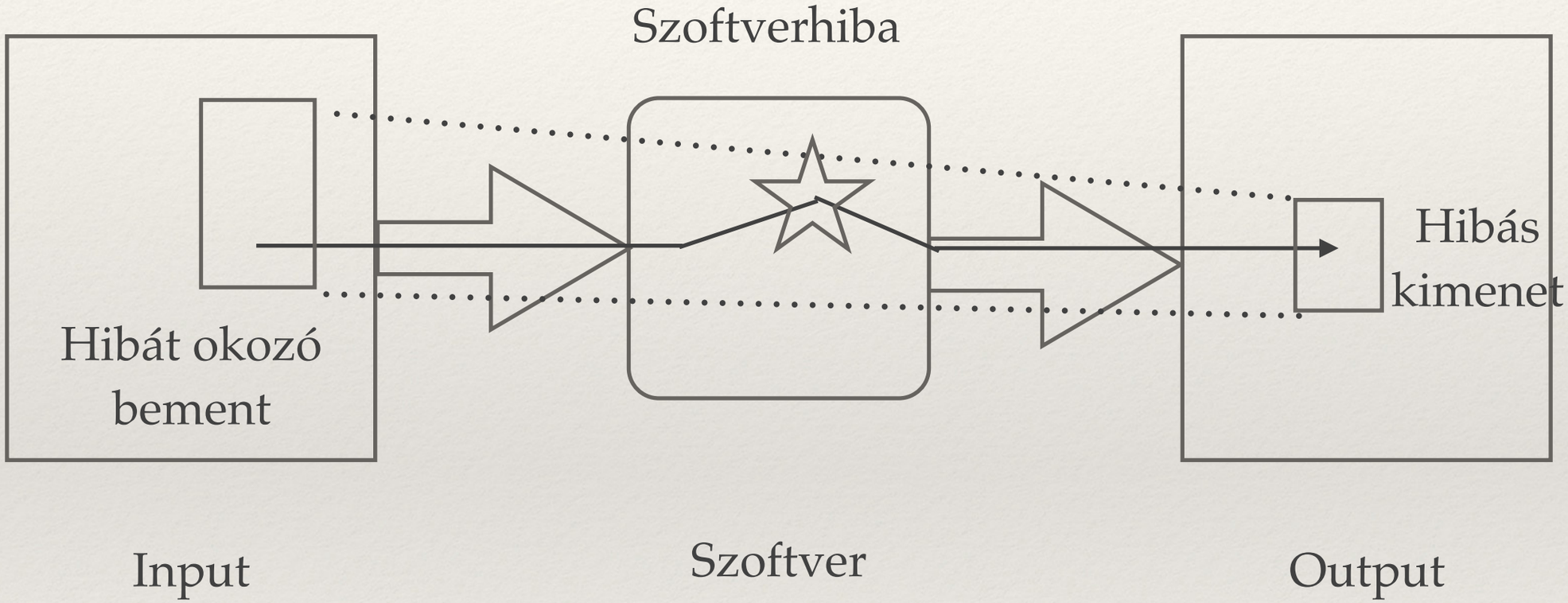
Szoftver-minőségbiztosítás

Specifikáció alapú (black-box) technikák

A szoftver mint leképezés



Funkcionális tesztelés

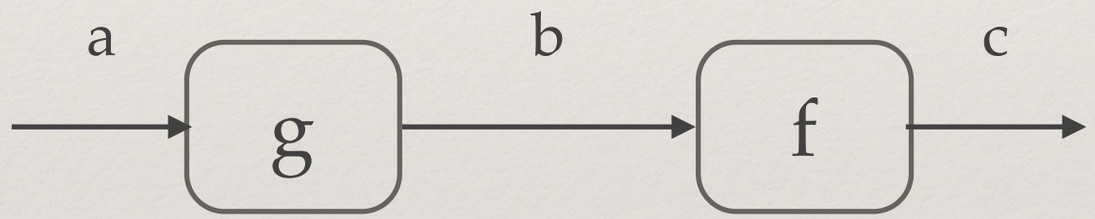
- program, mint input-output leképezés
- **implementáció nem ismert**
  - blackbox megközelítés
- implementáció-változás esetén is használható tesztesetek
- **implementációval párhuzamos teszteset fejlesztés**

Funkcionális tesztek tervezése

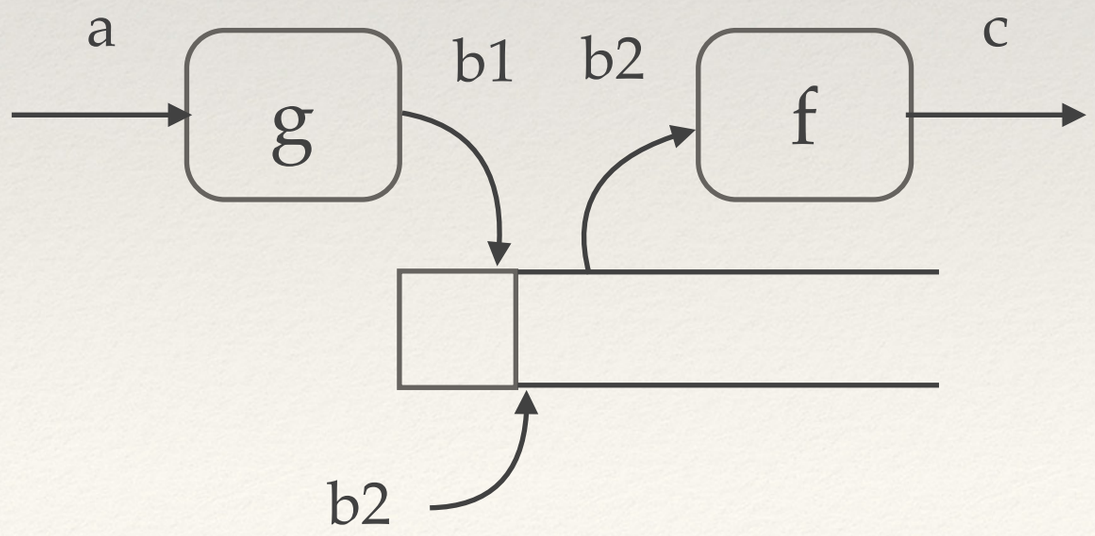
- Felt.: a szoftver működése determinisztikus
  - (many-to-one, one-to-one mapping)
  - Eltérő működés <- eltérő bemenetek
- A tesztelés általában a bemeneti halmazra (értelmezési tart.) fokuszál,
- de ezt kiegészíthetik a kimenet (értékészlet) alapú tesztek
- Cél, hogy **minél kevesebb teszttel, a lehető legszélesebb működési mód halmazt fedjük le**

Függvények összetétele

- Komplex szoftver működés, mint függvények összetétele (procedurák, subrutinok)
  - $(f \circ g)(.) = f(g(.))$
- Kauzális és nem kauzális adatfolyam (összetétel)



Kauzális folyam



Nem kauzális folyam

Ekvivalencia osztály tesztelés

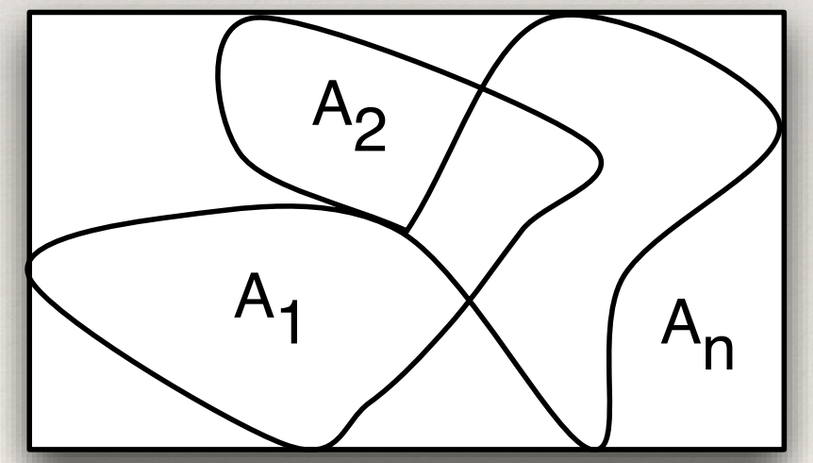
- Cél
  - Kiterjedt tesztelés / redundancia csökkentése
  - Tesztesetek számának csökkentése
    - Ekvivalens tesztek meghatározása
- Módszer
  - bemeneti ekvivalencia osztályok meghatározása
  - tesztesetek meghatározása

Ekvivalencia osztályok

- A működés (funkció) szempontjából "egyforma" bemenetek osztályokat (diszjunkt részhalmazokat, partíciókat) alkotnak

$$A_1 \cup A_2 \cup \ldots \cup A_n = B$$

$$i \neq j \Rightarrow A_i \cap A_j = \emptyset$$



Az ekvivalancia osztályok meghatározása

- Heurisztikus folyamat
- Két fő osztálytípus
  - **érvényes** bemenet
  - **érvénytelen** bemenet

