

Szoftver- és rendszerellenőrzés (VIMIMA01)

# Specifikáció alapú tesztervezési módszerek

Majzik István, Micskei Zoltán

<http://www.inf.mit.bme.hu/>



**Bill Sempf**

@sempf

QA Engineer walks into a bar. Orders a beer. Orders 0 beers. Orders 9999999999 beers. Orders a lizard. Orders -1 beers. Orders a sfdeljknesv.

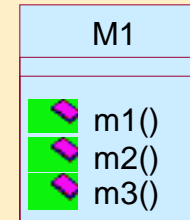
# Klasszikus tesztelési feladat

- A tesztelendő program beolvas 3 egész szám paramétert, egy háromszög három oldalának hosszait. Válaszként kiírja, hogy a háromszög általános, egyenlő szárú vagy egyenlő oldalú.
  - » Glen Myers, *The Art of Software Testing*, 1979
- Milyen tesztek terveznénk ehhez a programhoz?
- Eltérő megoldási javaslatok:
  - Beck: 6 teszt
  - Binder: 65 teszt
  - Jorgensen: 185 teszt!
  - Specifikáció hiányosságok?

# Teszttervezés módszerei

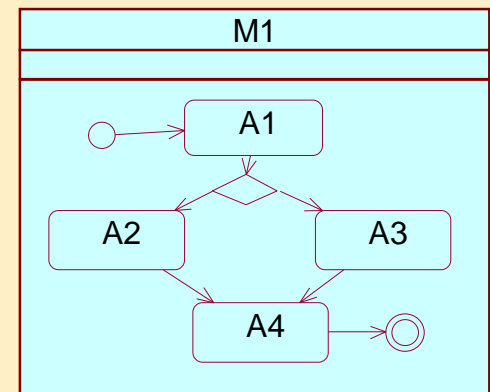
## I. Specifikáció alapú

- A rendszer mint „fekete doboz” adott
- Csak a külső viselkedés (funkció) ismert, a belső felépítés (pl. forráskód) nem
- Tesztelés alapja: **specifikált funkciók léte**; extra funkciók hiánya



## II. Struktúra alapú

- A rendszer mint „üvegdoboz” adott
- A belső struktúra is ismert
- Tesztelés alapja a belső működés: programgráf bejárása



# I. Specifikáció alapú tesztelési módszerek

## Cél:

- A funkcionális specifikációra építve,
- reprezentatív adatok keresése az egyes funkciók teszteléséhez.

## Módszerek:

### **1. Ekvivalencia particionálás**

2. Határérték-analízis

3. Ok-hatás analízis / Döntési táblák

4. Kombinatorikus módszerek

5. Véges automata alapú

6. Használati eset tesztelés

# 1. Ekvivalencia particionálás

- Equivalence Class Partitioning (ECP)
- Bemenet és kimenet **ekvivalencia osztályai**:
  - Olyan adatok, amelyek várhatóan **ugyanazt a hibát fedik le** (ugyanazt a programrészt járják be)
  - Cél: **Egy-egy** ekvivalencia osztályból **egy-egy** teszt adat (az adott bemenethez illetve kimenet alapján); a többi adat esetén a helyesség induktívan következik
- Bemenet értelmezését ismerni kell!
  - Tesztelő tudásán múlik a módszer hatékonysága

# Ekvivalencia osztályok meghatározása

## Meghatározás heurisztikus folyamat:

1. Érvényes és érvénytelen bemeneti adatok
2. Partíciók tovább finomítása

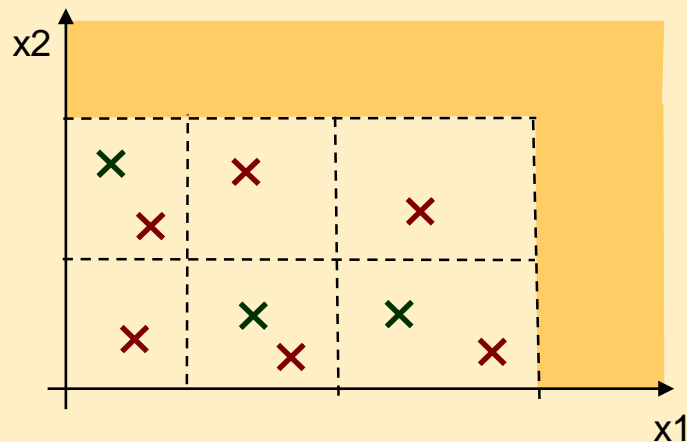
## Heurisztikák a meghatározáshoz:

- Tartomány (pl. 1-1000)
  - $< \text{min}$ ,  $\text{min-max}$ ,  $> \text{max}$
- Halmaz (pl. RED, GREEN, BLUE)
  - érvényes elem, érvénytelen
- Specifikus (pl. első karakternek @-nak kell lennie)
  - feltétel teljesül, feltétel nem teljesül
- Egyéni (pl. február hónap)
  - egyéni eset külön partícióba

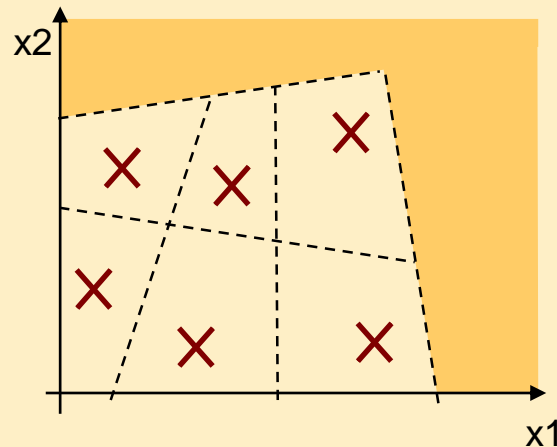
# Tesztesetek származtatása

Tesztek meghatározása több bemenet esetén:

- **Érvényes** ekvivalencia osztályok:  
egy teszt minél több osztályt fedjen le
- **Érvénytelen** ekvivalencia osztályok:  
először minden érvénytelen osztályhoz külön teszt legyen  
(egymás hatását ne oltásák ki), majd több osztály kombinációja is



- Gyenge ill.
- **Erős** normál ekvivalencia osztályok

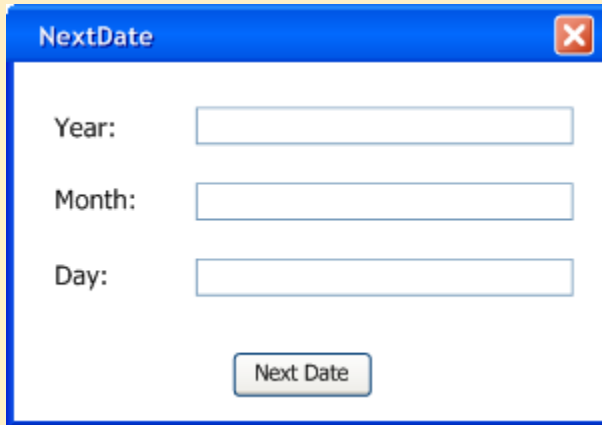


- Dimenzióként nem függetlenül alakítható partíciók: **Erős** osztályok



# Példa: NextDate program

- NextDate program



The image shows a screenshot of a Windows-style application window titled "NextDate". The window has a blue title bar with a close button (an 'X' in a red square) in the top right corner. The main content area is white and contains three input fields for date components: "Year:", "Month:", and "Day:". Each label is followed by a rectangular text box. Below these fields is a button labeled "Next Date".

- Következő naptári napot határozza meg a Gregorián naptár alapján
- Mik a bemenet ekvivalencia osztályai?
- Mik a kimenet ekvivalencia osztályai?

# Példa: Ekvivalencia osztályok meghatározása

Bemenet	Érvényes	Érvénytelen
Hónap	V1: 30 napos hónap V2: 31 napos hónap V3: február	I1: $\geq 13$ I2: $\leq 0$ I3: nem szám I4: üres
Nap	V4: 1-30 V5: 1-31 V6: 1-28 V7: 1-29	I5: $\geq 32$ I6: $\leq 0$ I7: nem szám I8: üres
Év	V8: 1582-9999 V9: nem szökőév V10: szökőév V11: század nem szökőév V12: század szökőév	I9: $\leq 1581$ I10: $\geq 9999$ I11: nem szám I12: üres
Speciális	V13: 1752.09.03-1752.09.13.	I13: 1582.10.5-1582.10.14.

