



# Testiranje softvera

ETF BEOGRAD, 2019/2020.

VEŽBE #4, ASISTENT: DR DRAŽEN DRAŠKOVIĆ



# Kombinatorno testiranje

*COMBINATORIAL SOFTWARE TESTING TECHNIQUES*

# Kombinatorno testiranje

- ▶ Testirati kombinacije vrednosti, ali nije praktično testirati sve moguće kombinacije zbog njihovog ukupnog broja
- ▶ Uzima se podskup kombinacija koji zadovoljava neku od definisanih kombinatornih strategija testiranja
- ▶ Kombinatorne strategije testiranja su:
  - ▶ Pokrivanje svih kombinacija
  - ▶ Pokrivanje svih pojedinih vrednosti
  - ▶ Pokrivanje parova vrednosti
  - ▶ Pokrivanje n-torki vrednosti

# Zadatak 1 - Metod testiranja svih parova

- ▶ Dobra startna tačka za diskusiju o metodi testiranja svih parova (eng. *All-pairs, Pairwise*) je Dekartov proizvod. Dekartov proizvod je scenario u kome je svaka jedinica iz grupe povezana sa svakom jedinicom iz druge grupe, tako da su sve kombinacije jedinica kreirane uključujući sve grupe.
- ▶ Na primer, zamislimo sledeću jednostavnu veb aplikaciju sa korisničkim ekranom, na kome se nalaze dve padajuće liste i dugme OK. Lista 1 sadrži tri vrednosti: 0, 1 i 2. Lista 2 sadrži dve vrednosti: 10 i 20. Korisnik treba da izabere jednu od vrednosti iz svake liste, a nakon pritiska na dugme OK, proizvod dve unete vrednosti se prikazuje na ekranu.
- ▶ Promenljive i njihove vrednosti izgledaju kao u tabeli.
- ▶ Kako biste testirali ovaj program?  
Koliko različitih ulaznih kombinacija postoji?  
Koliko vremena bi bilo potrebno za testiranje svih mogućih kombinacija?

**All-Pairs primer 1**

Lista 1:  Lista 2:

Rezultat:

Lista 1	Lista 2
0	10
1	20
2	

# Zadatak 1 - Metod testiranja svih parova - rešenje (1)

- ▶ Postoji  $3 \times 2 = 6$  mogućih kombinacija vrednosti obe liste. Šest rezultujućih kombinacija je rezultat Dekatovog proizvoda dva ulaza i izgledao bi ovako:

**All-Pairs primer 1**

Lista 1:  Lista 2:

Rezultat:

Lista 1	Lista 2
0	10
0	20
1	10
1	20
2	10
2	20

# Zadatak 1 - Metod testiranja svih parova (2)

- ▶ b) Verzija 2.0 našeg programa ima nove karakteristike:

Lista 1 sada sadrži celobrojne (*int*) vrednosti od 0 do 9 i Lista 2 je promenjena u tekstualno polje, dopuštajući unos celobrojnih (*int*) vrednosti izmedju 1 i 99. Dodali smo i dva polja za potvrdu (*checkbox*). Kada je čekirano prvo polje, dobijeni proizvod se množi sa -1 (vrši negaciju). Kada je čekirano drugo polje, vrši se množenje dobijenog proizvoda sa samim sobom (generiše se kvadrat dobijenog proizvoda).

Lista	Textbox	Checkbox1 (-x)	Checkbox2 (x*x)
0	1	On	On
1	2	Off	Off
2	3		
3	4		
4	...		
5	96		
6	97		
7	98		
8	99		
9			

# Zadatak 1 - Metod testiranja svih parova - rešenje (2)

- ▶ U ovom slučaju imamo  $10 \times 99 \times 2 \times 2 = 3\,960$  mogućih korektnih ulaznih vrednosti. Takođe postoji i određeni broj nekorektnih ulaza (obratite pažnju da je tekstualno polje uvelo novi koncept - mogućnost nevalidnog ulaza, o čemu će biti reči kasnije).
- ▶ Kako rešiti probleme?
- ▶ Prvo: identifikujemo parametre (promenljive) i vrednosti koje svaka promenljiva može uzeti
- ▶ Drugo: pojednostaviti vrednosti. Neophodno je grupisati vrednosti u nekoliko kategorija (grupa) i uvesti pojam graničnih vrednosti. Na primer, za slučaj tekstualnog polja, umesto listanja svih validnih vrednosti od 1 do 99, koristiti reprezentativnu vrednost.
- ▶ U našem primeru, korisnik ima više slobode za unos nevalidne vrednosti u tekstualno polje nego što je to slučaj sa padajućom listom. Primetiti da je vrednosti moguće grupisati čak i ukoliko nisu diskretne. Na primer, vrednost kategorisane za proizvoljni tekst mogu biti: svi alfabetski karakteri, svi numerički karakteri, velika slova, mala slova, reči pod navodnicima, itd.