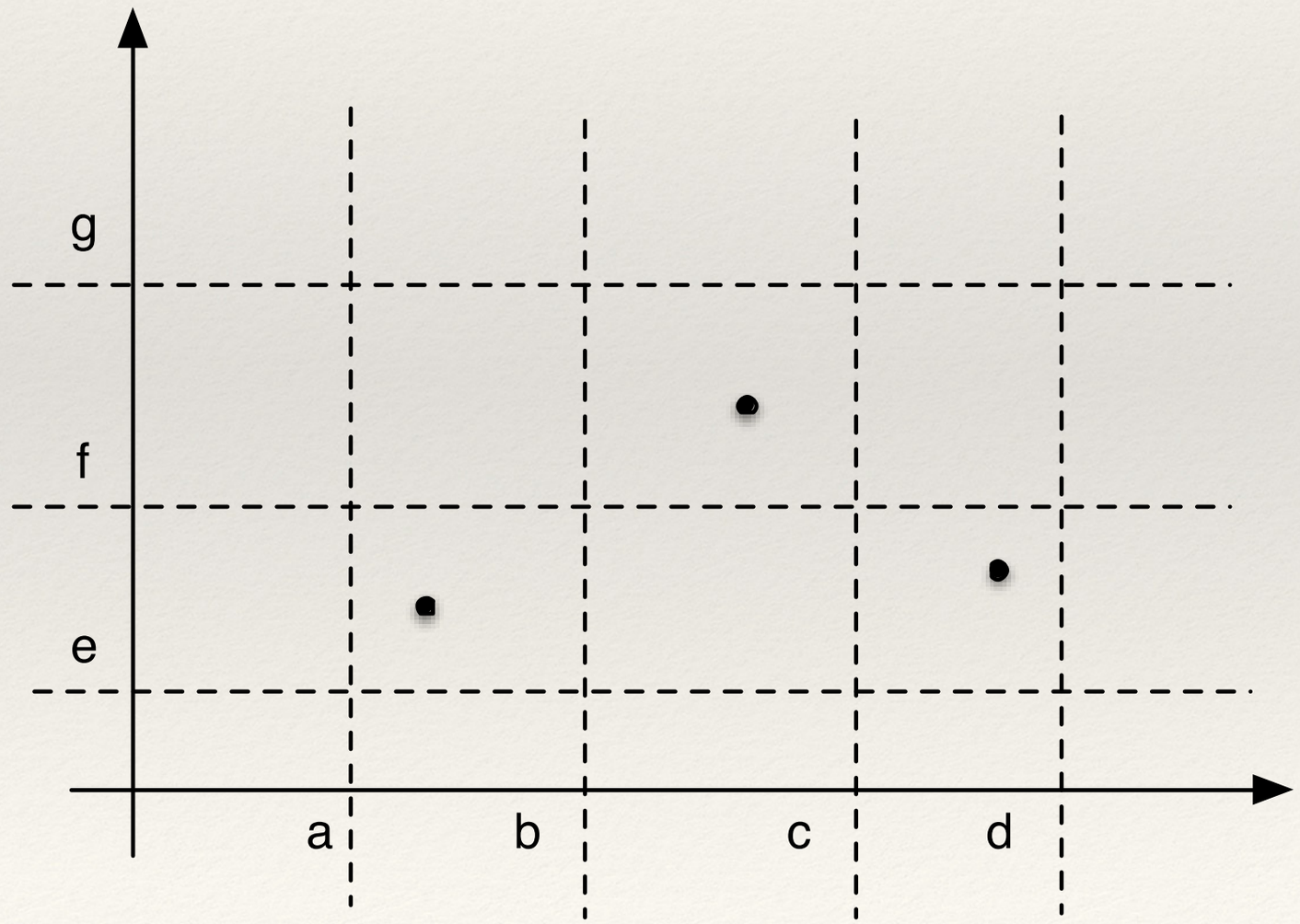
Tesztek meghatározása

- Ekvivalencia osztályok reprezentálása a tesztkészletben
- **Gyenge/erős** tesztkészlet (többszörös vs. egyszerű hibák)
- **Robusztus/normál** tesztkészlet (érvénytelen vs. érvényes esetek)

Ekvivalencia osztály tesztelés (pl.)

Gyenge, normál tesztkészlet

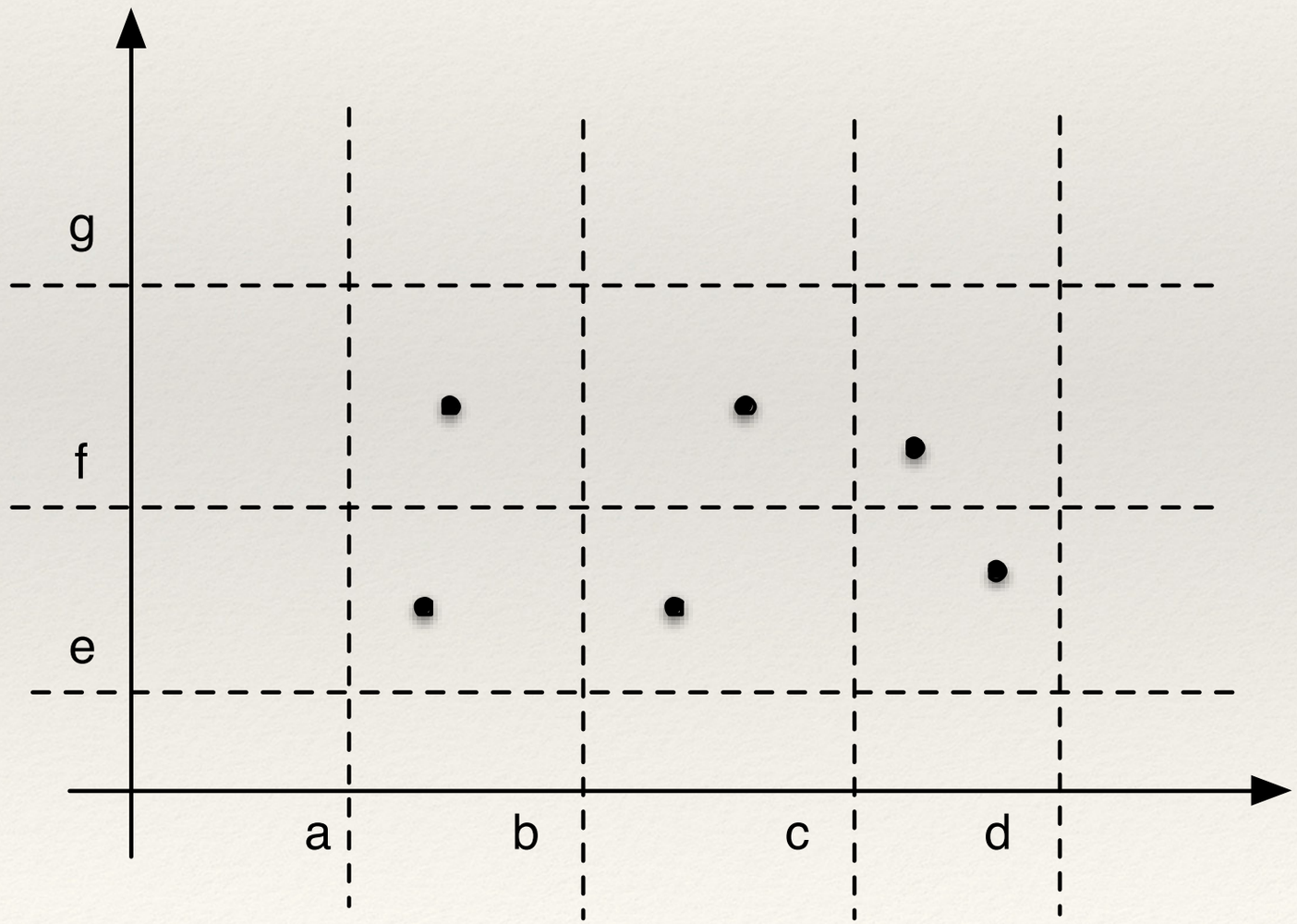
- $a \le x_1 \le d$ [a,b) [b,c) [c,d]
- $e \le x_2 \le g$ [e,f) [f,g]



Ekvivalencia osztály tesztelés (pl.)