# Példa: Tesztesetek származtatása

Egy lehetséges kombináció:

Helyes érték véletlenszerű választása

Teszt	Hónap	Nap	Év	Egyéb	Kimenet
T1	$V1 \cup V2 \cup V3$	V6	V8		Érvényes
T2	V1	V4	$V9 \cap V8$		
T3	V2	V5	$V10 \cap V8$		
T4	V3	V6	$V11 \cap V8$		
T5	V7	V7	$V12 \cap V8$		Érvényes
T6				V13	Érvényes
T7					Hiba
T8	I2				Hiba
T9	I3				Hiba
T10	I4				Hiba
T11		I1			Hiba
...					

Egy paraméter érvénytelen, többi érvényes

Szerepeljen minden osztály

# I. Specifikáció alapú tesztelési módszerek

## Cél:

- A funkcionális specifikációra építve,
- reprezentatív adatok keresése az egyes funkciók teszteléséhez.

## Módszerek:

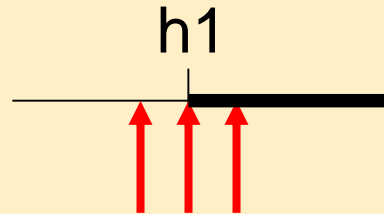
1. Ekvivalencia particionálás
- 2. Határérték-analízis**
3. Ok-hatás analízis / Döntési táblák
4. Kombinatorikus módszerek
5. Véges automata alapú
6. Használati eset tesztelés

## 2. Határérték-analízis (Boundary Value Analysis)

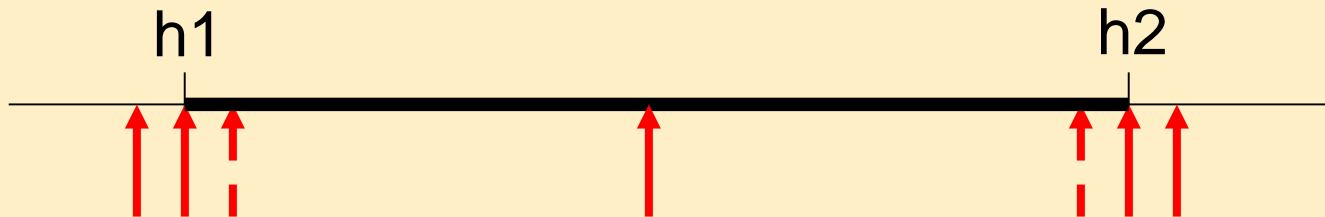
- Az adattartományok határait vizsgálja
  - Egy-egy ekvivalencia osztály **hatáira** koncentrálnak
  - **Bemeneti és kimeneti** tartományokra is
  - Alsó és felső határokra
- Tipikus megtalált hibák
  - Hibás relációs operátorok
  - Hibák a ciklusok be- és kilépési feltételeinél
  - Hibák az adatstruktúrák méreténél
  - ...

# Határérték analízis

- Tipikus adatok:
  - Egy határérték 3 tesztet jelent



- Egy tartomány 5-7 tesztet jelent



# Példa: határértékek a NextDate esetén

- Mik a határértékek a NextDate esetén?
- Hónap
  - 1, 12
  - Tesztelendő: 0, 1, (2), 3-10, (11), 12, 13
- Nap
  - 1, 31
  - Tesztelendő: 0, 1, (2), 3-29, (30), 31, 32
  - Finomítás: 28, 29, 30 is határérték lehet
- Év
  - 1582, 9999
  - Tesztelendő: 1581, 1582, (1583), 1584-9997, (9998), 9999, 10000

# I. Specifikáció alapú tesztelési módszerek

## Cél:

- A funkcionális specifikációra építve,
- reprezentatív adatok keresése az egyes funkciók teszteléséhez.