# Zadatak 1 - Metod testiranja svih parova - rešenje (3)

- ▶ Grupišimo 99 korektnih vrednosti, plus beskonačan broj nekorektnih vrednosti u tri grupe:
  - ▶ bilo koja validna celobrojna vrednost,
  - ▶ bilo koja nevalidna celobrojna vrednost,
  - ▶ bilo koji alfabetski karakter (koji bi bio nevalidan).
- ▶ Grupišimo 10 vrednosti u padajućoj listi na dve:
  - ▶ 0 i bilo koja druga vrednost.

Lista	Textbox	Checkbox1 (-x)	Checkbox2 (x*x)
0	Validni ceo broj	On	On
<b>Bilo koja vrednost</b>	Nevalidni ceo broj	Off	Off
	Alfabetski karakter		

Textbox (3)	Lista (2)	Negativni proizvod (2)	Kvadrat proizvoda (2)



# Zadatak 1 - Metod testiranja svih parova - rešenje (4)

Textbox (3)	Lista (2)	Negativni proizvod (2)	Kvadrat proizvoda (2)
<b>Validni ceo broj</b>			
<b>Validni ceo broj</b>			
<b>Nevalidni ceo broj</b>			
<b>Nevalidni ceo broj</b>			
<b>Alfabetски karakter</b>			
<b>Alfabetски karakter</b>			

# Zadatak 1 - Metod testiranja svih parova - rešenje (5)

Textbox (3)	Lista (2)	Negativni proizvod (2)	Kvadrat proizvoda (2)
Validni ceo broj	0		
Validni ceo broj	<b>Bilo koja druga</b>		
Nevalidni ceo broj	0		
Nevalidni ceo broj	<b>Bilo koja druga</b>		
Alfabetски karakter	0		
Alfabetски karakter	<b>Bilo koja druga</b>		

# Zadatak 1 - Metod testiranja svih parova - rešenje (6)

Textbox (3)	Lista (2)	Negativni proizvod (2)	Kvadrat proizvoda (2)
Validni ceo broj	0	On	
Validni ceo broj	Bilo koja druga	Off	
Nevalidni ceo broj	0	On	
Nevalidni ceo broj	Bilo koja druga	Off	
Alfabetски karakter	0	On	
Alfabetски karakter	Bilo koja druga	Off	

Da li imamo sve kombinacije?

# Zadatak 1 - Metod testiranja svih parova - rešenje (7)

Textbox (3)	Lista (2)	Negativni proizvod (2)	Kvadrat proizvoda (2)
Validni ceo broj	0	On	
Validni ceo broj	Bilo koja druga	Off	
Nevalidni ceo broj	0	Off	
Nevalidni ceo broj	Bilo koja druga	On	
Alfabetски karakter	0	On	
Alfabetски karakter	Bilo koja druga	Off	

Da li imamo sada sve kombinacije?

Textbox + Lista, Textbox + Negativan proizvod, Lista + Negativan proizvod

# Zadatak 1 - Metod testiranja svih parova - rešenje (8)

Textbox (3)	Lista (2)	Negativni proizvod (2)	Kvadrat proizvoda (2)
Validni ceo broj	0	On	Čekiran
Validni ceo broj	Bilo koja druga	Off	Nije čekiran
Nevalidni ceo broj	0	Off	Čekiran
Nevalidni ceo broj	Bilo koja druga	On	Nije čekiran
Alfabetски karakter	0	On	Nije čekiran
Alfabetски karakter	Bilo koja druga	Off	Čekiran

Da li imamo sve kombinacije?

