

Vērtību apgabalu testēšana.
Robežvērtību testēšana.
Vērtību apgabalu un robežvērtību testēšanas
sarežģītība.

Ronalds Rundāns
2024

Izmantotā literatūra

- Vineta Arnicāne. Promocijas darbs „Sarežģītību ietekmēta programmatūras testēšana”, 2013.

„Sarežģītību ietekmēta programmatūras testēšana”

- vērtību apgabalu testēšanas modelis
- novērtēta testēšanas sarežģītība kā izpildāmo testu skaits dažādiem ekvivalences klašu un robežvērtību testēšanas jeb domēntestēšanas kritērijiem.
- tehnoloģiskā sarežģītība, kas praktiskos lietojumos padara neiespējamu šo testēšanas modeļu lietošanu.

Prezentācijas mērķi

- Aplūkot, kas ir programmas vērtību apgabals.
- Kādas robežvērtības izmanto ar dažādām robežvērtību testēšanas metodēm.
- Salīdzināt vērtību apgabalu un robežvērtību testēšanas sarežģītības.

Kas ir testēšana?

- Programmatūras testēšana ir process, ko veic cilvēku, datortehnikas un resursu kopums.
- Mērķis : realizēt ieinteresēto pušu dotos uzdevumus dotā laika un budžeta ietvaros.

Kas ir domēns?

Kas ir domēns?

- Katram programmas parametra pieļaujamo vērtību kopa.
- Piemēram, mēneši : Jan, Feb, ... , Dec

Kas ir vērtību apgabalu testēšana?

Kas ir vērtību apgabalu testēšana?

- Pamatpieeja ir sadalīt visu testējamā objekta ievaddomēnu apakšdomēnos jeb ekvivalences klasēs.
- Ievadi no viena apakšdomēna tiek apstrādāti vienādā veidā -> izraisa programmai viena veida izvadu vai uzvedību.

Kā atrodam ekvivalences klases?

Programmas ievaddomēna ekvivalences klases var tikt iegūtas ar programmas :

- strukturālo analīzi – ceļu analīzi
- funkcionālo analīzi – sadalīšanu ekvivalences klasēs,
- domēnanalīzi,
- robežvērtību analīzi.

Sadalīšana ekvivalences klasēs

- Pirmais solis ir katram programmas parametram noteikt pieļaujamo vērtību kopu jeb domēnu - pieļaujamo vērtību ekvivalences klases katram parametram.
- Otrais solis ir noteikt katra parametra ***NE**pieļaujamo vērtību ekvivalences klasi.

*Nepieļaujamā vērtība ir tāda, ko ir iespējams „iedot” programmas parametram, bet kas nav iekļauta pieļaujamo vērtību klasē.

Robežvērtību analīze

- Uzmanību pievērš ekvivalences klašu robežvērtībām, jo prakse rāda, ka robežvērtības bieži parāda kļūdas programmā.
- Robežvērtību testēšanas metodes testpiemēros izvēlas robežvērtības, izņēmumu jeb speciālās, īpašās vērtības, kā arī vērtības, kuras ir tuvu iepriekšminētajām vērtībām.

Robežvērtību testēšanas jēga

- Pārbaudīt situācijas, kad ievadparametriem ir katras ekvivalences klases ekstremālās vērtības (maksimālā, minimālā), mazliet mazākas (par mazu vērtību ε) vai mazliet lielākas nekā ekstremālās vērtības, kā arī kad vērtība ir nomināla – tā atrodas ekvivalences klases iekšienē ievērojami tālāk no klases galapunktiem nekā par attālumu ε .

Robežvērtību testēšana

- Vērtību ε nosaka testētājs atkarībā no domēna datu tipa un specifikācijas.
- Piemēram, veseliem skaitļiem ε būs 1, bet vērtība, kas izsaka naudas vienības latos ar precizitāti līdz santīmam kā ε ir lietderīgi pieņemt 0.01.

Domēntestēšanas metodes (vai domēnanalīzes metodes)

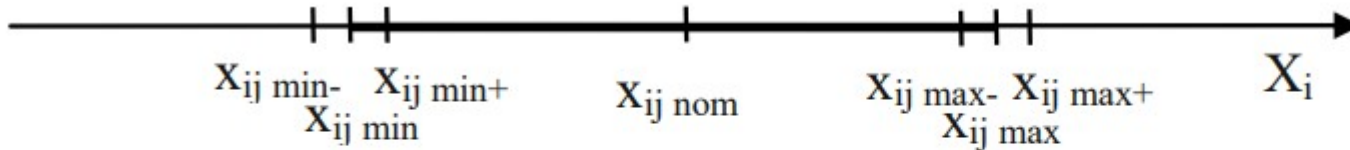
- Šīs metodes ņem vērā arī savstarpējo atkarību vai mijiedarbību starp programmas ievadparametriem.
- Šajās metodēs ievaddomēnu parasti aplūko kā ģeometrisku figūru, bet tās šķautnes uzskata par robežām.
- Vairumā gadījumu tiek analizēti domēni ar lineārām robežām, taču ir metodes, kas atļauj pētīt arī nelineārās robežas.