## Módszerek:

1. Ekvivalencia particionálás
2. Határérték-analízis
- 3. Ok-hatás analízis / Döntési táblák**
4. Kombinatorikus módszerek
5. Véges automata alapú
6. Használati eset tesztelés



### 3. Ok-hatás analízis

A bemenetek és kimenetek kapcsolatának vizsgálata (ha ez egyszerűen leírható)

- **Ok:** egy-egy bemeneti ekvivalencia osztály
- **Hatás:** egy-egy kimeneti ekvivalencia osztály
- Ezekből logikai változókat képzünk

Boole-gráf: Okok és hatások összekapcsolása

- ÉS, VAGY kapcsolatok
- Meg nem engedett kombinációk

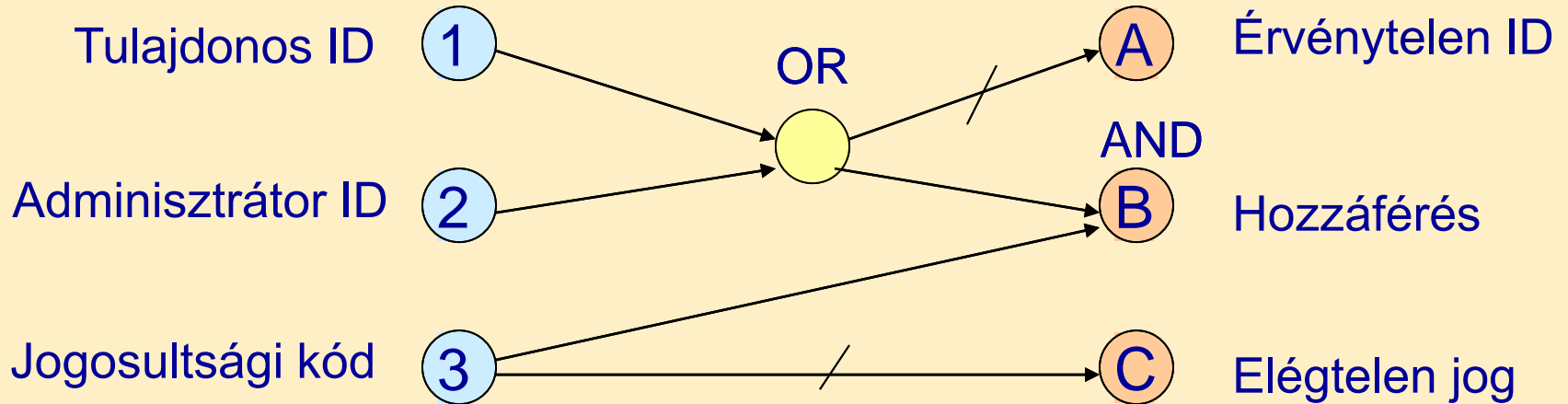
Tesztelési cél: A gráf szisztematikus végigjárása

- Logikai hálózat igazságtáblázatának lefedése
- Egy oszlop egy tesztnek felel meg

# Példa: Ok-hatás leírása

Bemenetek:

Kimenetek:



	T1	T2	T3	...	
Bemenetek	1	0	1	0	
	2	1	0	0	
	3	1	1	1	
Kimenetek	A	0	0	1	
	B	1	1	0	
	C	0	0	0	

# I. Specifikáció alapú tesztelési módszerek

## Cél:

- A funkcionális specifikációra építve,
- reprezentatív adatok keresése az egyes funkciók teszteléséhez.

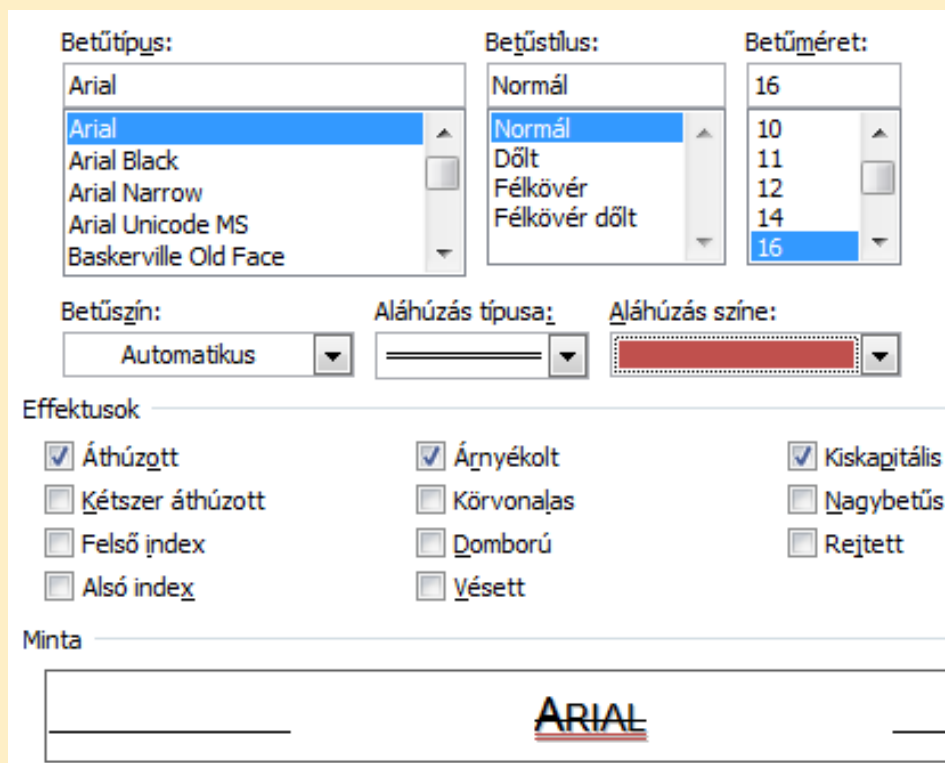
## Módszerek:

1. Ekvivalencia particionálás
2. Határérték-analízis
3. Ok-hatás analízis / Döntési táblák
- 4. Kombinatorikus módszerek**
5. Véges automata alapú
6. Használati eset tesztelés

## 4. Kombinatorikus módszerek

- Paraméterek kombinációja

- Paraméterek kombinációja okozza a legtöbb hibát
- 3-nál több paraméter esetén már rengeteg eset
- Ritka kombinációk veszélyesek lehetnek



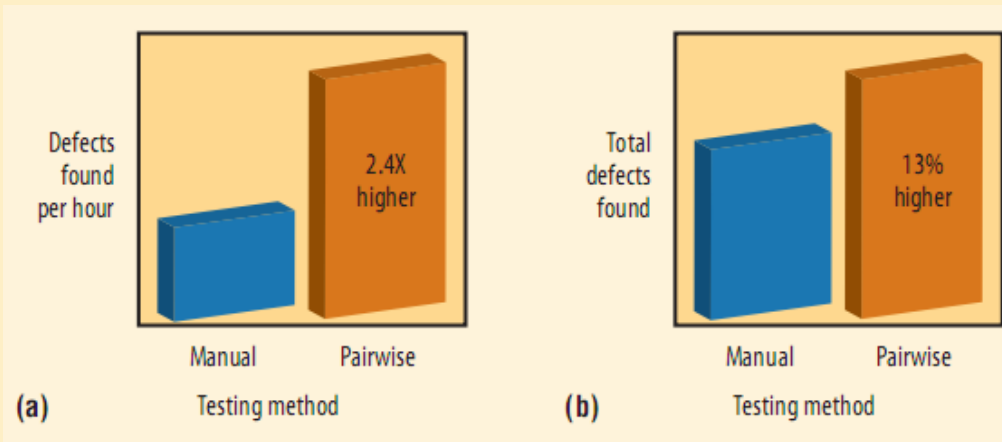
The image shows a font settings dialog box with the following sections:

- Betűtípus:** A list of fonts including Arial, Arial Black, Arial Narrow, Arial Unicode MS, and Baskerville Old Face. 'Arial' is selected.
- Betűstílus:** A list of styles including Normál, Dőlt, Félkövér, and Félkövér dőlt. 'Normál' is selected.
- Betűméret:** A list of font sizes including 10, 11, 12, 14, and 16. '16' is selected.
- Betűszín:** A dropdown menu set to 'Automatikus'.
- Aláhúzás típusa:** A dropdown menu showing a solid underline.
- Aláhúzás színe:** A color selection box showing a red color.
- Effektusok:** A grid of checkboxes for various effects:
  - Áthúzott
  - Árnyékolt
  - Kiskapitális
  - Kétszer áthúzott
  - Körvonalas
  - Nagybetűs
  - Felső index
  - Domború
  - Rejtett
  - Alsó index
  - Véssett
- Minta:** A preview area showing the word 'ARIAL' in a red, underlined, shadowed font.

