

Alegerea cazurilor de test în testarea funcțională -2

Prof. univ. dr. ing. Florica Moldoveanu

Testarea bazata pe clase de echivalență (1)

- ❖ O clasă de echivalență a domeniului de intrare al unității testate este alcătuită din toți vectorii de test care produc aceeași comportare a unității testate (funcție/program).
- Clasele de echivalență rezultă din predicatele definite asupra valorilor variabilelor de intrare.

Testarea bazata pe clase de echivalență (Equivalence class testing):

- Pentru testarea minimală este suficient sa se foloseasca un singur caz de test pentru fiecare clasă de echivalență → Numarul de cazuri de test este mai mic decat în cazul folosirii tuturor combinatiilor între valorile selectate din domeniile variabilelor de intrare.

Testarea valorilor de frontieră (Boundary Value Testing)

- ❖ Se aleg cazuri de test cu valori la frontierele claselor de echivalență ale variabilelor de intrare: pe frontiere, langa frontiere în afara clasei de echivalență și în interiorul clasei de echivalenta.
- Permite detectarea cazurilor de implementare eronata a conditiilor de frontiera (predicate gresite)
- Considerarea CT cu valori de frontiera crește numărul de CT pentru fiecare clasă de echivalență.

Testarea bazata pe clase de echivalenta (2)

1. Program/unitate program cu o singura variabila de intrare

Unitatea testata calculeaza pretul unei calatorii cu autobuzul astfel:

0 lei pentru copii cu varsta sub 7 ani

2 lei pentru tineri cu varsta intre 7 si 18 ani

5 lei pentru adulti (19-64)

3 lei pentru cei cu varsta peste 64

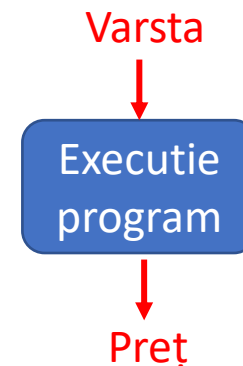
Clasele de echivalenta ale domeniului de intrare:

CE1. $0 \leq \text{varsta} < 7$

CE2. $7 \leq \text{varsta} < 19$

CE3. $19 \leq \text{varsta} < 65$

CE4. $65 \leq \text{varsta} < 101$



Cazuri de test – testarea minimala (normala):

CT1. varsta=4; RA (rezultat asteptat): pret = 0

CT2. varsta=10; RA: pret = 2

CT3. varsta=30; RA: pret = 5

CT4. varsta=70; RA: pret = 3

Cazuri de test – testarea minimala robusta:

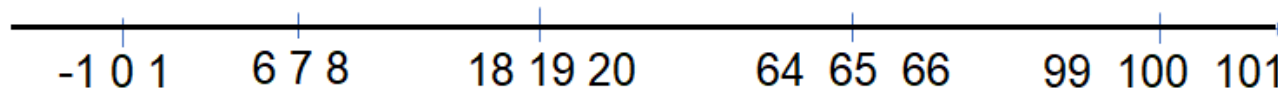
Se adauga cazuri de test cu valori de intrare nevalide.

CT5. varsta=-1; RA: mesaj “varsta nevalida”

CT6. varsta=110; RA: mesaj “varsta nevalida”

Testarea bazata pe clase de echivalenta (3)

Cazuri de test – testarea cu valori la frontierele claselor de echivalenta:



CTF1. varsta=0; RA: pret = 0 CTF2. varsta = -1; RA: “varsta nevalida” CTF3. varsta=1; RA: pret = 0

CTF4. varsta=6, RA: pret = 0 CTF5. varsta=7; RA: pret = 2 CTF6: varsta=8, RA: pret = 2

CTF7. varsta=18; RA: pret = 2 CTF8: varsta=19; RA: pret = 5 CTF9: varsta=20; RA: pret = 5

CTF10. varsta=64; RA: pret = 5 CTF11: varsta=65; RA: pret = 3 CTF12: varsta=66; RA: pret = 3

CTF13. varsta=99; RA: pret = 3 CTF14. varsta=100; RA: pret = 3

CTF15. varsta = 101; RA: “varsta nevalida”

Testarea bazata pe clase de echivalenta (4)

2. Program cu doua variabile de intrare

$10 \leq i1 \leq 100$, cu subintervalele: $[10, 20)$, $[20, 70)$, $[70, 100]$ \rightarrow 3 clase de echivalenta