Erős, normál tesztkészlet

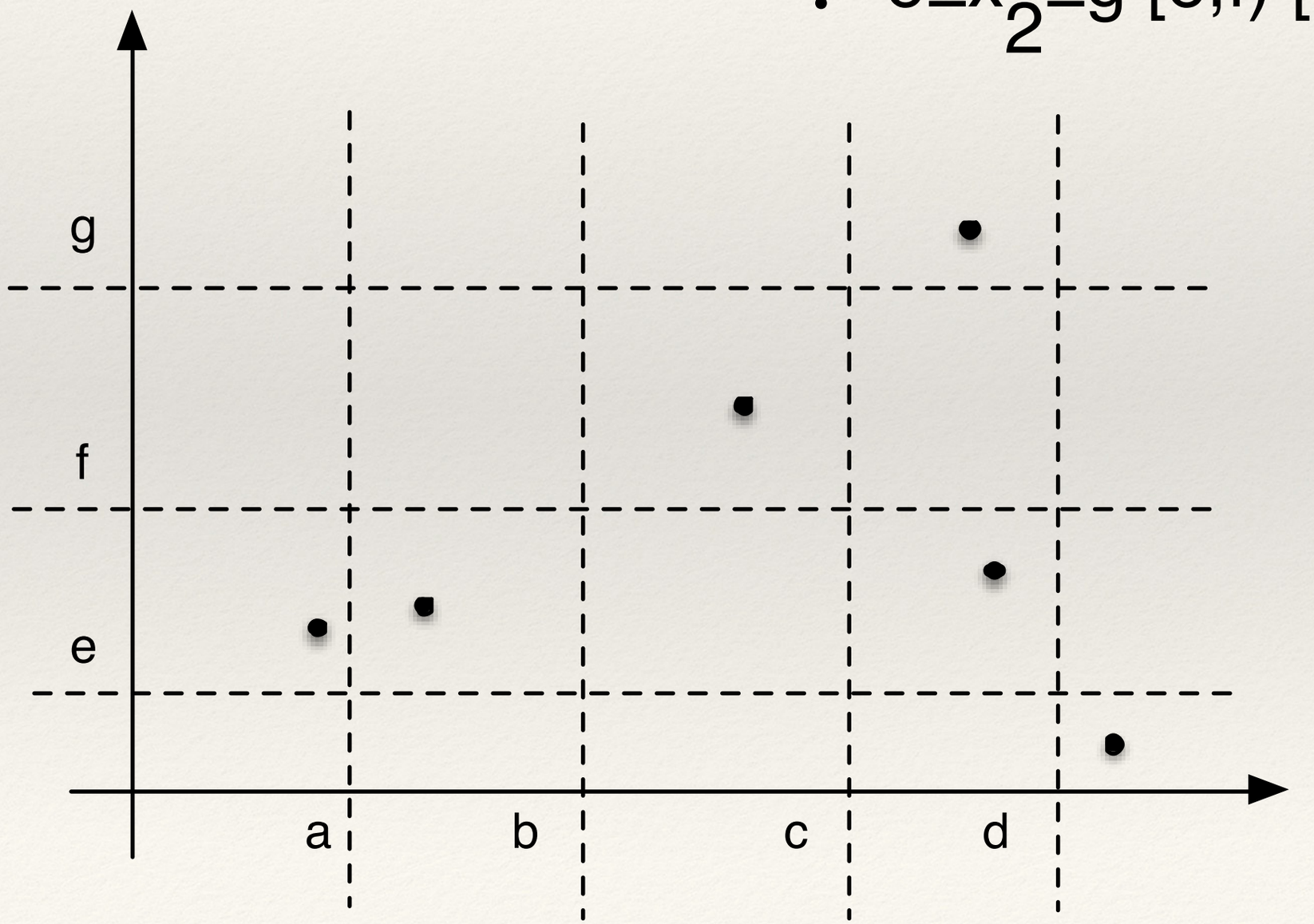
- $a \le x_1 \le d$ [a,b) [b,c) [c,d]
- $e \le x_2 \le g$ [e,f) [f,g]



Ekvivalencia osztály tesztelés (pl.)

Gyenge, robusztus tesztkészlet

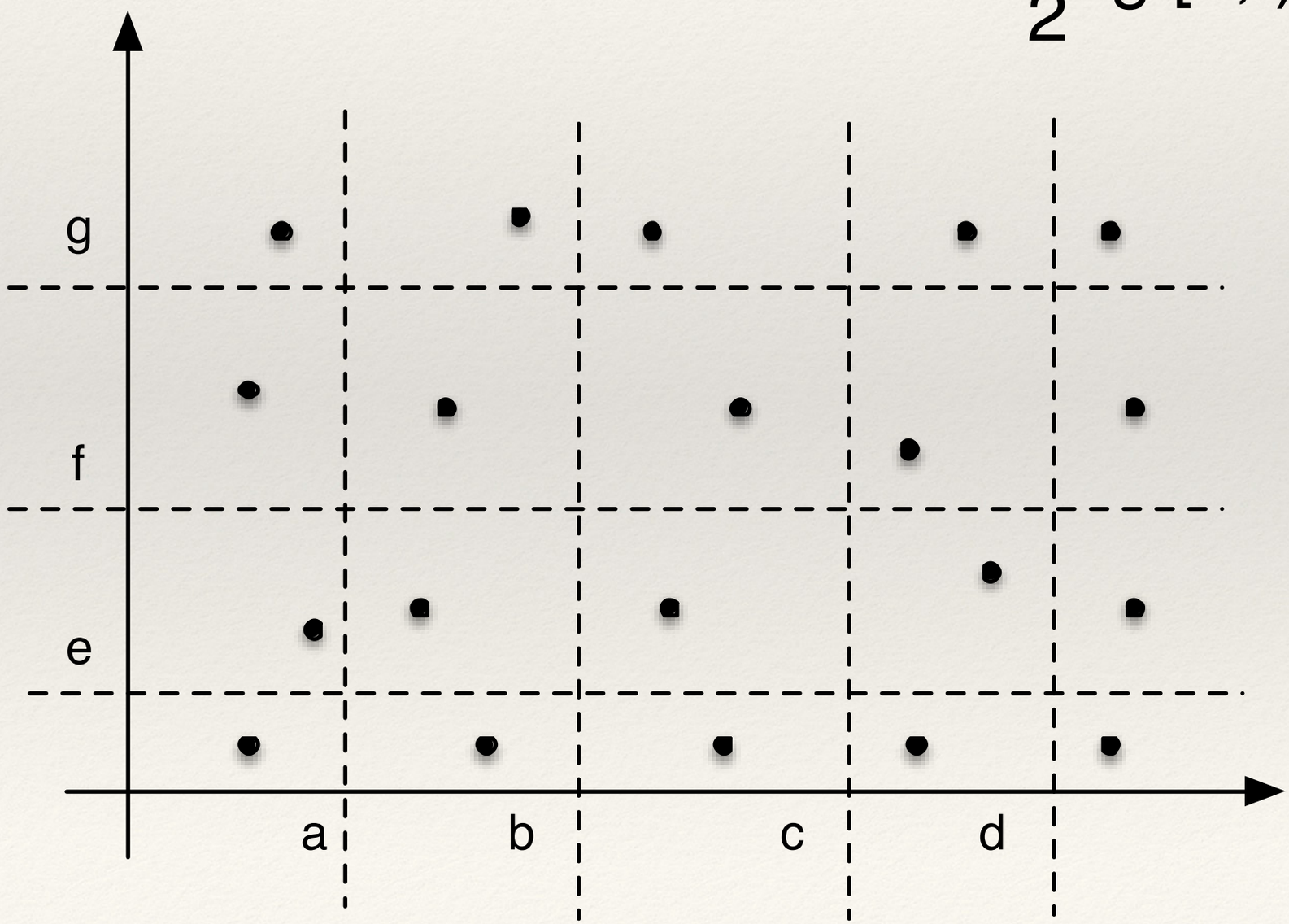
- $a \leq x_1 \leq d$ [a,b) [b,c) [c,d]
- $e \leq x_2 \leq g$ [e,f) [f,g]



Ekvivalencia osztály tesztelés (pl.)

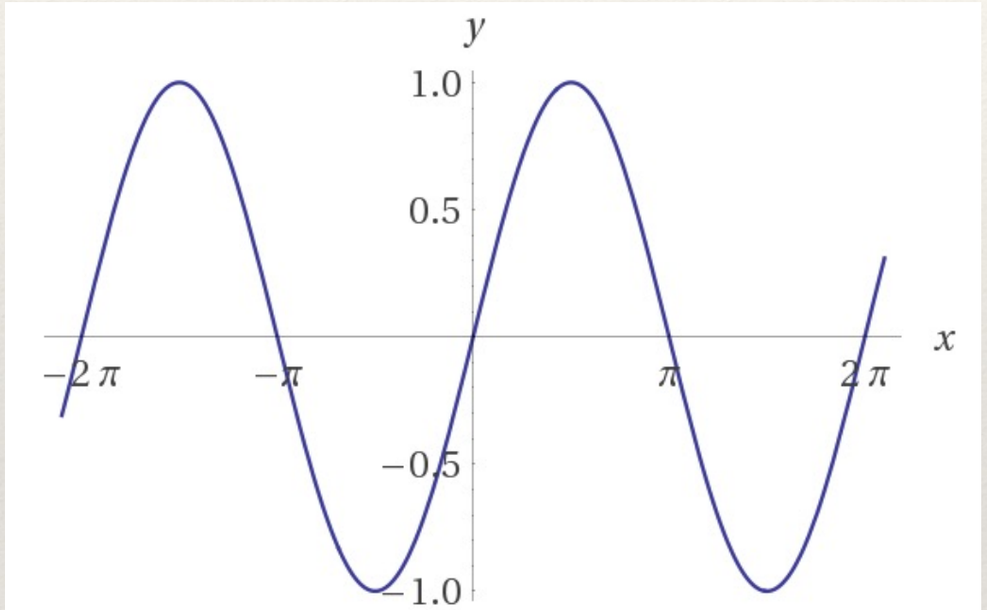
Erős, robusztus tesztkészlet

- $a \le x_1 \le d$ [a,b) [b,c) [c,d]
- $e \le x_2 \le g$ [e,f) [f,g]



Szemléltetés

- Adott egy **`double sine(double x)`** C függvény
  - x ↦ sin(x) közelítése (4 tizedes pont., 7-ed f. Taylor poli.)