Da smo testirali sve kombinacije vrednosti imali bismo  $3 \times 2 \times 2 \times 2 = 24$  kombinacije

# Zadatak 1 - Metod testiranja svih parova (3)

- ▶ c) Verzija 3.0 našeg programa doda još dva polja za selektovanje. Polje za selektovanje 3 će dati faktorijel za dobijeni proizvod, a polje za selektovanje 4 će konvertovati izlaz u heksadecimalnu notaciju. Na osnovu izmena potrebno je da dodamo još dve kolone u naše tabele i unesemo njihove vrednosti.

Textbox (3)	Lista (2)	Negativni proizvod (2)	Kvadrat proizvoda (2)	Factorial (2)	Hex (2)
<b>Validni ceo broj</b>	0	On	Čekiran	Da	
<b>Validni ceo broj</b>	Bilo koja druga	Off	Nije čekiran	Ne	
<b>Nevalidni ceo broj</b>	0	Off	Čekiran	Ne	
<b>Nevalidni ceo broj</b>	Bilo koja druga	On	Nije čekiran	Da	
<b>Alfabetски karakter</b>	0	On	Nije čekiran	Ne	
<b>Alfabetски karakter</b>	Bilo koja druga	Off	Čekiran	Da	

# Zadatak 1 - Metod testiranja svih parova - rešenje (9)

Textbox (3)	Lista (2)	Negativni proizvod (2)	Kvadrat proizvoda (2)	Factorial (2)	Hex (2)
<b>Validni ceo broj</b>	0	On	Čekiran	Da	Dec
<b>Validni ceo broj</b>	Bilo koja druga	Off	Nije čekiran	Ne	Hex
<b>Validni ceo broj</b>	Bilo koja druga	On	Nije čekiran	Ne	Hex
<b>Nevalidni ceo broj</b>	0	Off	Čekiran	Ne	Hex
<b>Nevalidni ceo broj</b>	Bilo koja druga	On	Nije čekiran	Da	Dec
<b>Nevalidni ceo broj</b>	Bilo koja druga	Off	Nije čekiran	Da	Dec
<b>Alfabetски karakter</b>	0	On	Nije čekiran	Ne	Dec
<b>Alfabetски karakter</b>	Bilo koja druga	Off	Čekiran	Da	Hex

Dolazimo do svih parova sa osam test primera,  
umesto svih  $3 \times 2 \times 2 \times 2 \times 2 \times 2 = 96$  kombinacija.

## Zadatak 2 - Testiranje IPO procedurom

- ▶ Razmotrimo program sa tri faktora A, B i C. Faktor A može uzeti vrednosti iz skupa  $\{a_1, a_2, a_3\}$ , faktor B iz skupa  $\{b_1, b_2\}$  i faktor C iz skupa  $\{c_1, c_2, c_3\}$ . Potrebno je izgenerisati pokrivajući niz sa mešovitim nivoima, za tri navedena faktora, primenom *In-parameter-order* (IPO) procedure.
- ▶ **Analiza problema:** procedura *In-parameter-order* (IPO), koja služi za generisanje pokrivajućih nizova sa mešovitim nivoima.
- ▶ Ulaz:
  - ▶ (a)  $n \geq 2$ : Broj parametara (faktora);
  - ▶ (b) Broj vrednosti (nivoa) za svaki faktor.
- ▶ Izlaz: MCA (pokrivajući niz sa mešovitim nivoima).
- ▶ IPO procedura sastoji se iz tri koraka:
  - ▶ Korak 1: Glavna procedura (GP)
  - ▶ Korak 2: Horizontalni rast (HR)
  - ▶ Korak 3: Vertikalni rast (VR)



## Zadatak 2 - Testiranje IPO procedurom - Rešenje (1)

- ▶ Glavna procedura
- ▶ Konstruišemo sve kombinacije parova vrednosti za prva dva faktora A i B. Dobijamo sledeći skup parova:  
 $T = \{ (a_1, b_1), (a_1, b_2), (a_2, b_1), (a_2, b_2), (a_3, b_1), (a_3, b_2) \}$
- ▶ Elemente skupa  $T$  označićemo sa:  $t_1, t_2, \dots, t_6$ .
- ▶ Kompletna IPO procedura bi se završila u ovom trenutku, ukoliko bi broj parametara bio 2 ( $n=2$ ). U našem primeru  $n=3$ , iz tog razloga potrebno je da nastavimo algoritam, a korak koji sledi je horizontalni rast (HR).

## Zadatak 2 - Testiranje IPO procedurom - Rešenje (2)

▶ Horizontalni rast:

Kreirati skup svih aktivnih parova (AP), između parametara A i C, a zatim između B i C.