$20 \leq i2 \leq 50$, cu subintervalele: $[20, 30)$ și $[30, 50]$ \rightarrow 2 clase de echivalenta

➤ Comportarea programului depinde de clasele de echivalență din care fac parte valorile variabilelor de intrare. **Clasele de echivalență ale domeniului de intrare (DI) al programului:**

CE1: $10 \leq i1 \leq 20$ și $20 \leq i2 \leq 30$; CE2: $10 \leq i1 \leq 20$ și $30 \leq i2 \leq 50$;

CE3: $20 < i1 \leq 70$ și $20 \leq i2 \leq 30$;

Cazuri de test – testarea minimală:

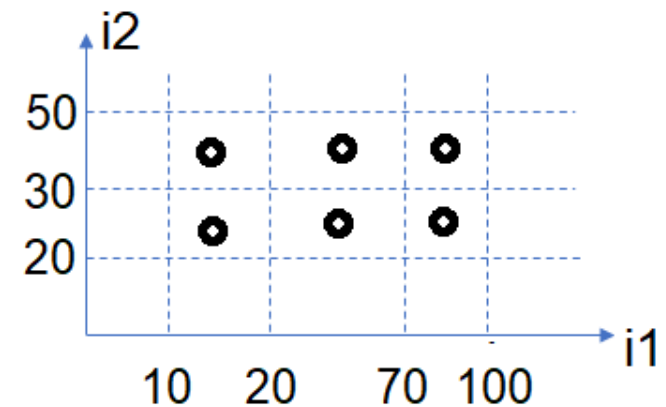
Un caz de test pentru fiecare clasă de echivalență a

DI al programului:

CT1. $i1=15, i2=25$ CT2. $i1=17, i2=40$

CT3. $i1 = 50, i2 = 23$ CT4. $i1=60, i2= 45$

CT5. $i1 = 80, i2 = 26$ CT6. $i1=90, i2=35$



Testarea bazata pe clase de echivalenta (5)

Cazuri de test – testarea minimala robustă:

Se adaugă cazuri de test cu valori din afara domeniilor variabilelor de intrare

CT1. $i_1=15, i_2=25$ CT2. $i_1=17, i_2=40$

CT3. $i_1 = 50, i_2 = 23$ CT4. $i_1=60, i_2= 45$

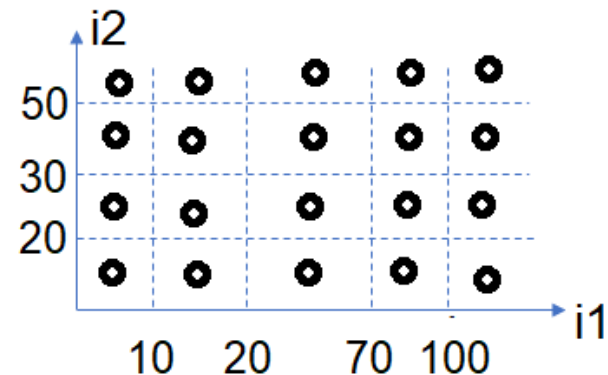
CT5. $i_1 = 80, i_2 = 26$ CT6. $i_1=90, i_2=35$

CT7. $i_1 = 5, i_2 = 10$ CT8. $i_1= 3, i_2 = 24$

CT9. $i_1 = 4, i_2 = 36$ CT10. $i_1= 1, i_2 = 55$

.....

CT19. $i_1=110, i_2 = 3$ CT20. $i_1=120, i_2 = 60$



Cazuri de test – testarea extinsă

- Mai multe cazuri de test pentru fiecare clasă de echivalență

Testarea bazata pe clase de echivalenta (6)

3. Triunghiul

Unitatea testata primeste 3 numere, a, b, c, si verifica daca ele pot reprezenta lungimile laturilor unui triunghi; a,b,c au valori in intervalul (0,200]. Rezultatul este unul dintre mesajele:

- triunghi echilateral/ isoscel/oarecare, nu este un triunghi

De asemenea, se afiseaza mesaje specifice atunci cand valorile variabilelor de intrare sunt in afara domeniului.