Programmas piemērs

- Pieņemsim, ka P ir programma ar N ievadmainīgajiem X_i un kur $1 \leq i \leq N$.
- Katram parametram ievaddomēns D_i ir sadalīts M_i ekvivalences klasēs un to galapunkti ir robežvērtības.

Ekvivalences klases grafiks

- Katram parametram X_i atbilstošā domēna D_i , katru sakārtotu elementu ekvivalences klasi d_{ij} attēlo:



2.1. att. Ekvivalences klase d_{ij} : $[x_{ij \min}, x_{ij \max}]$, tās robežvērtības $x_{ij \min}$, $x_{ij \max}$, iekšējie OFF punkti $x_{ij \min+}$, $x_{ij \max-}$, ārējie OFF punkti $x_{ij \min-}$, $x_{ij \max+}$ un nominālā vērtība $x_{ij \text{ nom}}$

Ekvivalences klases grafiks

- Klases minimālā robežvērtība ir $x_{ij \min}$, maksimālā robežvērtība ir $x_{ij \max}$, kur $1 \leq j \leq M_i$.
- Klases nominālā vērtība ir $x_{ij \text{ nom}}$.
- Vērtības $x_{ij \min-}$, $x_{ij \max-}$ ir mazliet mazākas par atbilstošajām robežvērtībām.
- Vērtības $x_{ij \min+}$, $x_{ij \max+}$ ir mazliet lielākas par atbilstošajām robežvērtībām.

Divi pieņēmumi

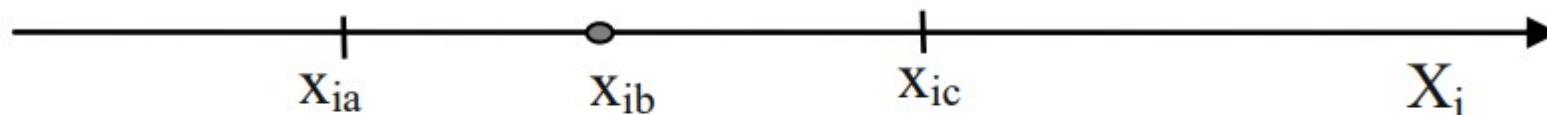
Katram parametram X_i visu ekvivalences klašu apvienojums veido domēnu D_i .

Tādējādi, $D_i = \bigcup_{j=1}^{M_i} d_{ij} \quad \forall i, j$, kur $1 \leq i \leq N$ un $1 \leq j \leq M_i$.

Ekvivalences klasēm nav kopīgu vērtību, tas ir, to šķēlums ir tukša kopa – $\forall i, j, k$,

kur $1 \leq i \leq N$, $1 \leq j \leq M_i$, $1 \leq k \leq M_i$ un $j \neq k$ $d_{ij} \cap d_{ik} = \emptyset$.

Ekvivalences klasēm var būt kopīgas robežvērtības



2.2. att. Kopējā robežvērtība x_{ib} intervāliem $[x_{ia}, x_{ib}]$ un $(x_{ib}, x_{ic}]$, $L_i=1$

