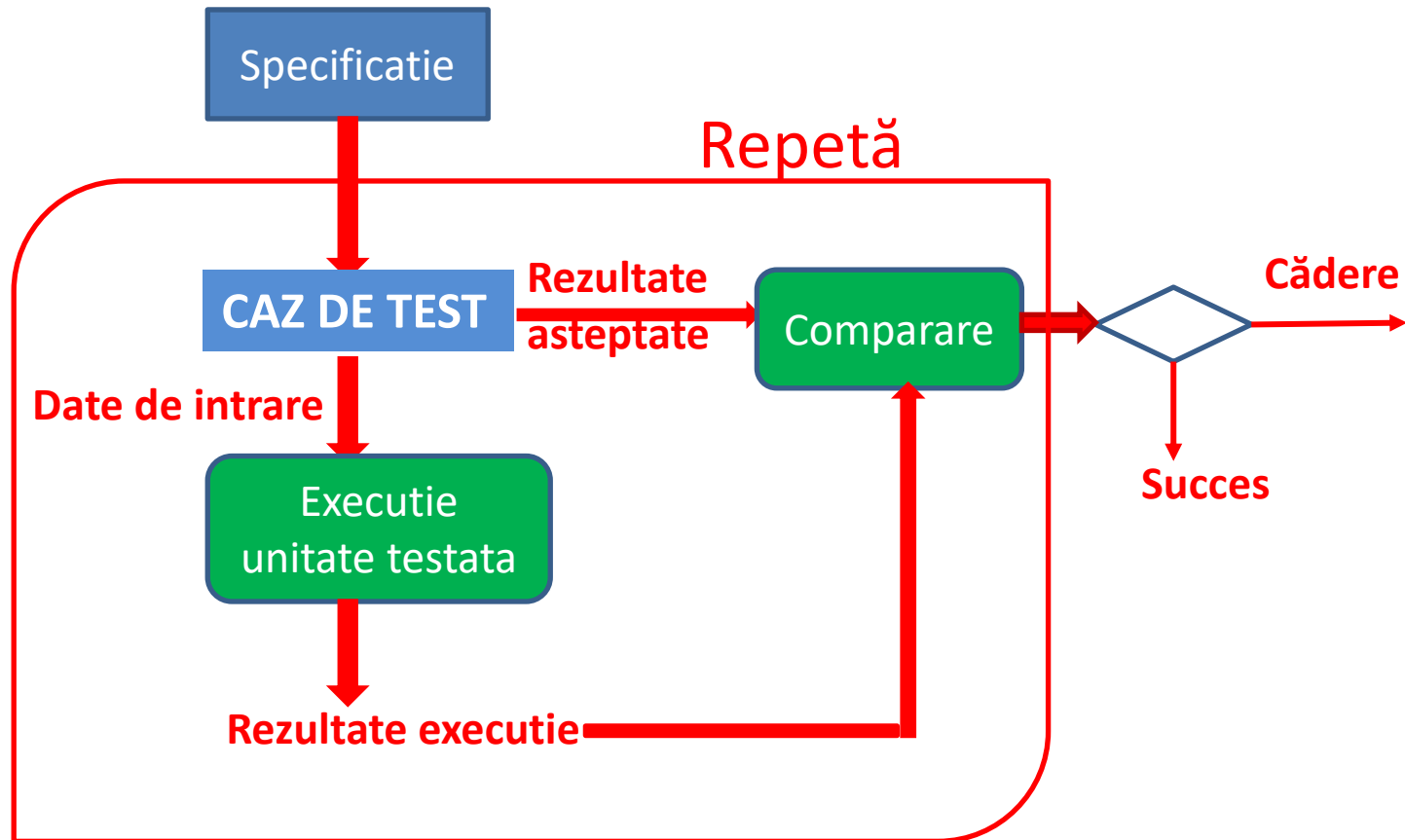


Alegerea cazurilor de test în testarea funcțională -1

Prof. univ. dr. ing. Florica Moldoveanu

Testare funcțională: testare black box



Date de intrare = vector de test:

- date prin care se verifica functionarea conform specificatiei
- date prin care se pot descoperi defecte de implementare

Alegerea cazurilor de test in testarea functionala

- Bazata pe specificatia unitatii testate
- Focalizata pe verificarea comportarii sale externe
- Unitatea testata este tratata ca o «cutie neagra»: **testare «black box»**
- Este utilizata la toate nivelurile de testare: unitara, de integrare, de sistem si de acceptare

Un caz de test cuprinde **un set de valori pentru variabilele de intrare ale unitatii testate** impreuna cu **rezultatele pe care unitatea testata trebuie sa le produca** (comportarea asteptata a programului).

- **Cazurile de test se aleg pentru a verifica functionarea unitatii testate conform specificatiei, dar și pentru a descoperi defecte în implementare.**

In testarea unei functii dintr-o clasa setul variabilelor de intrare/iesire include: parametrii functiei si variabilele membru ale clasei.

Setul de date de intrare folosit intr-un caz de test îl numim **vector de test.**

Tipuri de teste functionale

- **Testarea “ad hoc”:** programul este executat (in mod repetat) pentru intrari alese arbitrar observandu-se comportarea sa – este principala metoda de detectare a problemelor raportate de utilizatorii finali.
- **Testarea folosind “valori speciale” din domeniile variabilelor de intrare sau de iesire:**
 - Un vector de test contine o combinatie de **valori speciale** ale variabilelor de intrare
 - **Pairwise testing** – reduce numarul de combinatii intre valorile speciale
- **Testarea bazata pe clasele de echivalenta ale variabilelor de intrare/iesire:**

vectorii de test se aleg pe baza predicatelor definite asupra variabilelor de intrare sau iesire.
- **Teste statistice**
- **Testarea bazata pe specificatii formale:** specificatii de intrare-iesire (pre si post conditii), automate cu stari finite, s.a.

Testarea folosind “valori speciale” din domeniile variabilelor de intrare sau de iesire

Concepte cheie:

- Se identifica domeniul de valori al fiecărei variabile de intrare și de ieșire a unității testate, analizând specificația cerințelor și documentele de proiectare.
- Se selectează **valori speciale** (valori cu proprietăți *importante*) din domeniul fiecărei variabile de intrare sau de ieșire.
- Se proiectează cazuri de test considerând diferite combinații de **valori speciale din domeniile variabilelor de intrare**.
- Se proiectează cazuri de test continuând combinații de valori de intrare care produc **valori speciale din domeniile variabilelor de ieșire**.

Selectia valorilor *speciale* din domeniul unei variabile de intrare sau de iesire se bazeaza pe:

- Domeniul (tipul) variabilei
- Modul în care este utilizată: variabila de intrare, de ieșire, de intrare-ieșire

Alegerea valorilor de test (1)

❑ Criterii de selectie a valorilor speciale pentru variabile de intrare

1. **Domeniul este un set discret redus de valori** (ex. {3, -5, 10}, {True, False}, {P, Q, R}) → se efectueaza teste folosind fiecare valoare din set si valori din afara setului (intrari nevalide)

2. **Domeniul este compus din unul sau mai multe segmente continue de valori:**

Exemple: [0..1000]; [-1, 1] si [10, 100]; [val. minima- val. maxima a unui intreg pe 16 biti];

→ datele de test (**valorile speciale**) se selecteaza astfel:

- Valoarea minima a fiecarui segment
- Valoarea maxima a fiecarui segment
- O valoare reprezentativa din interiorul fiecarui segment
- Valori cu proprietati matematice speciale; ex: 0, 1, cea mai mica/ mare valoare reala
- Valori din afara fiecarui segment (intrari nevalide)

3. **Variabila de tip String** – se fac teste cu: şir vid, şir cu un singur caracter, şir cu numarul maxim de caractere, şir cu un numar intermediar de caractere, şir cu numar de caractere mai mare decat cel admis. Şirurile de test vor contine caractere speciale.