▶ Ovo nas dovodi do narednog skupa od 15 parova:

$$AP = \{ (a_1, c_1), (a_1, c_2), (a_1, c_3), (a_2, c_1), (a_2, c_2), (a_2, c_3), (a_3, c_1), (a_3, c_2), (a_3, c_3), \\ (b_1, c_1), (b_1, c_2), (b_1, c_3), (b_2, c_1), (b_2, c_2), (b_2, c_3) \}$$

▶ AP je skup parova koji još uvek nisu pokriveni.

▶ Označimo sa TP skup kombinacija dobijen proširenjem kombinacija iz skupa parova T.

▶ U ovom trenutku TP je prazan, iz razloga što nismo do sada izvršili nikakva proširenja skupa T.

## Zadatak 2 - Testiranje IPO procedurom - Rešenje (3)

- ▶ Proširiti  $t_1, t_2, t_3$  dodavanjem  $c_1, c_2, c_3$  vrednosti. Ovim proširenjem dobijamo:  
 $t_1' = (a_1, b_1, c_1), t_2' = (a_1, b_2, c_2),$  i  $t_3' = (a_2, b_1, c_3)$
- ▶ Vršimo ažuriranje do tada praznog skupa TP, koji sada postaje:  
 $TP = \{a_1, b_1, c_1), (a_1, b_2, c_2), (a_2, b_1, c_3)\}$
- ▶ Ažuriramo skup aktivnih parova iz AP koje je potrebno da pokrijemo, time što uklanjamo 3 kombinacije  $(a_1, c_1), (a_1, c_2)$  i  $(a_2, c_3)$  uključene u skup test primera  $T'$ :  
 $AP = \{ (a_1, c_3), (a_2, c_1), (a_2, c_2), (a_3, c_1), (a_3, c_2), (a_3, c_3), (b_1, c_2), (b_2, c_1), (b_2, c_3) \}$
- ▶ Još uvek nismo izvršili proširenja  $t_4, t_5, t_6$ , jer C ne poseduje dovoljan broj elemenata. Najbolji način za proširenje ovih parova pronaći ćemo u narednom koraku.

## Zadatak 2 - Testiranje IPO procedurom - Rešenje (4)

- ▶ Proširenje  $t_4, t_5, t_6$  pogodnim odabiranjem vrednosti C.
- ▶ Ukoliko izvršimo proširenje  $t_4=(a_2, b_2)$  sa  $c_1$ , izvršićemo pokrivanje dva ne pokrivena para iz AP skupa i to  $(a_2, c_1)$  i  $(b_2, c_1)$ . Ukoliko proširenje izvršimo sa vrednošću  $c_2$ , pokrивamo samo jedan par iz skupa AP. Proširenjem sa  $c_3$ , takođe pokrивamo samo jedan par iz AP skupa. Zato za proširenje test primera  $t_4$  iskoristićemo vrednost  $c_1$  i time eliminisati (pokriti) još 2 para iz AP. Sada ažurni skupovi izgledaju ovako:

$$TP = \{ (a_1, b_1, c_1), (a_1, b_2, c_2), (a_2, b_1, c_3), (a_2, b_2, c_1) \}$$

$$AP = \{ (a_1, c_3), (a_2, c_2), (a_3, c_1), (a_3, c_2), (a_3, c_3), (b_1, c_2), (b_2, c_3) \}$$

- ▶ Slično, vršimo proširenje za  $t_5$  i  $t_6$ , sa najboljim mogućim izborom vrednosti parametra C. Imamo dve mogućnosti izbora:

$$t_5' = (a_3, b_1, c_3) \text{ i } t_6' = (a_3, b_2, c_1)$$

$$t_5'' = (a_3, b_1, c_2) \text{ i } t_6'' = (a_3, b_2, c_3)$$

## Zadatak 2 - Testiranje IPO procedurom - Rešenje (5)

- ▶ Odabraćemo prvi slučaj izborom  $t_5'$  i  $t_6'$ . U tom slučaju uklanjamo dve kombinacije iz AP i ostajemo na 5, pa skupovi izgledaju ovako:

$$TP = \{ (a_1, b_1, c_1), (a_1, b_2, c_2), (a_2, b_1, c_3), (a_2, b_2, c_1), (a_3, b_1, c_3), (a_3, b_2, c_1) \}$$

$$AP = \{ (a_1, c_3), (a_2, c_2), (a_3, c_2), (b_1, c_2), (b_2, c_3) \}$$

- ▶ Prethodnim korakom, završili smo horizontalni rast. Ono što treba primetiti jeste da je ostalo pet parova iz AP, koji još uvek nisu pokriveni.
- ▶ Pored toga izgenerisali smo šest kompletnih kombinacija u skupu TP.
- ▶ Sada prelazimo na treći korak, odnosno vertikalni rast.