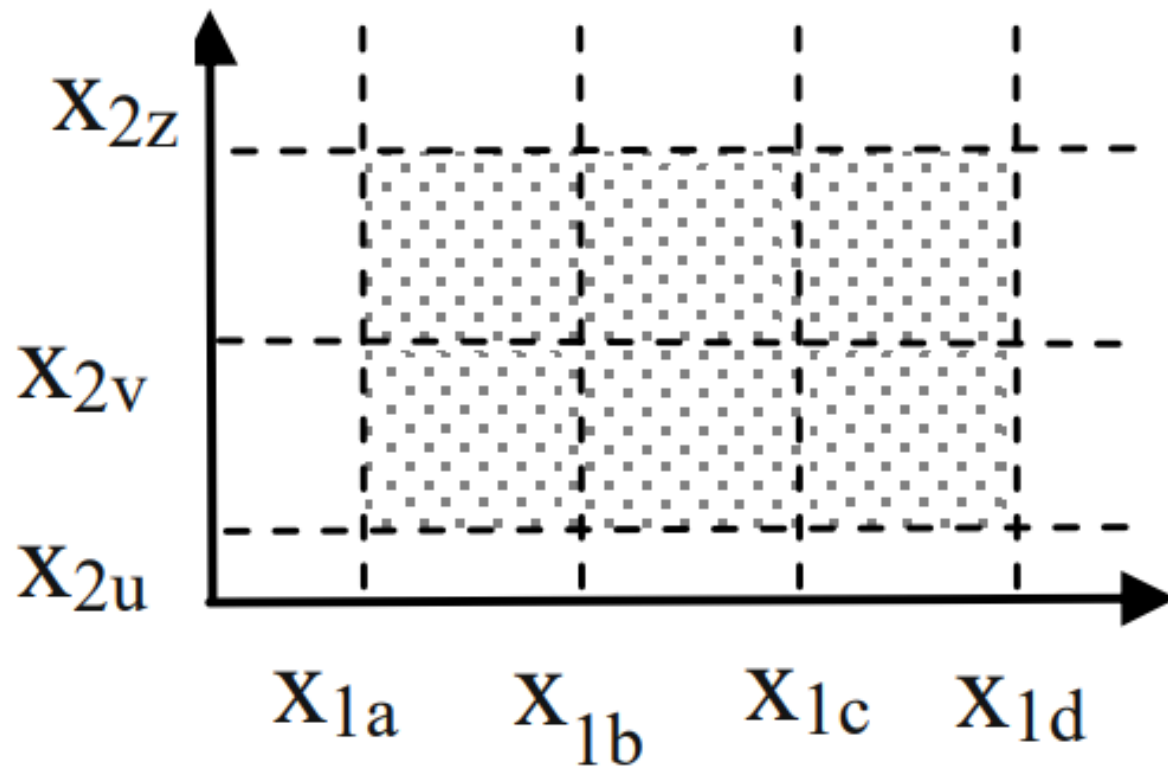
Kā novērtēt testēšanas sarežģītību?

- Nav vienas vispāratzītas programmatūras testēšanas metožu sarežģītības metrikas.
- Testēšanas sarežģītību kā testpiemēru skaitu, kas nepieciešami, lai demonstrētu programmas korektību.

Ekvivalences klašu testēšanas metožu sarežģītība

- Programma P ar divi parametriem: X_1 un X_2 .
- Parametra X_1 domēns ir intervāls $[x_{1a}, x_{1d}]$, kurš ir sadalīts trīs ekvivalences klasēs - $[x_{1a}, x_{1b})$, $[x_{1b}, x_{1c})$, un $[x_{1c}, x_{1d}]$.
- Parametra X_2 domēns ir intervāls $[x_{2u}, x_{2z}]$, kurš ir sadalīts divās ekvivalences klasēs - $[x_{2u}, x_{2v})$ un $[x_{2v}, x_{2z}]$.