Alegerea valorilor de test (2)

❑ Criterii de selectie a valorilor speciale pentru variabile de iesire

1. Domeniul este un set discret de valori

→ **set redus**: se efectueaza teste care conduc la obtinerea fiecărei valori din set;

→ **set mare**: se executa un numar de teste care conduc la obtinerea unui numar mare de valori din domeniul de iesire

2. Domeniul este compus din unul sau mai multe segmente continue de valori → cazurile de test se proiecteaza selectand:

- Valori de intrare care produc valorile minime ale segmentelor
- Valori de intrare care produc valorile maxime ale segmentelor
- Valori de intrare care produc cateva date din interiorul fiecarui segment

❑ Criterii de selectie pentru variabile de intrare- iesire (variabile de intrare in care se regasesc iesiri): teste suplimentare

- Un caz de test care produce o valoare de iesire a variabilei diferita de valoarea sa de intrare
- Un caz de test in care valoarea de iesire a variabilei este identica cu valoarea sa de intrare

Alegerea valorilor de test (3)

Variabilă de intrare tablou

- Un tablou poate avea una sau mai multe dimensiuni. Toate elementele unui tablou sunt de același tip.
- Cazurile de test trebuie să includă teste cu diferite dimensiuni ale tabloului (dacă dimensiunea este variabilă de intrare), în diferite combinații dacă este multi-dimensional: tablou vid, tablou cu dimensiunea minimă, tablou cu dimensiunea maximă, tablou cu o dimensiune intermediară, tablou cu prima dimensiune minimă și a 2-a dimensiune maximă, ș.a.
- Cazurile de test trebuie să includă tablouri cu valori speciale, alese în funcție de tipul elementelor sale.
- Aceste criterii generale pot fi combinate cu unele euristici, ținând cont de felul în care este folosit tabloul.
- În unele cazuri, sunt necesare teste pe substructuri ale unui tablou multi-dimensional (pt. un tablou 2D: elementele diagonalei, matricea triunghiulară inferioară și cea superioară, rânduri, coloane).

Alegerea valorilor de test (4)

Variabila de intrare tablou – exemplu.

Fie o functie C de cautare a unui numar intreg intr-un tablou de numere intregi, specificata astfel

int cauta (int X[], int n, int y);

Functia cauta numarul y in tabloul X de n elemente; daca este gasit intoarce indexul primei sale aparitii in tablou altfel intoarce -1.

Setul minim de cazuri de test trebuie sa verifice functia in urmatoarele situatii:

- 1) tablou vid ($n=0$) ; functia trebuie sa intoarca -1;
- 2) tablou cu un singur element ($n=1$): cel mai mic intreg, cel mai mare intreg, un numar intreg oarecare
- 3) tablou cu numar maxim de elemente (daca este specificat): valori diverse
- 4) tablou cu un numar intermediar de elemente: valori diverse

Alegerea valorilor de test (5)

Pentru cazurile (2), (3) si (4) se vor face teste in care:

- numarul y exista in tablou; functia intoarce indexul sau in tablou;
- numarul y nu exista in tablou; functia intoarce -1;

Dupa fiecare apel al functiei se va afisa tabloul X pentru a se verifica faptul ca nu a fost modificat.

➤ Setul de cazuri de test se imbogateste cu teste alese pe **criterii euristice**

Euristici de testare pentru programe de cautare:

- programele de cautare produc erori atunci cand elementul cautat este primul sau ultimul in lista;
- de multe ori programatorii neglijeaza situatiile in care colectia prelucrata in program are un numar de elemente neobisnuit, de exemplu zero sau unu;
- uneori, programele de cautare se comporta diferit in cazurile: numar de elemente din colectie par, respectiv numar de elemente din colectie impar; de aceea trebuie testate ambele situatii

Alegerea valorilor de test (6)

Tinand cont de aceste euristici, cazurile de test trebuie sa verifice functia in urmatoarele situatii:

- 1) tablou vid ($n=0$)
- 2) tablou cu un singur element ($n=1$)
 - a) valoarea parametrului y exista in tablou
 - b) valoarea parametrului y nu exista in tablou
- 3) tablou cu numar par de elemente (n este un numar par)
 - a) Valoarea lui y este prima in tablou (indexul 0)
 - b) Valoarea lui y este ultima in tablou (indexul $(n-1)$)
 - c) Valoarea lui y este intr-o pozitie oarecare a tabloului
 - d) Valoarea lui y nu exista in tablou
- 4) tablou cu numar impar de elemente (n este un numar impar)
 - a,b,c,d ca la punctul 3

In teste se vor folosi numere intregi cu valori extreme si valori normale.

Alegerea valorilor de test (7)

Exemplu de suite de test:

- 1) intrari: $n=0$; orice tablou; y orice numar
rezultate : valoarea functiei = -1, tabloul nemodificat
- 2) intrari: $n=1$; $x[0] = 10$; $y = 10$
rezultate : valoarea functiei = 0, tabloul nemodificat: $x[0] = 10$
- 3) intrari: $n=1$; $x[0] = 10$; $y = 15$
rezultate : valoarea functiei = -1, tabloul nemodificat: $x[0] = 10$
- 4) intrari: $n=2$; $x[0] = 10$, $x[1] = 20$; $y = 10$
rezultate: valoarea functiei = 0, tabloul nemodificat: $x[0] = 10$, $x[1] = 20$
- 5) intrari: $n=2$; $x[0] = 10$, $x[1] = 20$; $y = 20$
rezultate: valoarea functiei = 1, tabloul nemodificat: $x[0] = 10$, $x[1] = 20$

Alegerea valorilor de test (8)

6) intrari: $n=2$; $x[0] = 10$, $x[1] = 20$; $y = -1$

rezultate: valoarea functiei = - 1, tabloul nemodificat: $x[0] = 10$, $x[1] = 20$

7) intrari: $n=4$; $x[0] = -32677$, $x[1] = 0$; $x[2] = 32677$, $x[3] = -1$, $y = -1$

rezultate: valoarea functiei = 3, tabloul nemodificat

8) intrari: $n=6$; $x[0] = 32677$, $x[1] = -70$; $x[2] = 32677$, $x[3] = -1$, $x[4] = 0$; $x[5] = 100$, $y = 32677$

rezultate: valoarea functiei = 0, tabloul nemodificat

In mod similar se vor acoperi cazurile: n impar mare, cu valori diverse si elementul cautat in prima, ultima si o pozitie intermediara.

Alegerea valorilor de test (9)

Concluzii:

Alegerea cazurilor de test prin analiza domeniilor variabilelor de intrare:

- Conduce la prea multe cazuri de test: **vectorii de test** sunt combinatii ale valorilor selectate ale variabilelor de intrare (numarul de CT = nr de combinatii alese).
- Generarea iesirii asteptate pentru un vector de test este relativ simpla: proiectantul testului calculeaza iesirea asteptata analizand specificatia programului.

Alegerea cazurilor de test prin analiza domeniilor variabilelor de iesire:

- Numarul de cazuri de test obtinute este mai mic – nu este necesar sa se considere combinatii intre valorile speciale ale variabilelor de iesire
- Generarea unui vector de intrare necesar pentru a produce o valoare de iesire/combinatie de valori de iesire necesita analiza specificatiei in directia inversa.

Numarul de cazuri de test determinate prin analiza domeniilor variabilelor de intrare si de iesire este prea mare → **poate fi redus prin diferite metode.**

Pairwise testing(1)

Metoda produce un numar de CT mult mai mic decat in cazul folosirii tuturor combinatiilor intre valorile selectate.

Asigura că fiecare combinatie posibila intre valorile selectate pentru fiecare pereche de variabile de intrare este acoperita prin cel putin un test.