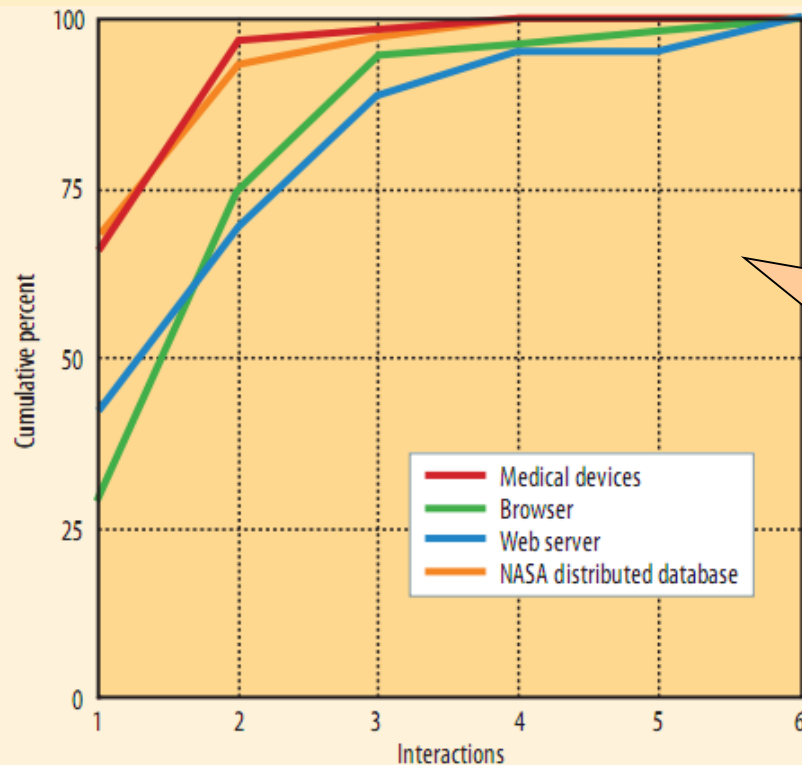
# Kombinatorikus tesztelési technikák

- Ad hoc, „best guess”
  - Intuíció, követelmények, tipikus hibák alapján
- Minden választás (each choice)
  - Minden lehetőség szerepeljen egyszer
  - Alap tesztkészletnek hasznos csak
- N-szeres tesztelés (n-wise testing)
  - Tetszőlegesen választott  $n$  darab paraméter minden lehetséges kombinációjának lefedése a tesztelési cél
  - Elnevezés még: n-wise coverage
  - Speciális eset ( $n=2$ ): Páronkénti tesztelés (pair-wise testing)

# N-wise testing hatékonysága



Ad-hoc és páronként szisztematikus tesztelés összehasonlítása (10 projektre)



A hibák jelentős része 2 paraméter kapcsolatán múlik (de alkalmazástól függően lehet még elég sok hiba, ami 3 vagy több paraméter speciális kombinációja esetén deríthető ki)

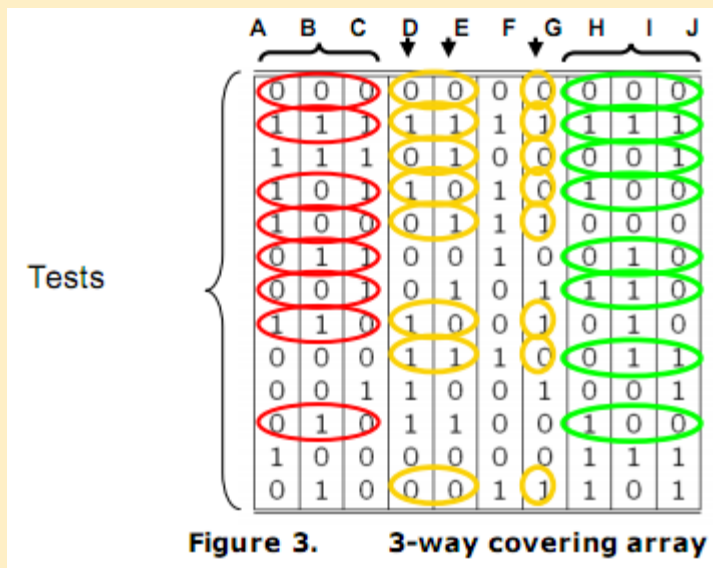
Forrás: R. Kuhn et al. „Combinatorial Software Testing”, IEEE Computer, 42:8, 2009