X1 un X2 domēni

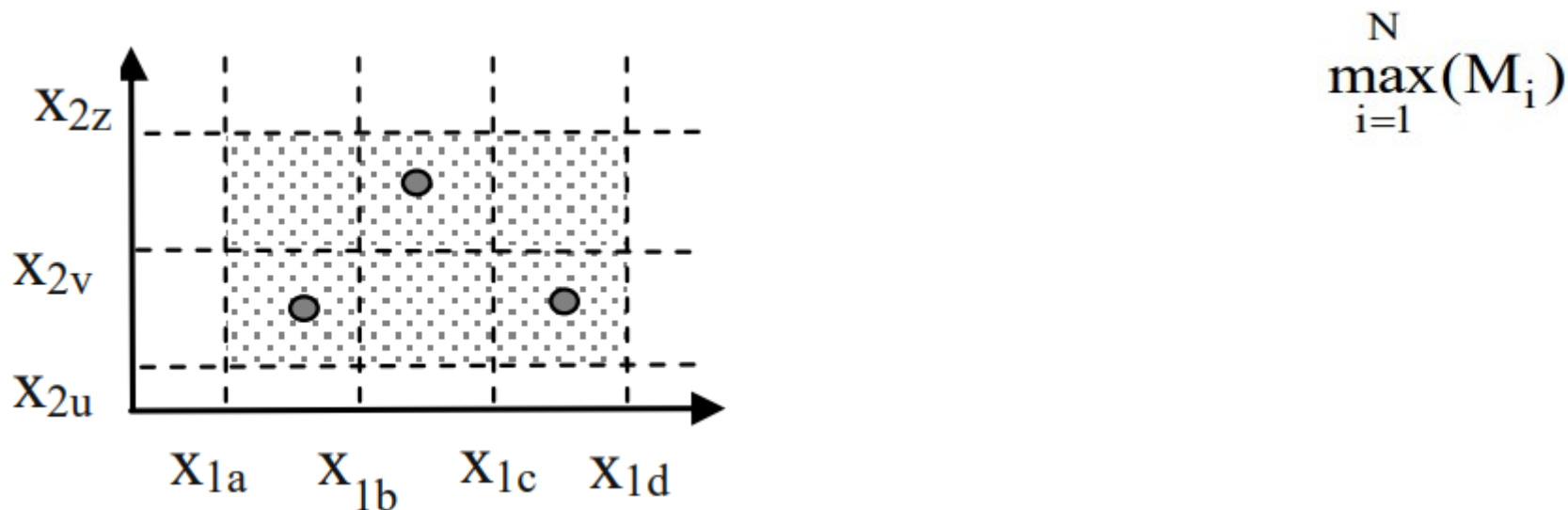


1.Vājā ekvivalences klašu testēšanas metode

- Pārbauda vienu reprezentantu no katra parametra pieļaujamo vērtību katras ekvivalences klases.
- Metodes pamatā ir vienas kļūdas pieņēmums.
- To lietojot, tiek pieņemts, ka ir pietiekami, ja katru ekvivalences klasi testē vienreiz.

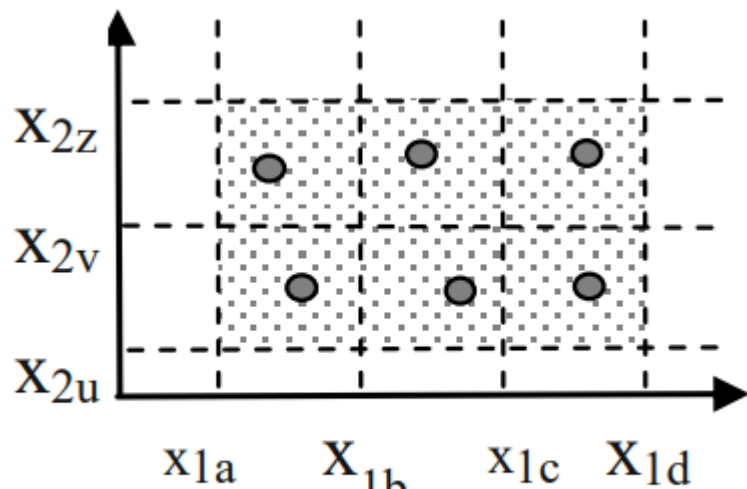
1. Vājā ekvivalences klašu testēšanas metode

- Vispārīgi, ja programmai ir N parametri, vājā ekvivalences klašu testēšanas metode rada testkomplektu, kura apjoms ir



2. Stiprā ekvivalences klašu testēšanas metode

- Stiprā ekvivalences klašu testēšanas metode izmanto N kļūdu pieņēmumu.
- Testkomplektā iekļauj testpiemēru no visu parametru ekvivalences klašu Dekarta reizinājuma.



$$\prod_{i=1}^N M_i$$

3. Robustā vājā ekvivalences klašu testēšanas metode

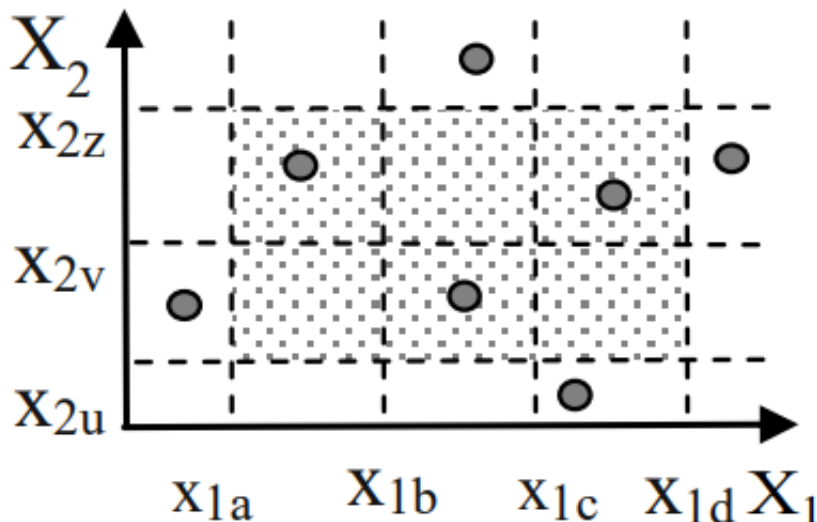
- 1) pieļaujamajām vērtībām izvēlas tikai vienu vērtību no katras ekvivalences klases, turklāt katrā testpiemērā izmanto tikai pieļaujamās vērtības;
- 2) nepieļaujamajām vērtībām izvēlas vienu vērtību no katras ekvivalences klases un katrā testpiemērā vienam parametram piemēro vērtību no nepieļaujamajām vērtībām, bet pārējiem visiem – pieļaujamās vērtības.
- Metode neļauj izmantot nepieļaujamās vērtības vienā testpiemērā diviem vai vairākiem parametriem