## Zadatak 2 - Testiranje IPO procedurom - Rešenje (6)

### ▶ VERTIKALNI RAST

- ▶ Za svaki par  $P_n$ , iz skupa AP, koji nedostaje, izgenerisaćemo novu kombinaciju u skupu TP, tako da P bude pokriveno. Počnimo sa parom  $P_{n_1} = (a_1, c_3)$ .
- ▶ Kombinacija  $t_7 = (a_1, *, c_3)$  pokriva par P. Primetiti da vrednost za faktor b nije bitna, znak \* upravo predstavlja proizvoljnu vrednost.
- ▶ Sledeće, razmotrimo  $P_{n_2} = (a_2, c_2)$ . Ovaj traženi par pokriven je kombinacijom  $t_8 = (a_2, *, c_2)$
- ▶ Sledeće, razmotrimo  $P_{n_3} = (a_3, c_2)$ . Ovaj traženi par pokriven je kombinacijom  $t_9 = (a_3, *, c_2)$
- ▶ Nadalje, razmotrimo  $P_{n_4} = (b_1, c_2)$ . Od prethodnog koraka imamo kombinaciju  $t_9 = (a_3, *, c_2)$ , koju možemo modifikovati sa  $*=b_1$ , da bi dobili kombinaciju  $t_9 = (a_3, b_1, c_2)$ . Ovom izmenom, pokrili smo traženi par  $P_{n_4}$ , bez dodavanja novih kombinacija u skupu test primera.
- ▶ Konačno, razmotrimo  $P_{n_5} = (b_2, c_3)$ . Već imamo kombinaciju  $t_7 = (a_1, *, c_3)$ , pa možemo izvršiti modifikovanje sa  $*=b_2$  čime dobijamo kombinaciju  $t_7 = (a_1, b_2, c_3)$ . Ovom izmenom, pokrili smo traženi par  $P_{n_5}$ , bez dodavanja novih kombinacija u skupu test primera.

## Zadatak 2 - Testiranje IPO procedurom - Rešenje (7)

Test primer	F (A)	F (B)	F (C)	Obuhvaćene kombinacije iz AP
TP-1	$a_1$	$b_1$	$c_1$	
TP-2	$a_1$	$b_2$	$c_2$	
TP-3	$a_2$	$b_1$	$c_3$	
TP-4	$a_2$	$b_2$	$c_1$	
TP-5	$a_3$	$b_1$	$c_3$	
TP-6	$a_3$	$b_2$	$c_1$	
	$a_1$	$b_2$	$c_3$	pokriva $Pn_1$ i $Pn_5$
	$a_2$	*	$c_2$	pokriva $Pn_2$
	$a_3$	$b_1$	$c_2$	pokriva $Pn_3$ i $Pn_4$

- ▶  $TP = \{ (a_1, b_1, c_1), (a_1, b_2, c_2), (a_1, b_2, c_3), (a_2, b_1, c_3), (a_2, b_2, c_1), (a_2, b_1^{\#}, c_2), (a_3, b_1, c_3), (a_3, b_2, c_1), (a_3, b_1, c_2) \}$

# moglo je umesto  $b_1$  da bude i  $b_2$

## Zadatak 2 - Testiranje IPO procedurom - Rešenje (8)

Test primer	F (A)	F (B)	F (C)
TP-1	$a_1$	$b_1$	$c_1$
TP-2	$a_1$	$b_2$	$c_2$
TP-3	$a_1$	$b_2$	$c_3$
TP-4	$a_2$	$b_1$	$c_2$
TP-5	$a_2$	$b_1$	$c_3$
TP-6	$a_2$	$b_2$	$c_1$
TP-7	$a_3$	$b_1$	$c_2$
TP-8	$a_3$	$b_1$	$c_3$
TP-9	$a_3$	$b_2$	$c_1$