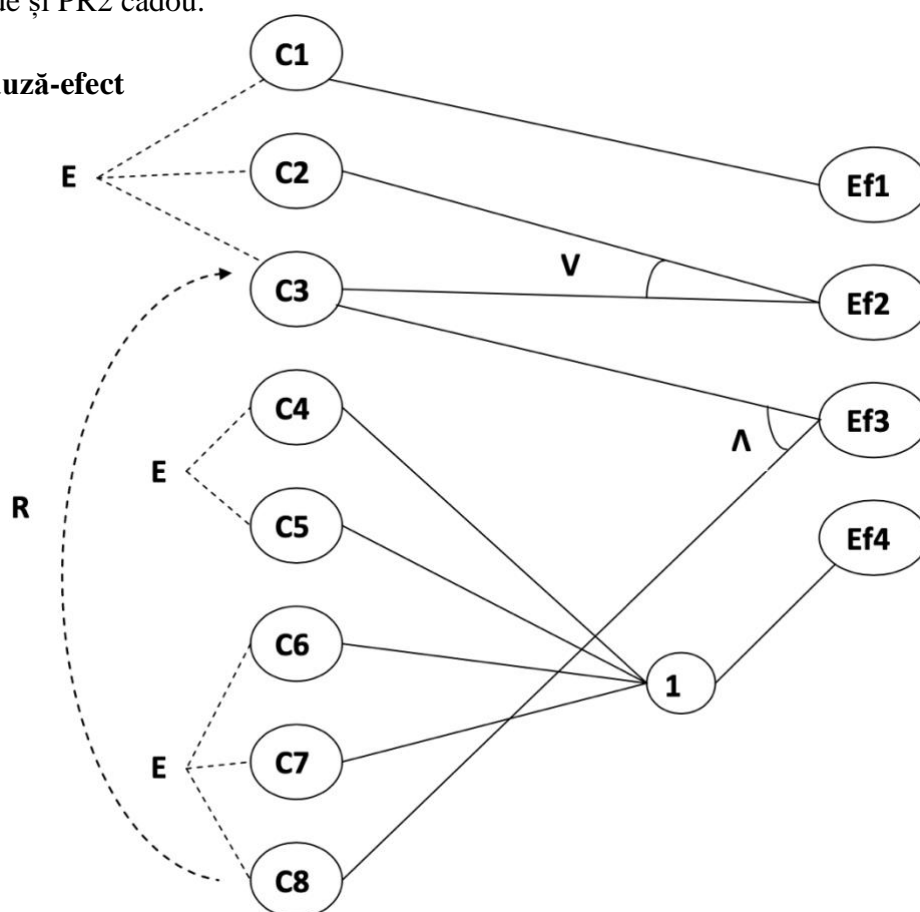
C6: Cumpărare M20

C7: Cumpărare M23

C8: Cumpărare M30

Ef1: RAM256

Ef2: RAM512 și PR1



(pot fi considerate efecte separare, dar se complică fără motiv graful)

Ef3: RAM1G și PR2

Ef4: nici un cadou

## Crearea tabelului de decizie din graful cauză-efect

**Input:** Un graf cauză-efect având cauze  $C_1, \dots, C_p$  și efecte  $Ef_1, \dots, Ef_q$ .

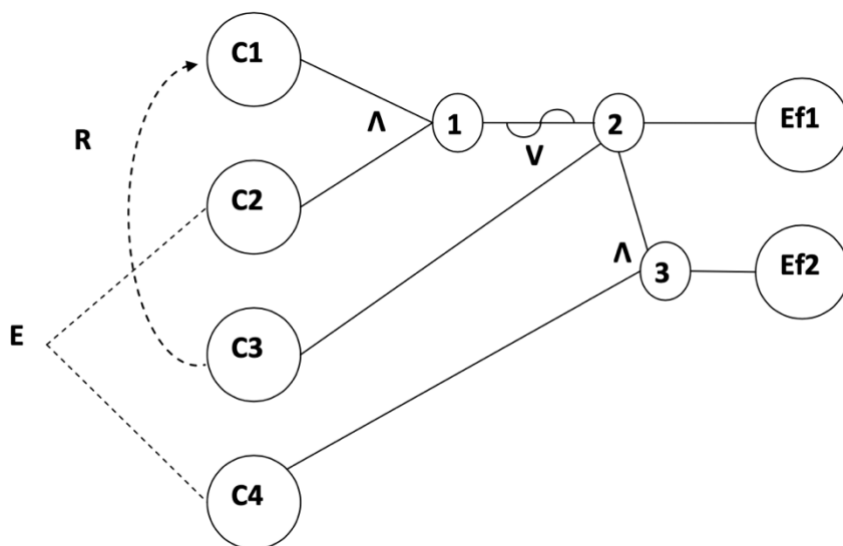
**Output:** Un tabel de decizie având  $N = p + q$  randuri și  $M$  coloane, unde  $M$  depinde de relația dintre cauze și efect