3. Robustā vājā ekvivalences klašu testēšanas metode

- N parametru gadījumā ģenerēto testpiemēru skaits ir

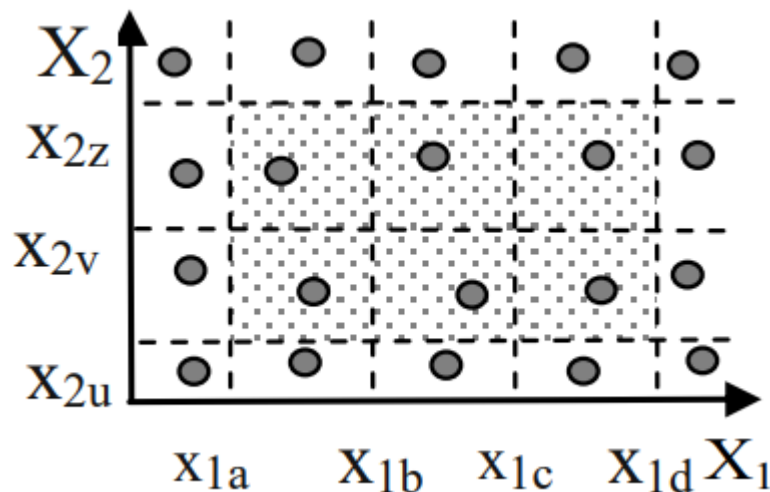
$$\max_{i=1}^N (M_i) + \sum_{i=1}^N Q_i$$

,kur Q_i ir nepieļaujamo vērtību ekvivalences klašu kopas apjoms parametram X_i



4. Robustā stiprā ekvivalences klašu testēšanas metode

- Testkomplektā iekļauj testpiemēru no visu parametru visu - gan pieļaujamo, gan nepieļaujamo vērtību ekvivalences klašu Dekarta reizinājumu.
- N parametru gadījumā ģenerēto testpiemēru skaits ir $\prod_{i=1}^N (M_i + Q_i)$