Studii empirice au aratat ca metoda poate conduce la detectarea a circa 70% din defecte folosind cazurile de test generate

Exemplu:

Fie un sistem cu 3 intrari, X, Y, Z, pentru care s-au selectat valorile:

$X = \{\text{True}, \text{False}\}$, $Y = \{1, 100\}$, $Z = \{Q, R\}$

Numarul total de combinatii (vectori de intrare) este: $2 \times 2 \times 2 = 8$. Numarul de CT generate prin

Pairwise Testing este 4.

Caz de test	Intrare X	Intrare Y	Intrare Z
CT1	True	1	Q
CT2	True	100	R
CT3	False	100	Q
CT4	False	1	R

Pairwise testing(2) – tablouri ortogonale

Notatie: $L_{\text{Runs}}(\text{Levels}^{\text{Factors}})$

Factors: numarul de variabile de intrare;

Levels: nr maxim de valori distincte ale fiecarei variabile

Runs: numarul de cazuri de test

$L_4(2^3)$ Variabile de intrare

Runs	1	2	3
1	1	1	1
2	1	2	2
3	2	1	2
4	2	2	1

$X = \{\text{true}, \text{false}\},$

$Y = \{1, 100\},$

$Z = \{Q, R\}$

Maparea valorilor
variabilelor pe niveluri

Cazuri de test	X	Y	Z
1	true	1	Q
2	true	100	R
3	false	1	R
4	false	100	Q

➤ 4 cazuri de test in loc de $2 \times 2 \times 2 = 8!$

Tabloul ortogonal pentru 3 variabile,
fiecare cu 2 valori posibile

- Valorile distincte ale variabilelor de intrare se mapeaza pe niveluri: [1-levels].

Pairwise testing(3) – tablouri ortogonale(cont)

$L_9(3^4)$

Factors

Runs	1	2	3	4
1	1	1	1	1
2	1	2	2	2
3	1	3	3	3
4	2	1	2	3
5	2	2	3	1
6	2	3	1	2
7	3	1	3	2
8	3	2	1	3
9	3	3	2	1

A = { -1, 100, 200 }

B = { a, b, c }

C = { -0.1, 0.1, 0 }

D = { true, false }

Maparea valorilor
variabilelor pe niveluri

Cazuri de test	A	B	C	D
1	-1	a	-0.1	true
2	-1	b	0.1	false
3	-1	c	0	true
4	100	a	0.1	false
5	100	b	0	true
6	100	c	-0.1	false
7	200	a	0	false
8	200	b	-0.1	true
9	200	c	0.1	true

Tabloul ortogonal pentru 4 variabile,
fiecare cu cel mult 3 valori posibile

➤ 9 cazuri de test in loc de $3 \times 3 \times 3 \times 3 = 81$

- Daca o variabila are un numar de valori distincte < levels, valorile sale vor fi mapate pe mai multe niveluri.

Pairwise testing(4)

Tablourile se numesc “ortogonale” (termen imprumutat din matematica) deoarece:

- fiecare combinatie posibila intre valorile a doua variabile apare o singura data;
- fiecare vector de test (linie a tabloului) este diferit de ceilalti – furnizeaza o informatie unica;
- fiecare vector de test este independent statistic de ceilalti (aparitia unuia nu influenteaza aparitia celorlalti) – corelatia dintre ei este nula.

Metoda “pairwise testing” este eficienta:

- Cazurile de test se obtin simplu:
 - tablourile ortogonale se genereaza automat
 - maparea valorilor variabilelor pe niveluri se poate face automat
- Este o metoda statistica sistematica de testare a interactiunilor intre perechi de variabile
- Furnizeaza o acoperire uniforma reprezentativa a tuturor combinatiilor intre perechi de variabile
- S-a dovedit foarte utila in testele de integrare, testele de interfata utilizator, testarea de sistem.

Lecturi suplimentare

1. <https://www.softwaretestinghelp.com/what-is-pairwise-testing/>
2. <http://www.testingeducation.org/wtst5/PairwisePNSQC2004.pdf>
3. <https://blog.issart.com/real-life-examples-of-how-to-use-pairwise-technique-in-test-design/>