### Procedura de creare a tabelului de decizie

1. Inițializează  $nr\_coloane = 0$  (tabel de decizie gol)
2. For  $i = 1$  to  $q$ 
  - 2.1.  $e = Ef_i$  (selectează următorul efect pentru procesare)
  - 2.2. Găsește combinațiile de condiții care produc apariția efectului  $e$ .  
Fie  $V_1, \dots, V_{m_i}$  aceste combinații,  $m_i > 0$ . Setează  $V_k(j)$ ,  $p < j \leq p+q$ , la 1 dacă efectul  $Ef_j$  apare ca urmare a combinației respective și la 0 în caz contrar
  - 2.3. Actualizează tabelul de decizie. Adaugă  $V_1, \dots, V_{m_i}$  la tabel pe post de coloane succesive începând cu poziția  $nr\_coloane + 1$ .
  - 2.4.  $nr\_coloane = nr\_coloane + m_i$ .

Nr coloane rezultate este  $M = nr\_coloane$ .

### Exemplu



Pas 1:  $nr\_coloane = 0$

Pas 2:  $i = 1$

Pas 2.1:  $e = Ef_1$

Pas 2.2:

Se caută valorile lui C1, C2, C3 astfel încât

$$\neg (C1 \wedge C2) \vee C3 = 1$$

1	0	1
0	1	1
0	0	1
1	1	1
1	0	0
0	1	0
0	0	0

În plus, se aplică contrângerea C3 implică C1

1	0	1
1	1	1
1	0	0
0	1	0
0	0	0

Se adaugă C4 = 0 și valorile corespunzătoare pentru Ef1 și Ef2

<b>V1</b>	1	0	1	0	1	0
<b>V2</b>	1	1	1	0	1	0
<b>V3</b>	1	0	0	0	1	0
<b>V4</b>	0	1	0	0	1	0
<b>V5</b>	0	0	0	0	1	0

Pas 2.3: Matricea obținută este transpusă și adăugată la tabelul de decizie începând cu poziția nr\_coloane + 1 = 1

	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>C1</b>	1	1	1	0	0
<b>C2</b>	0	1	0	1	0
<b>C3</b>	1	1	0	0	0
<b>C4</b>	0	0	0	0	0
<b>Ef1</b>	1	1	1	1	1
<b>Ef2</b>	0	0	0	0	0

Pas 2.4: Se actualizează nr\_coloane = 0 + 5 = 5

Pas 2:  $i = 2$

Pas 2.1:  $e = Ef2$

Pas 2.2:

Se caută valorile lui  $C1, C2, C3, C4$  astfel încât

$$(\neg (C1 \wedge C2) \vee C3) \wedge C4 = 1$$

Folosind combinațiile  $C1, C2, C3$  anterioare pentru  $Ef1$ , obținem

1	0	1	1
1	1	1	1
1	0	0	1
0	1	0	1
0	0	0	1

În plus, se aplică constrângerea ca  $C2$  și  $C4$  să nu existe simultan

1	0	1	1
1	0	0	1
0	0	0	1

Se adaugă valorile corespunzătoare pentru  $Ef1$  și  $Ef2$

<b>V1</b>	1	0	1	1	1	1
<b>V2</b>	1	0	0	1	1	1
<b>V3</b>	0	0	0	1	1	1

Pas 2.3: Matricea obținută este transpusă și adăugată la tabelul de decizie începând cu poziția  $nr\_coloane + 1 = 6$

	1	2	3	4	5	6	7	8
<b>C1</b>	1	1	1	0	0	1	1	0
<b>C2</b>	0	1	0	1	0	0	0	0
<b>C3</b>	1	1	0	0	0	1	0	0
<b>C4</b>	0	0	0	0	0	1	1	1
<b>Ef1</b>	1	1	1	1	1	1	1	1
<b>Ef2</b>	0	0	0	0	0	1	1	1

Pas 2.4: Se actualizează  $nr\_coloane = 5 + 3 = 8$

Procedura se termină. Nr coloane rezultate este  $M = 8$ .

### **Generarea cazurilor de testare**

Fiecare coloană din tabelul de decizie generează cel puțin un caz de testare, corespunzător combinației  $C_1, \dots, C_p$  respective.

Observație:  $C_1, \dots, C_p$  sunt în general expresii care folosesc variabile, etc., deci pentru o combinație pot fi selectate mai multe cazuri de testare.

Problemă: Explozie a stărilor datorită combinației de cauze.

Soluție: Limitarea numărului de cazuri de testare folosind euristici.