Domain

$\mathbb{R}$ (all real numbers)

Range

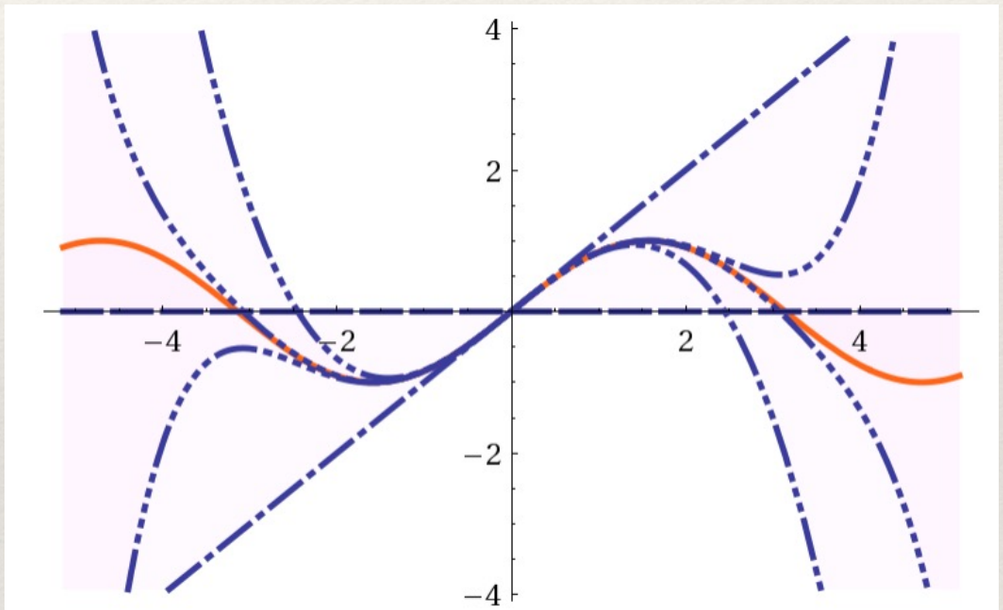
$\{y \in \mathbb{R} : -1 \le y \le 1\}$

Periodicity

periodic in x with period 2π

Parity

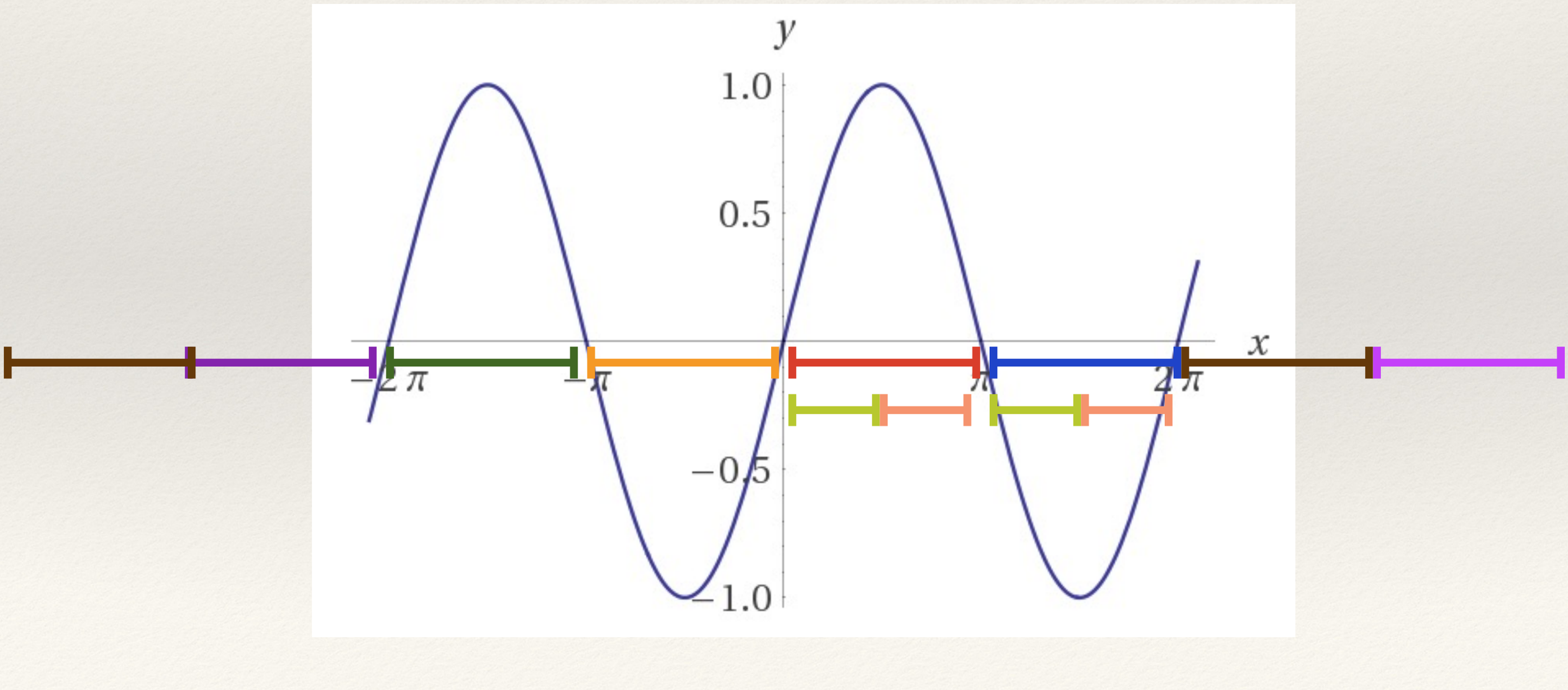
odd



$$\sin(x) = \sum_{k=0}^{\infty} \frac{(-1)^k\, x^{1+2\,k}}{(1+2\,k)!}$$

Szemléltetés (folyt.)

- Feltételezhető ekvivalencia osztályok a sinus fv. tulajdonságai ill. az impl. techn. alapján:

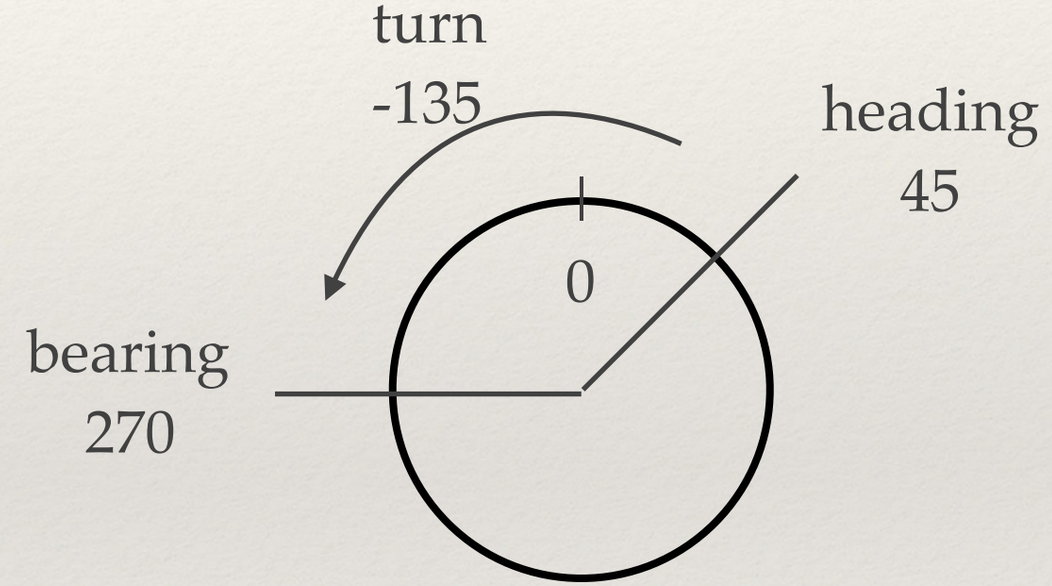


Szemléltetés 2

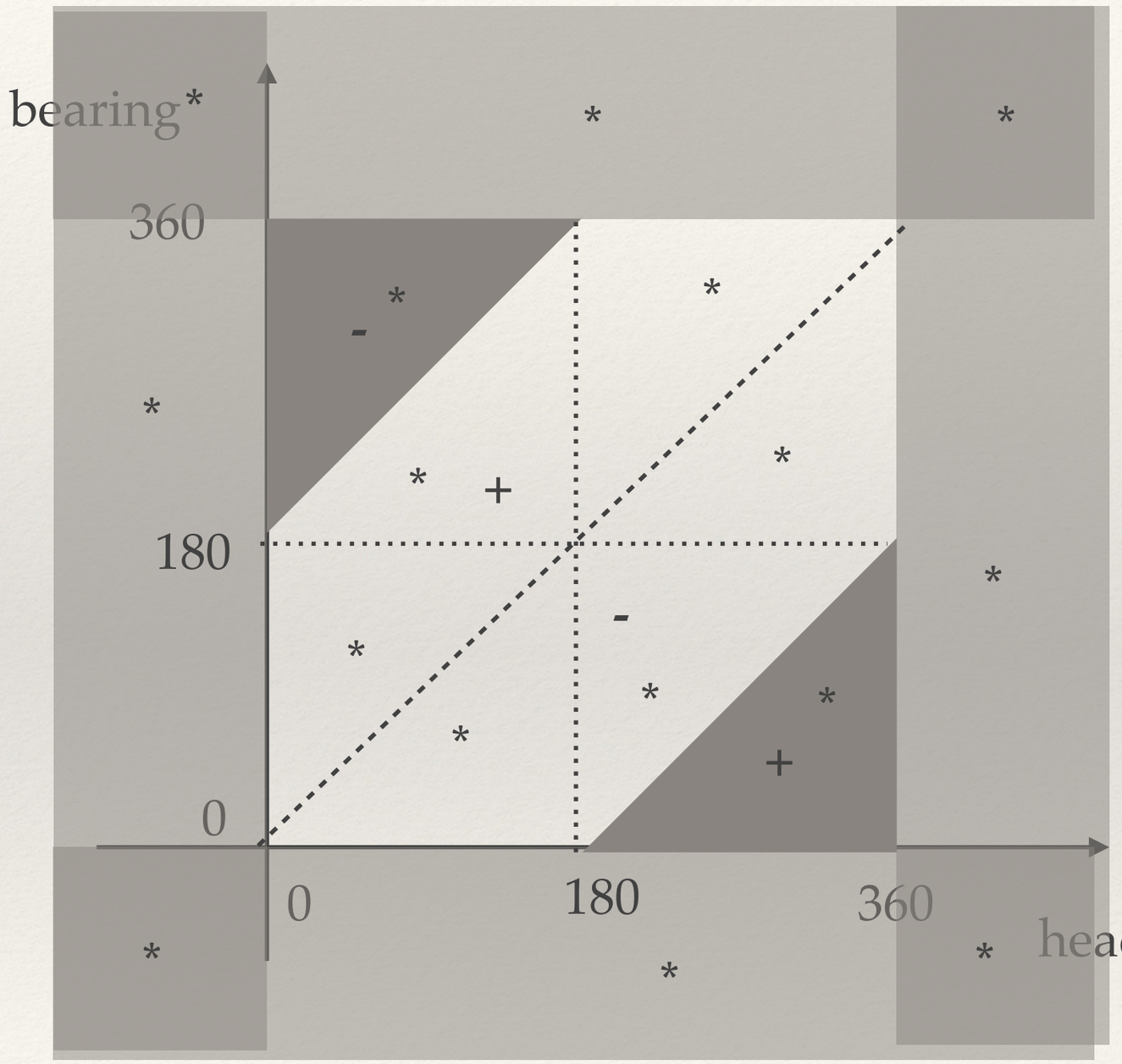
Egy navigációs szoftverben van egy
int turn(int heading, int bearing)
függvény, mely a heading irányszögről a bearing irányszögre fordulás mértékét adja meg. Az irányszögek tájoló irányok és 0-359 (°) tartományban érvényesek. Érvénytelen paraméter esetén a függvény visszatérési értéke -999. Érvényes paraméterek esetén a függvény a kisebb fordulási szöget adja vissza, a visszatérési érték negatív az óramutató járásával ellentétes (balra) és pozitív az óramutató járásával megegyező (jobbra) fordulási irány esetén.

Szemléltetés 2 (folyt.)

- Ekvivalencia osztályok meghatározása
  - 2 paraméter
  - Kisebb szög
  - Irány (előjel)
  - Átfordulás (0°/360°)



Szemléltetés 2 (folyt.)



Érvénytelen

Érvényes osztályok:

abs(bearing-heading)>180
=>másik irány

bearing-heading

* erős, robusztus teszt készlet

Szemléltetés 2 (folyt.)

Egy lehetséges implementáció:

```
int turn(int heading, int bearing){
  int t;
  if(heading<0 || heading>359 || bearing<0 || bearing>359)
    return -999;
  t=bearing-heading;
  if(abs(t)<=180)
    return t;
  else if(t>0)
    return t-360;
  else
    return t+360;
}
```

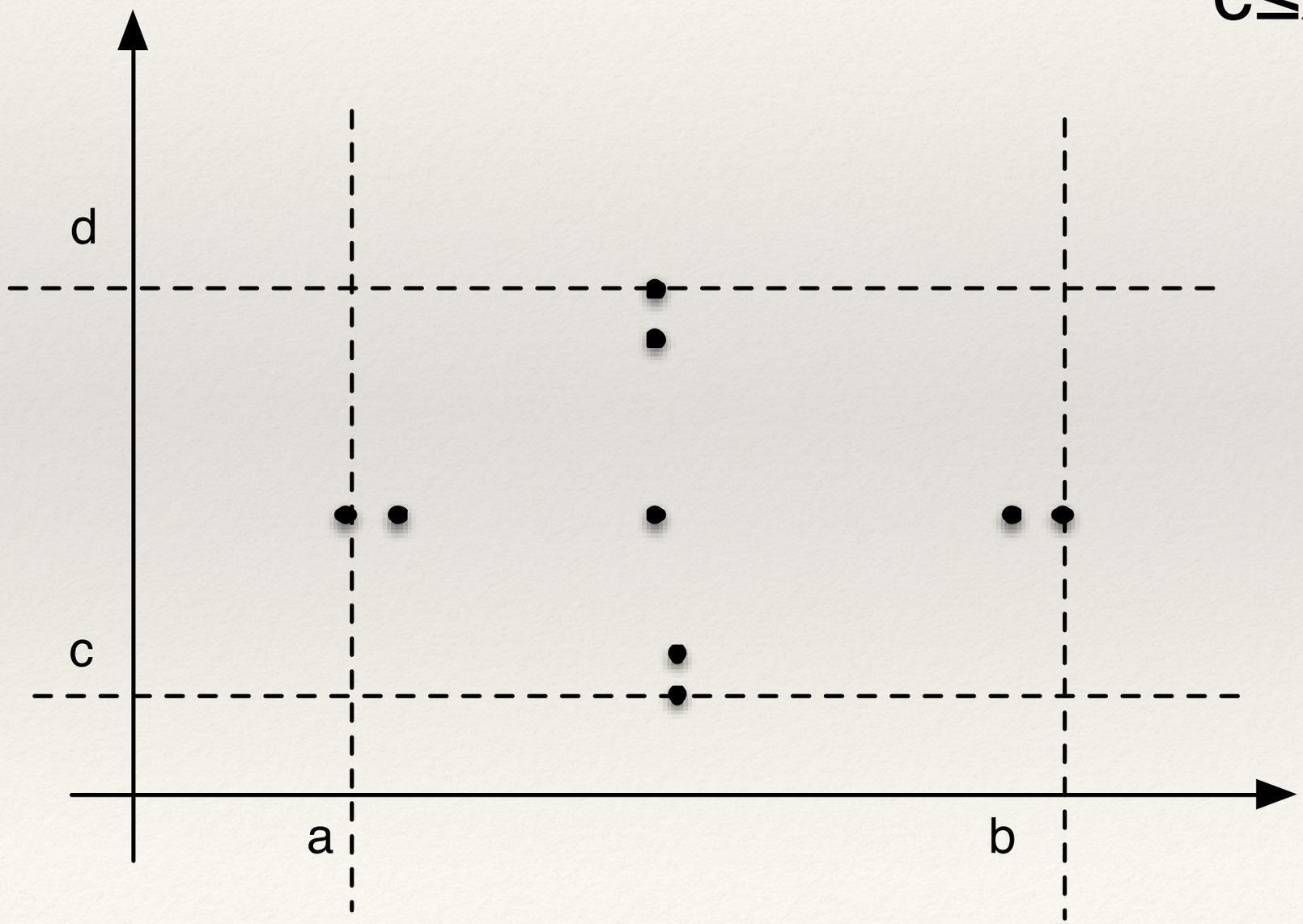
Határérték elemzés

- Hibamodell:
  - gyakran hibás intervallum határérték-kezelés a programokban (meghatározás nehézsége, progr. hiba: indexek, ciklusváltozók, relációs operátorok, stb.)
    - erősen (Ada, CHILL) és gyengén (C) típusos prog. nyelvek (compile/run time range fail handling)
    - változók alul/felül csordulása
- **bemeneti ekvivalencia osztályok határértékei körüli bemenetek**
- **kimeneti határértékeket** is figyelembe kell venni

Határérték elemzés (pl.)