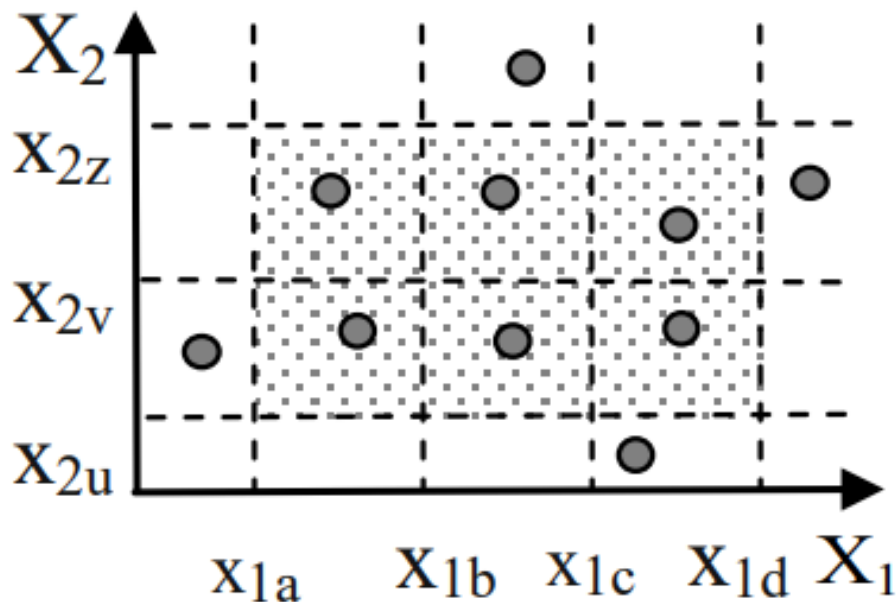


5. Robustā jauktā ekvivalences klašu testēšanas metode

- Testkomplektā iekļauj pa testpiemēram no visu parametru pieļaujamo vērtību ekvivalences klašu Dekarta reizinājuma.
- No nepieļaujamo vērtību ekvivalences klasēm ņem pa vienam reprezentantam un veido testpiemērus, kuros vienam parametram ir nepieļaujama vērtība, bet visiem pārējiem – pieļaujamās.
- Nav pieļauts, ja diviem vai vairāk parametriem vienā testpiemērā ir nepieļaujamās vērtības.

5. Robustā jauktā ekvivalences klašu testēšanas metode

- N parametru gadījumā ģenerēto testpiemēru skaits ir $\prod_{i=1}^N M_i + \sum_{i=1}^N Q_i$

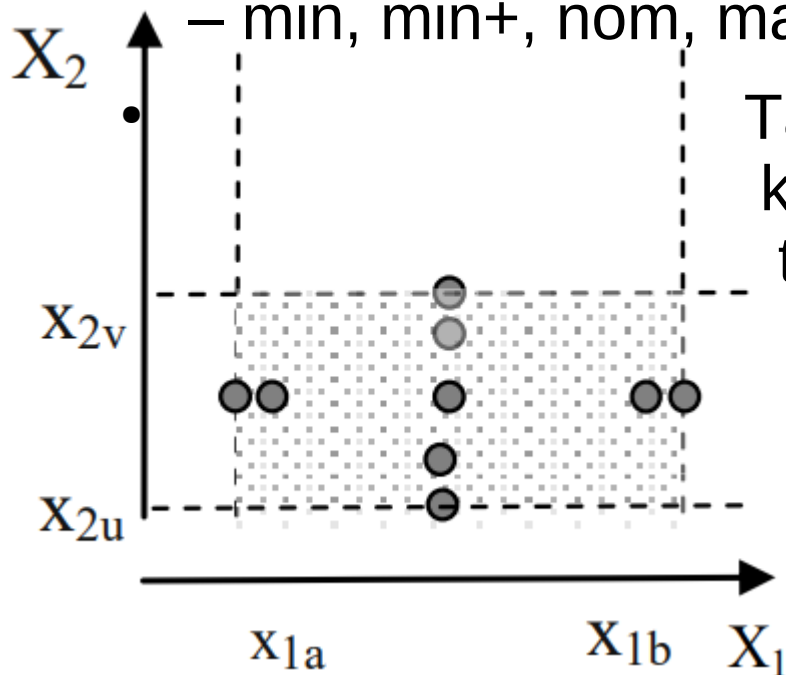


1.Vājā iekšējā robežvērtību testēšanas metode

- Pārbauda ekvivalences klašu robežvērtības, iekšējos OFF punktus un nominālās vērtības.
- Metodes pamatā ir vienas kļūdas pieņēmums – katrā testpiemērā vienam parametram ir pārbaudāmā vērtība, bet pārējiem – nominālās.

1. Vājā iekšējā robežvērtību testēšanas metode

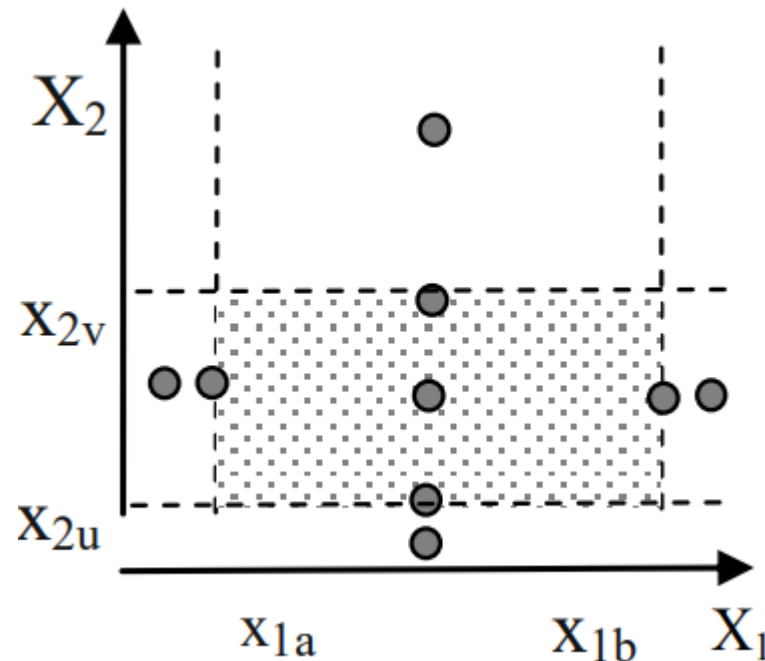
- Ja programmai ir tikai viens parametrs X_1 , iegūstam 5 testpiemērus no katras pieļaujamo vērtību ekvivalences klases – min, min+, nom, max-, max.