Clasele de echivalenta ale DI al programului sunt determinate de urmatoarele predicate:

P1: $a > 0 \ \&\& \ b > 0 \ \&\& \ c > 0 \ \&\& \ a \leq 200 \ \&\& \ b \leq 200 \ \&\& \ c \leq 200$ - domeniul variabilelor de intrare

P2: $a + b > c \ \&\& \ a + c > b \ \&\& \ b + c > a$ - conditia ca a,b,c sa reprezinte laturile unui triunghi

P3: $a == b \ \&\& \ b == c$ - triunghi echilateral

P4: $a == b \ \&\& \ b != c \ || \ a == c \ \&\& \ c != b \ || \ b == c \ \&\& \ c != a$ - triunghi isoscel

P5: $a != b \ \&\& \ a != c \ \&\& \ b != c$ - triunghi oarecare

Clasele de echivalenta - toate combinatiile de valori a,b,c care satisfac unul dintre predicatele

CE1. P1 && P3: triunghi echilateral

CE2. P1 && P2 && P4: triunghi isoscel

CE3. P1 && P2 && P5: triunghi oarecare

CE4. P1 && !P2: nu este un triunghi

Testarea bazata pe clase de echivalenta (7)

Cazuri de test – testarea minimala: un caz de test pentru fiecare clasa de echivalenta

CT	a	b	c	Rezultat asteptat
1	10	10	10	Triunghi echilateral
2	5	5	2	Triunghi isoscel
3	7	6	9	Triunghi oarecare
4	4	2	1	Nu este un triunghi

Cazuri de test – testarea minimala robusta:

Se adauga cazuri de test cu valori in afara domeniilor variabilelor a, b, c:

CT	a	b	c	Rezultat asteptat
5	-1	10	10	Valoare nevalida pentru a
6	5	-1	2	Valoare nevalida pentru b
7	7	6	0	Valoare nevalida pentru c
8	201	4	1	Valoare nevalida pentru a
9	12	201	10	Valoare nevalida pentru b

Testarea bazata pe clase de echivalenta (8)

Cazuri de test – testarea minimala robusta:

CT	a	b	c	Rezultat asteptat
10	1	40	201	Valoare nevalida pentru c
11	0	0	2	Valoare nevalida pentru a si b
12	-1	6	0	Valoare nevalida pentru a si c
13	10	-1	-1	Valoare nevalida pentru b si c
14	0	0	0	Valoare nevalida pentru a, b si c

Cazuri de test – testarea extinsa: mai multe cazuri de test pentru fiecare clasa de echivalenta

CT	a	b	c	Rezultat asteptat
1	1	1	1	Triunghi echilateral
2	100	100	100	Triunghi echilateral
3	200	200	200	Triunghi echilateral

Testarea bazata pe clase de echivalenta (9)

Cazuri de test – testarea extinsa:

CT	a	B	c	Rezultat asteptat
4	50	50	100	Triunghi isoscel
5	70	80	70	Triunghi isoscel
6	10	40	40	Triunghi isoscel
7	2	3	4	Triunghi oarecare
8	60	20	50	Triunghi oarecare
9	199	200	198	Triunghi oarecare
10	24	1	25	Nu este un triunghi
11	2	1	1	Nu este un triunghi
12	199	1	198	Nu este un triunghi

Cazuri de test – testarea extinsa robusta: se adauga cazuri de test cu valori nevalide pentru a, b, c.

Testarea bazata pe clase de echivalenta (10)

Metoda partitionarii in clase de echivalenta (CE) poate fi extinsa si asupra variabilelor de iesire:

pentru fiecare CE a domeniului de iesire se creaza un caz de test cu date de intrare care produc un rezultat apartinand unei CE de iesire.

Exemplu: Fie un program care trebuie sa produca intre 3 si 6 numere cuprinse intre 1000 si 2500.

Atunci, se vor alege intrari astfel incat iesirea programului sa fie:

- 3 numere egale cu 1000
- 3 numere egale cu 2500
- 6 numere egale cu 1000
- 6 numere egale cu 2500
- intre 3 si 6 numere cuprinse intre 1000 si 2500
- rezultat eronat: mai putin de 3 numere sau mai mult de 6 numere sau valori in afara intervalului [1000..2500]

In alegerea cazurilor de test trebuie sa se elimine redundanțele rezultate din

considerarea atat a claselor de echivalenta de intrare cat si a celor de iesire.