Kétváltozós fv. tesztkészlet

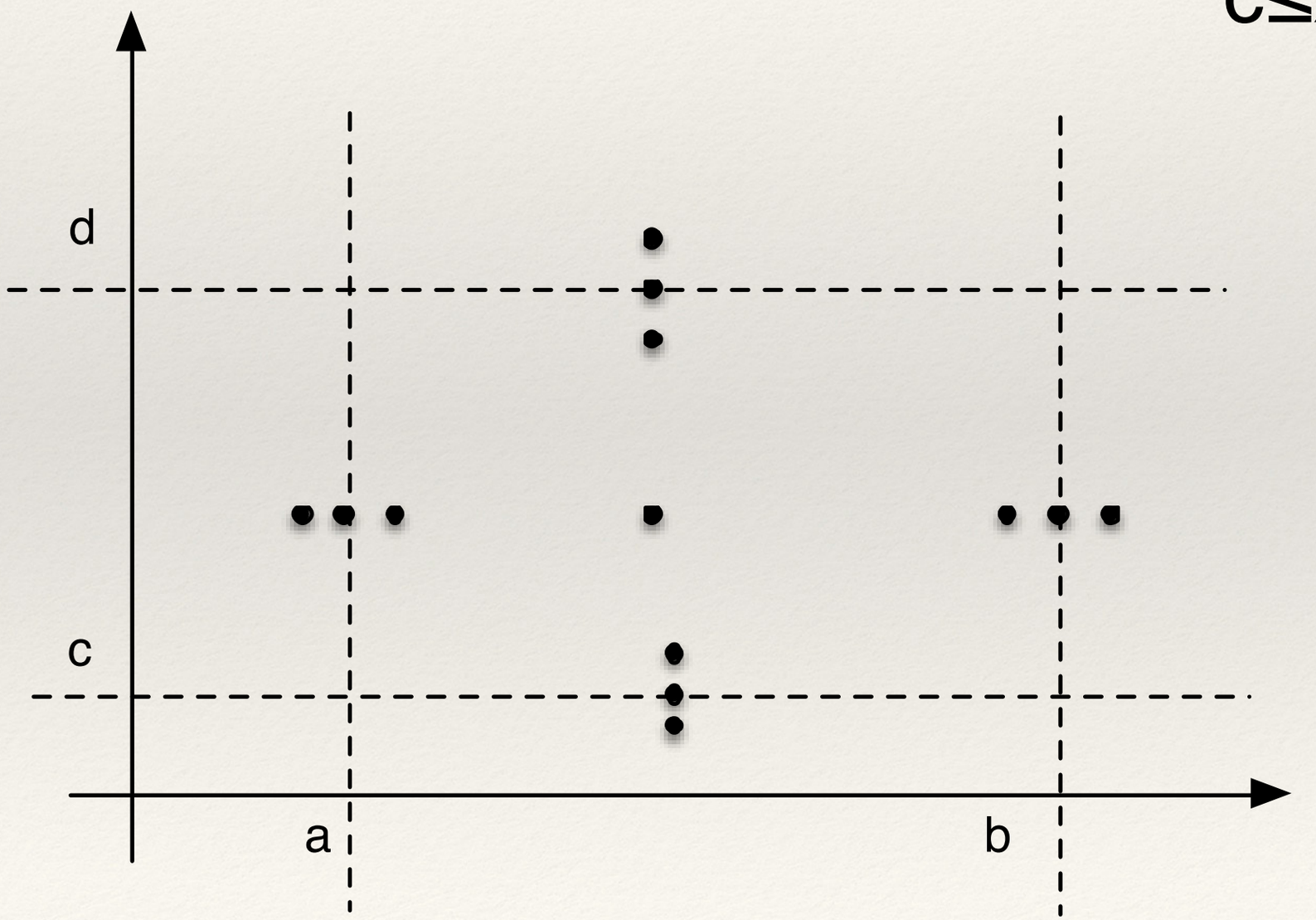
- a≤x1≤b
- c≤x2≤d



Határérték elemzés (pl.)

Robusztus tesztkészlet

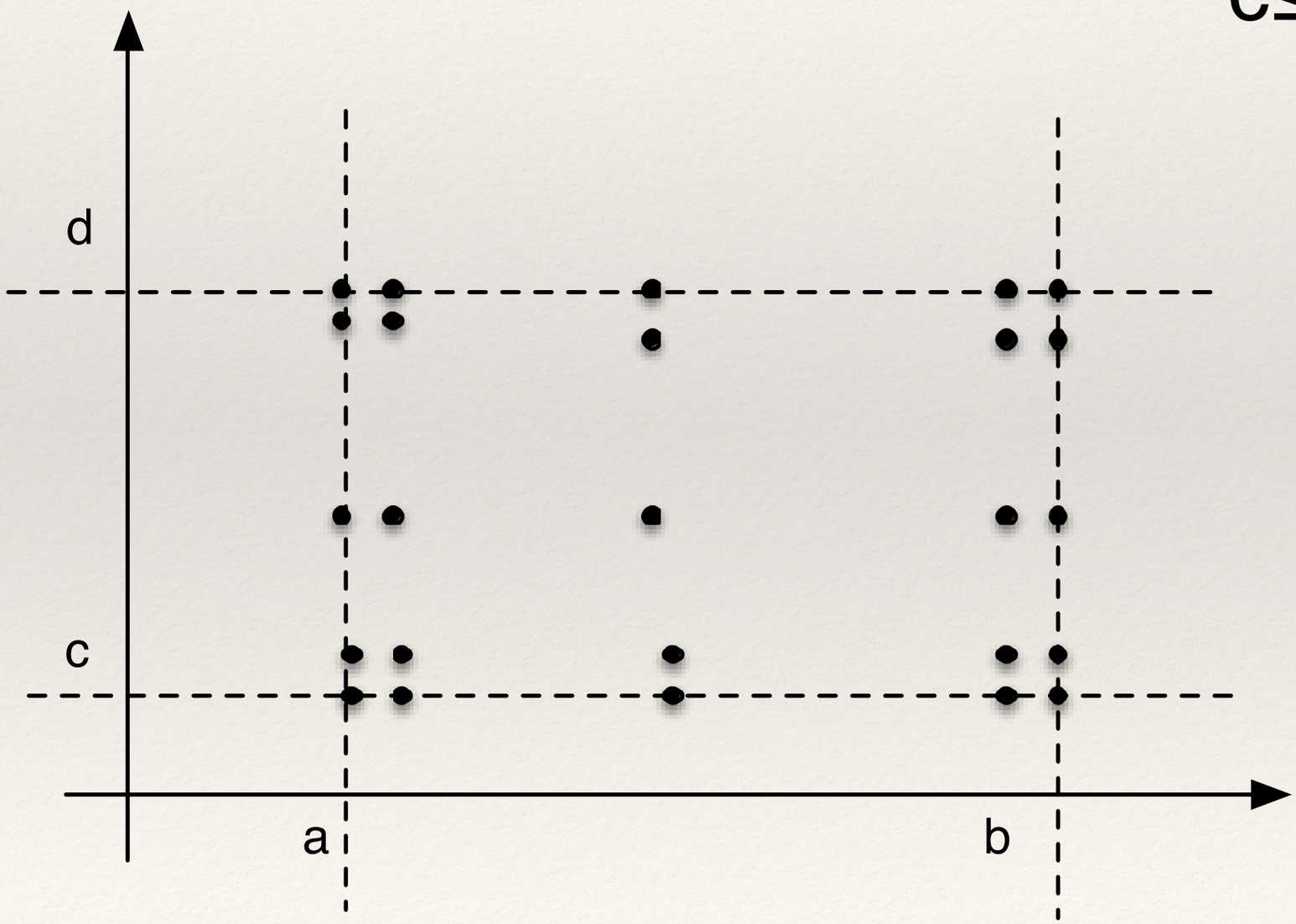
- a≤x1≤b
- c≤x2≤d



Határérték elemzés (pl.)