Tā kā pieļaujamo vērtību ekvivalences klašu skaits ir M_1 , testpiemēru skaits testkomplektā ir $5M_1$

2.Vājā ārējā robežvērtību testēšanas metode

- Vājā ārējā robežvērtību testēšanas metode pārbauda ekvivalences klašu robežvērtības, ārējos OFF punktus un nominālos punktus:



2. Vājā ārējā robežvērtību testēšanas metode

- Ģenerētā testkomplekta apjoms ir tāds pats kā vājajai iekšējo RVT metodei – tas nav mazāks par

$$(4N + 1) \prod_{i=1}^N M_i - \sum_{i=1}^N (L_i \prod_{\substack{j=1 \\ j \neq i}}^N M_j)$$

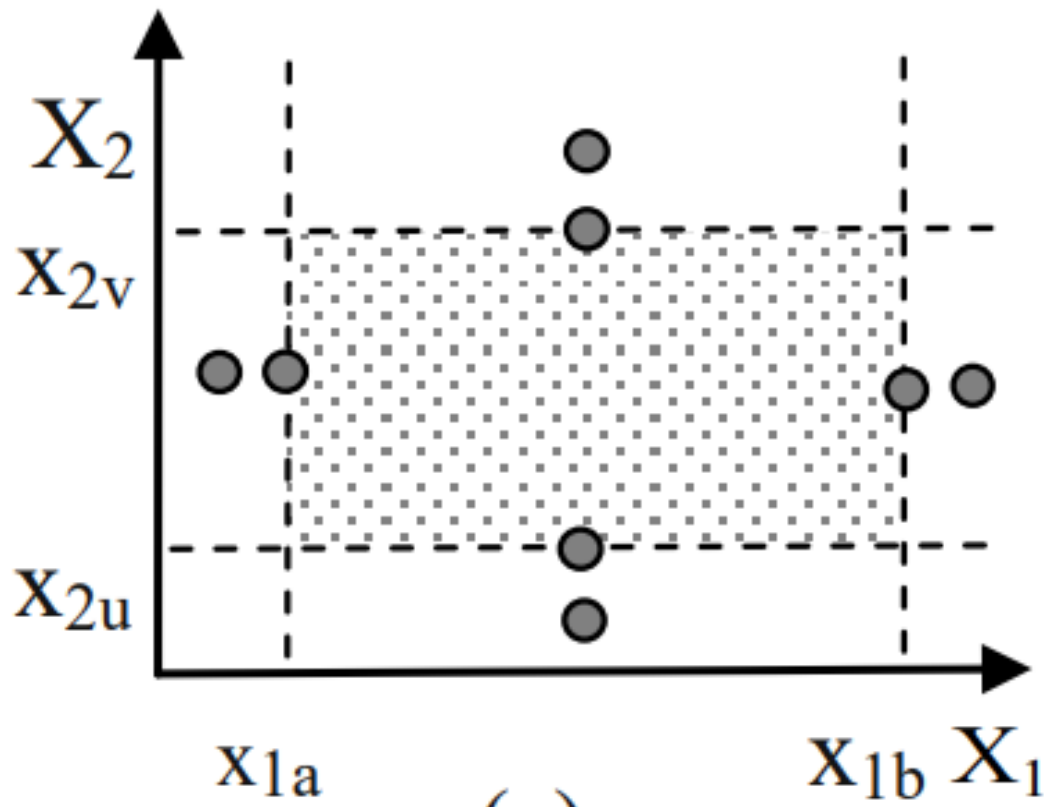
testpiemēriem un nav lielāks par

$$(4N + 1) \prod_{i=1}^N M_i \text{ testpiemēriem.}$$

3. Vājā vienkāršā ārējā robežvērtību testēšanas metode

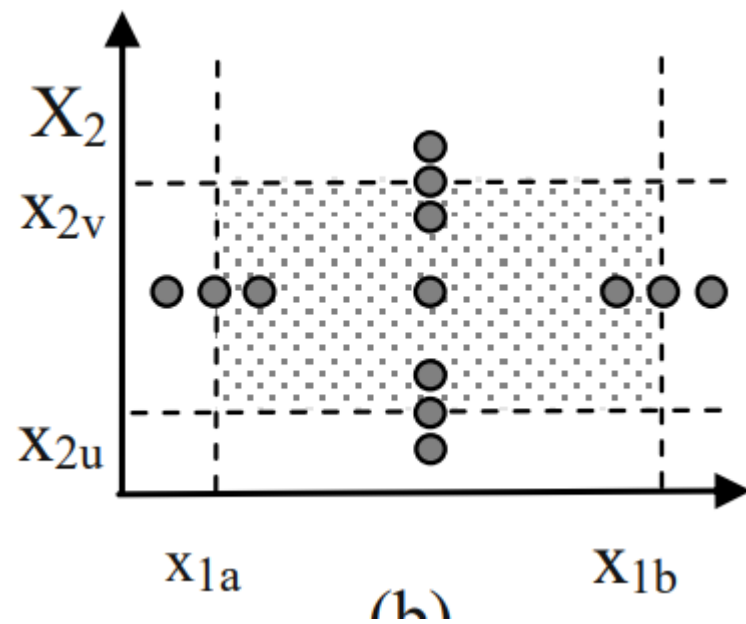
- Vājā vienkāršā ārējā robežvērtību testēšanas metode pārbauda ekvivalences klašu robežvērtības un ārējos OFF punktus.
- Atšķirībā no vājās ārējās RVT metodes, šī metode neizmanto nominālpunktus.