# Példa: Pair-wise tesztelés

- Adottak a következő konfigurációs lehetőségek:
  - OS: Windows, Linux
  - CPU: Intel, AMD
  - Protocol: IPv4, IPv6
- Kombinációk száma?
- Páronkénti tesztelést megvalósító tesztkészlet?
- Lehetséges megoldás:
  - 1: Windows, Intel, IPv4
  - 2: Windows, AMD, IPv6
  - 3: Linux, Intel, IPv6
  - 4: Linux, AMD, IPv4

# N-szeres tesztelés a gyakorlatban

- Feladat: coverage array előállítása



Forrás: D. R. Kuhn, R. N. Kacker, Y. Lei  
[Practical Combinatorial Testing](#)  
NIST Special Publication 800-142

- Támogató eszközök
  - <http://www.pairwise.org>
  - PICT - Pairwise Independent Combinatorial Testing (MS)
  - [ACTS](#) - Advanced Combinatorial Testing Suite (NIST)



# I. Specifikáció alapú tesztelési módszerek

## Cél:

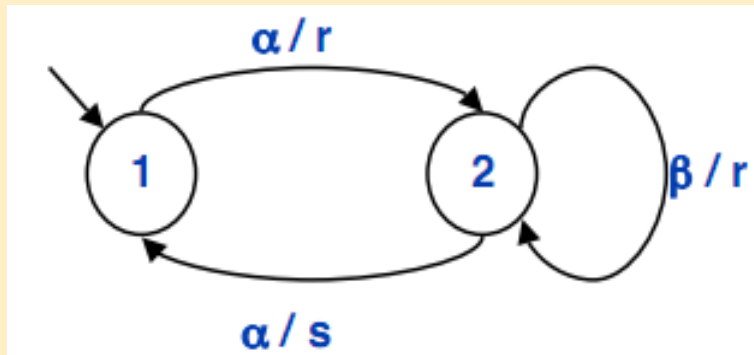
- A funkcionális specifikációra építve,
- reprezentatív adatok keresése az egyes funkciók teszteléséhez.

## Módszerek:

1. Ekvivalencia particionálás
2. Határérték-analízis
3. Ok-hatás analízis / Döntési táblák
4. Kombinatorikus módszerek
- 5. Véges automata alapú**
6. Használati eset tesztelés

## 5. Véges automata alapú

- Specifikáció egy véges automatával adott
- Tipikus tesztelési célok:
  - Minden állapot, minden átmenet, nem megengedett átmenetek tesztelése, stb.



- Problémák:
  - Milyen állapotban van a rendszer?
  - Végállapot / kezdőállapot
- Módszerek
  - Automatikus tesztgenerálás (ld. később)
  - W, Wp módszerek

# I. Specifikáció alapú tesztelési módszerek

## Cél:

- A funkcionális specifikációra építve,
- reprezentatív adatok keresése az egyes funkciók teszteléséhez.

## Módszerek:

1. Ekvivalencia particionálás
2. Határérték-analízis
3. Ok-hatás analízis / Döntési táblák
4. Kombinatorikus módszerek
5. Véges automata alapú
- 6. Használati eset tesztelés**

## 6. Használati eset tesztelés

- Tesztek származtathatók a használati esetekből
- Tesztesetek:
  - 1 teszt: fő ág („happy path”, „mainstream”)
    - Ellenőrzés: utófeltételek vizsgálata
  - Alternatív lefutások: mindegyikhez külön teszteset
  - Előfeltételek (nem)teljesülése
- Tipikusan integrációs és elfogadási tesztek

# Módszerek együttes alkalmazása

## Alap módszerek tipikus sorrendje:

1. Ekvivalencia particionálás
2. Határérték-analízis
3. Ok-hatás analízis, vagy kombinatorikus, vagy véges automata alapú

## Kiegészítés: Véletlen tesztek

- Véletlen teszt adatok generálása
- Kis számítási teljesítményt igényel, gyors
- Hibafedése nem garantálható
- Teszt eredmény kiértékelése:
  - Válasz számítása, szimulálása
  - Csak „elfogadhatósági vizsgálat” (durva hibák kiszűrése)