Legrosszabb eset tesztkészlet

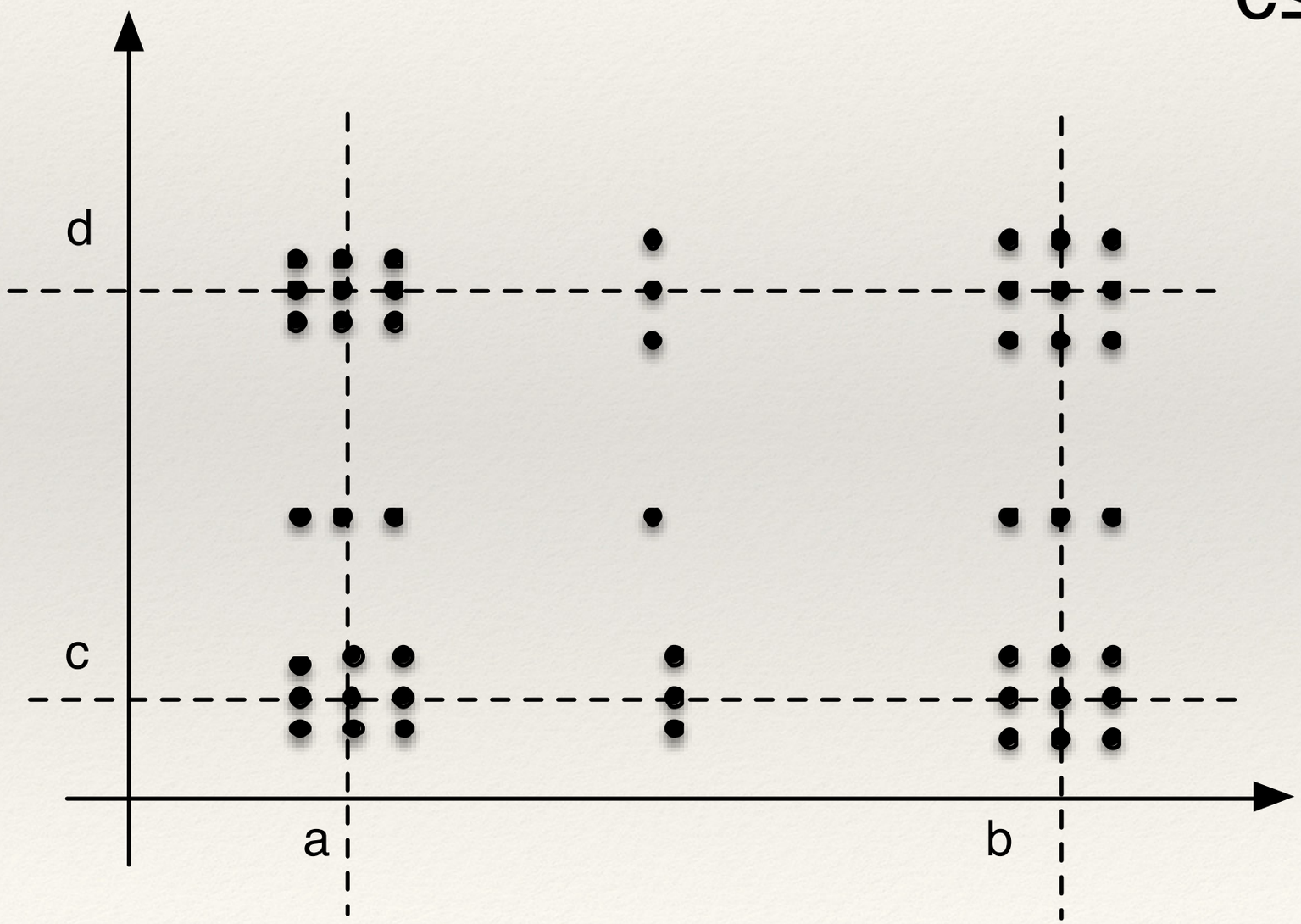
- a≤x1≤b
- c≤x2≤d



Határérték elemzés (pl.)

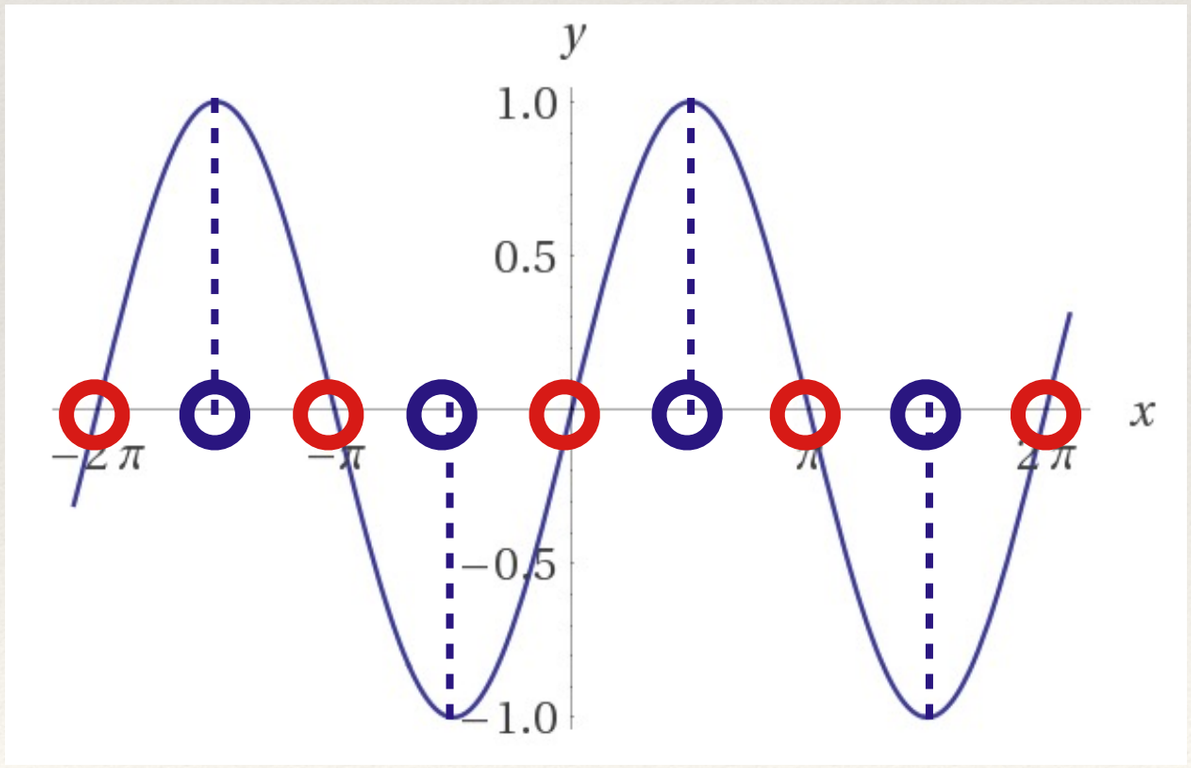
Robusztus legrosszabb eset tesztkészlet

- $a \leq x1 \leq b$
- $c \leq x2 \leq d$



Szemléltetés (folyt.)

- Határérték tesztesetek
  - Input domén
  - Output terjedelem



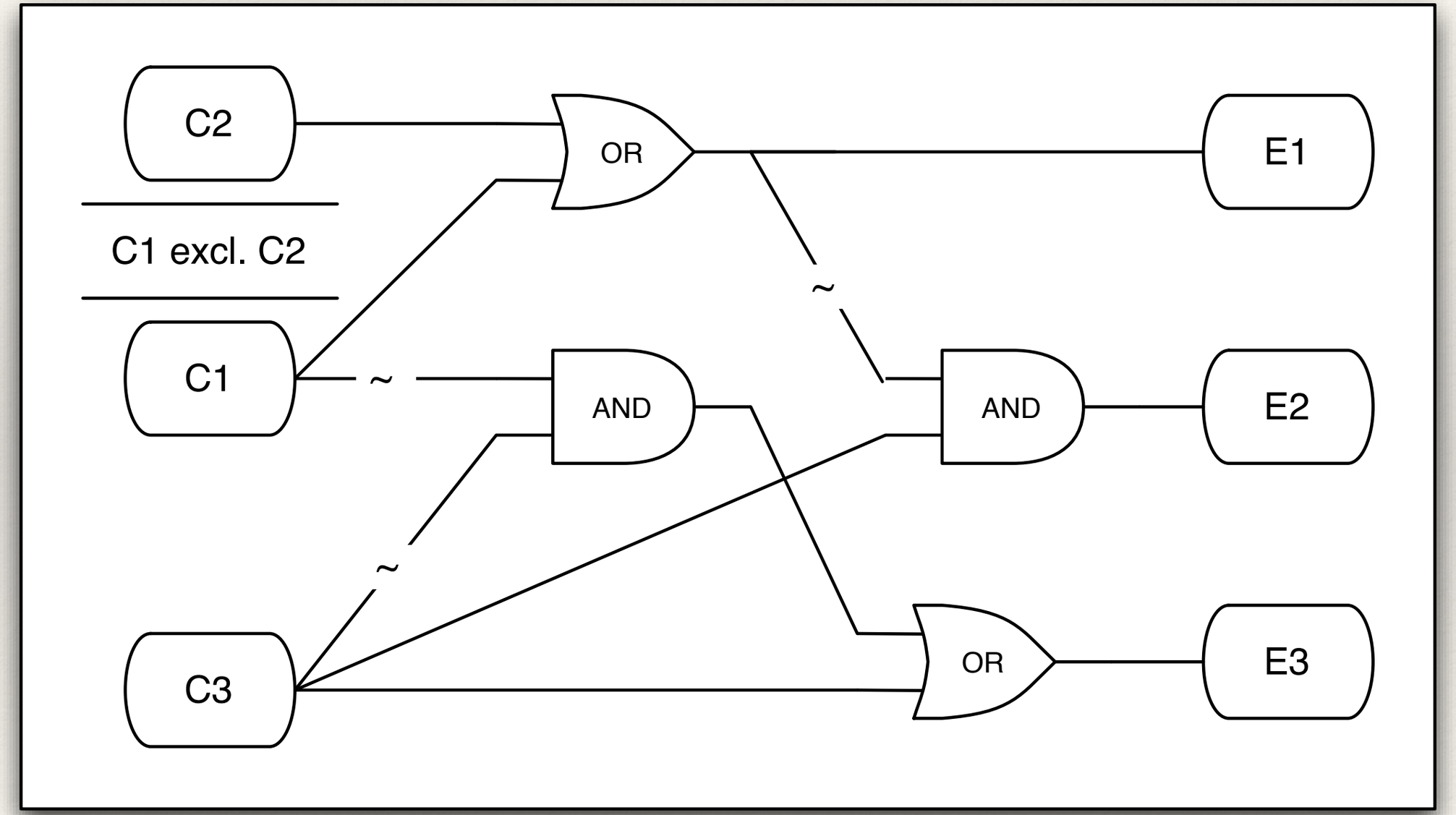
Döntési tábla (ok-hatás) elemzés

- **bool gráf**
  - bináris bemeneti feltételek (cause)
  - bináris kimeneti feltételek (effect/action)
  - logikai műveletek (és, vagy, nem)
- **kiegészítő megszorítások**
  - **E**xclusive, **I**(at least one), **O**ne and only one, **R**equires,
  - **M**asks
- => **döntési táblázat**
  - => tesztek leszármaztatása