Testarea bazata pe clase de echivalenta (11)

4. Testarea unei functii pentru rezolvarea ecuatiei de gradul 2

Specificația funcției:

`void ec_grad2(float a, float b, float c)`

Funcția rezolvă ecuația de gradul 2: $a \cdot x^2 + b \cdot x + c = 0$

1) $a \neq 0 \ \&\& \ b \neq 0 \ \&\& \ c \neq 0$

2) Dacă $b^2 - 4ac > 0$ atunci funcția afișează mesajul “rădăcini reale” și valorile celor 2 rădăcini reale, sub forma: “x1=” valoare , “x2=” valoare.

3) Dacă $b^2 - 4ac = 0$ atunci funcția afișează mesajul “rădăcini confundate” și valoarea celor 2 rădăcini reale, sub forma: “x1 = x2=” valoare.

4) Dacă $b^2 - 4ac < 0$ atunci funcția afișează mesajul “rădăcini complexe” și “partea reala =” valoarea părții reale, “partea imaginara =” valoarea părții imaginare.

Testarea bazata pe clase de echivalenta (12)

P1: $a \neq 0 \ \&\& \ b \neq 0 \ \&\& \ c \neq 0$

P2: $b^2 - 4ac > 0$

P3: $b^2 - 4ac = 0$

P4: $b^2 - 4ac < 0$

Clasele de echivalență ale domeniului de intrare:

CE1. P1&&P2

CE2. P1&&P3

CE3. P1&&P4

Cazuri de test – testarea minimala: un caz de test pentru fiecare clasa de echivalenta

CT	a	b	c	Rezultat asteptat
1	2	5	-3	Radacini reale: $x_1=0.5$; $x_2=-3$
2	5	20	32	Radacini complexe; partea reala=-2; partea imaginara= 1.54919
3	1	4	4	Radacini confundate; $x_1=x_2= -2$

Testarea bazata pe clase de echivalenta (13)

Cazuri de test – testarea minimala robusta:

Se adauga cazuri de test cu valori in afara domeniilor variabilelor a, b, c:

CT	a	b	c	Rezultat asteptat
1	0	10	-1	Valoare nevalida pentru a
2	2	0	3	Valoare nevalida pentru b
3	5	4	0	Valoare nevalida pentru c
4	0	0	5	Valoare nevalida pentru a si b
5	0	10	0	Valoare nevalida pentru a si c
6	1	0	0	Valoare nevalida pentru b si c

Testarea bazata pe clase de echivalenta (14)

Avantajele testarii bazate pe clase de echivalenta:

- Numarul de cazuri de test necesare pentru acoperirea unui domeniu larg de intrari este in general mic.
- **Este o metoda de testare sistematica.**
- Probabilitatea de descoperire a defectelor este mai mare decat in cazul alegerii aleatoare a unei suite de teste de aceeasi dimensiune.

Testele statistice (1)

- ❖ Se efectueaza în timpul testării de sistem
- ❖ Sunt, ca și testele funcționale, teste “black box”.
- ❖ **Sunt teste în care datele de intrare se aleg aleator**, după o lege de probabilitate care poate fi:
 - uniformă pe domeniul datelor de intrare al programului testat - aceste teste se mai numesc teste statistice uniforme;
 - similară cu distribuția datelor de intrare estimate pentru exploatarea programului – aceste teste se mai numesc teste statistice operaționale.
- **Rezultatele experimentale ale testelor statistice uniforme sunt inegale.** Pentru unele programe s-au dovedit foarte eficiente, ele conducând la descoperirea unor defecte importante. Pentru altele s-au dovedit ineficiente.

Testele statistice(2)

Testele statistice uniforme

- Asigură o bună acoperire a căilor de execuție ale programului dacă domeniile de intrare ale căilor de execuție au o probabilitate asemănătoare.
- Nu acoperă căile care corespund tratării excepțiilor. De aceea trebuie completate cu teste folosind date de intrare în afara domeniului intrărilor.

Testele statistice operaționale sunt în general teste de fiabilitate.

- Prin ele se urmărește nu numai descoperirea de defecte ci și comportarea programului în timp.
- Căderile observate în timpul acestor teste permit estimarea măsurilor de fiabilitate, cum ar fi MTBF (Mean Time Between Failures).

Lecturi suplimentare

1. <https://www.guru99.com/equivalence-partitioning-boundary-value-analysis.html>
2. <https://www.geeksforgeeks.org/equivalence-partitioning-method/>
3. <https://hackernoon.com/equivalence-class-partitioning-and-boundary-value-analysis-in-black-box-testing-znh3umt>