Döntési tábla és tesztelés

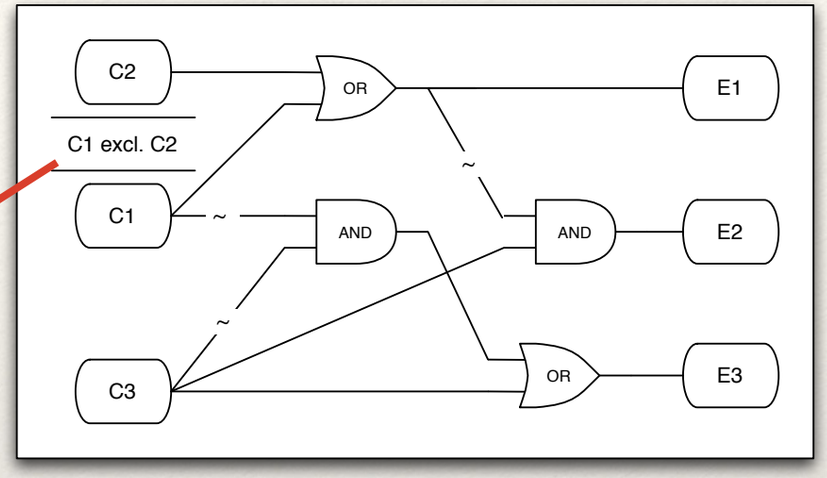
- Komplex üzleti logika (szabályok) **szisztematikus áttekintése** (fejlesztők, tesztelők)
- Jól használhatók a teszttervezésben. Segíti a tesztelőket különböző bemeneti kombinációk, szoftver állapotok hatásainak feltárásában.
- **Minden kombináció tesztelése nem mindig kivitelezhető.** A szabályok részhalmaza is tesztelhető: priorizálás, hatékony tesztesetek kiválogatása.

Bool gráf (pl.)



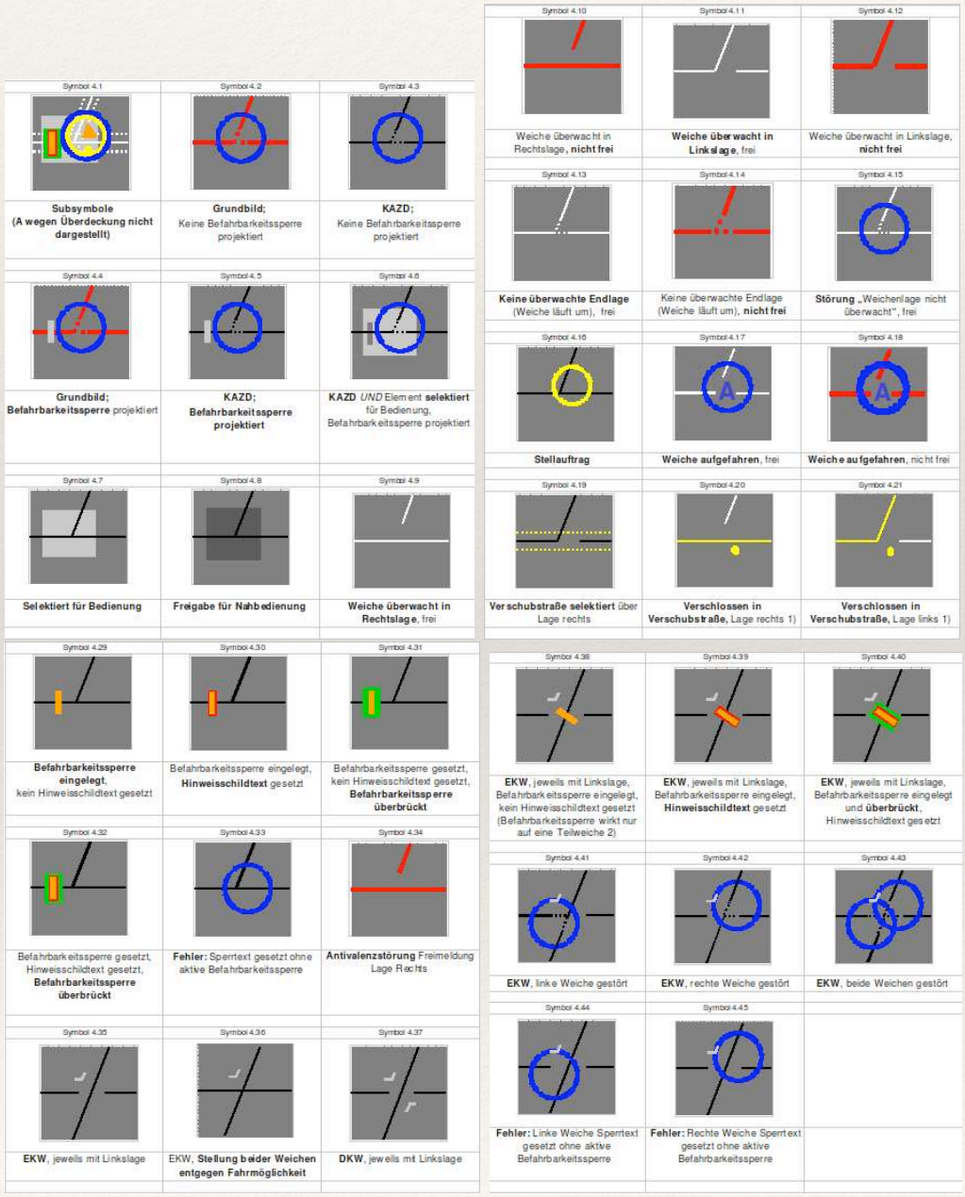
Döntési táblázat

R	1	2	3	4	5	6	7	8
C1	F	F	F	T	T	F	T	T
C2	F	F	T	F	F	T	T	T
C3	F	T	F	F	T	T	F	T
E1	F	F	T	T	T	T	*	*
E2	F	T	F	F	F	F	*	*
E3	T	T	T	F	T	T	*	*



Ekviv. oszt.

Döntési tábla (pl.)




```
DecTabBegin SwitchPosition
Determine the position for ElementSwitch.
heartDashed is initialized to false.
leftShort and rightShort are initialized to true.
n n n n n n n - | mySwitch.getUndefined()
n y n n n - y - | mySwitch.getDisturbed()
y y n n n - - - | mySwitch.getForced()
- - y n n - - - | mySwitch.getMoving()
- - - y n y - - | mySwitch.getLeft()
- - - n y y - - | mySwitch.getRight()
================|==========================
- - x - - x x x | heartDashed = true;
- - x - - x x x | heartCol = trackColor;
x x - - - x x x | dRing = Color.BLUE;
- - - - - x - x | isUndef = true;
x x - - - - - - | theA = Color.BLUE;
- - - x - - - - | leftShort = false;
- - - - x - - - | rightShort = false;
- - - x - - - - | rightColor = secondTrackColor;
- - - - x - - - | leftColor = secondTrackColor;
- - - - x - - - | rightColor = trackColor;
- - - x - - - - | leftColor = trackColor;
DecTabEnd SwitchPosition
```

Módszerek összehasonlítása

Véletlen teszt generálás

- korlátozott hibalefedési képesség
- tesztesetek előállítása **nagy tömegben**
- determinisztikus tervezési hibák felfedése

Valószínűségi funkcionális teszt generálás

- Tesztesetek generálása valószínűségi alapon
  - bemeneti értékkészlet eloszlása (**működési profilból**)
  - funkcionalitás
- **Kockázat** és hiba valószínűség -> tesztesetek száma

Állapotátmenet tesztelés

- Szoftver (komponens) működése véges állapotgéppel (FSM) leírható
  - állapotok, átmenetek, események, tevékenységek
- Érvényes állapotátmenetek tesztelése (**átmenet lefedettség**)
- Állapot szekvenciák végigkövetése (**forgatókönyv**)
- Összes állapot elérése (**állapot lefedettség**)
- Események tesztelése különböző állapot kontextusokban (**kontextus lefedettség**)
- Érvénytelen átmenetek tesztelése

Use-case tesztelés

- Use-case: szereplők, rendszer interakciók
  - Use-case lefutások - aktivitás diagrammok (lépések sorozatai), feltételek

- Teljes rendszer tranzakcióinak végigkövetése a tesztekben (**szokásos és alternatív lefutások**)
  - **rendszer- és átvételi tesztek**
  - integrációs megközelítés