- Ģenerētā testkomplekta apjoms ir mazāks nekā vājai iekšējai RVT vai vājajai ārējai RVT metodei – apakšējā sarežģītības robeža ir $4N \prod_{i=1}^N M_i - \sum_{i=1}^N (L_i \prod_{\substack{j=1 \\ j \neq i}}^N M_j)$ testpiemēri, bet augšējā - $4N \prod_{i=1}^N M_i$

3. Vājā vienkāršā ārējā robežvērtību testēšanas metode



4. Robustā vājā robežvērtību testēšanas metode

- Ģenerētā testkomplekta apjomu var aprēķināt līdzīgi kā vājajai iekšējai RVT metodei – tas būs ne mazāks par $(6N+1)\prod_{i=1}^N M_i - \sum_{i=1}^N (L_i \prod_{\substack{j=1 \\ j \neq i}}^N M_j)$ testpiemēriem un ne lielāks par $(6N+1)\prod_{i=1}^N M_i$ testpiemēriem.



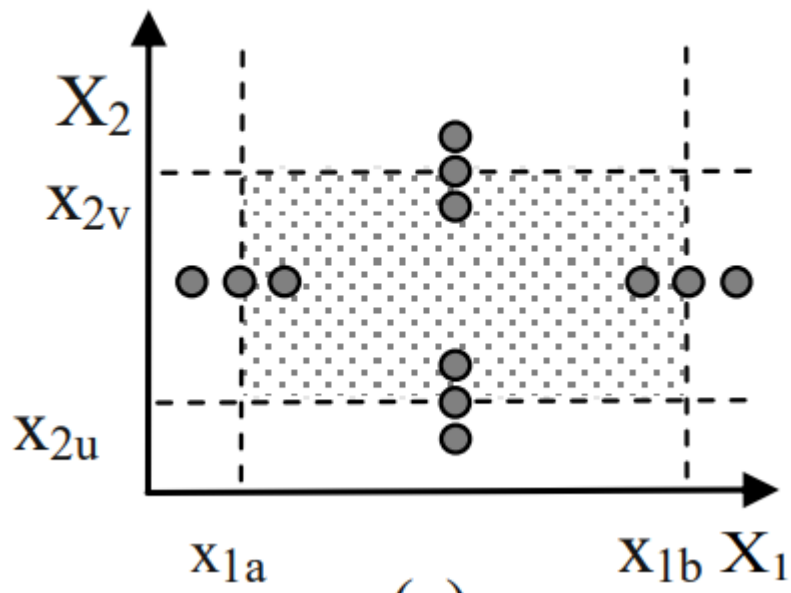
5. Robustā vājā vienkāršā robežvērtību testēšanas metode

- Ģenerētā testkomplekta apjomu aprēķina līdzīgi kā vājajai iekšējai RVT metodei, iegūstot ne mazāk kā

$$6N \prod_{i=1}^N M_i - \sum_{i=1}^N (L_i \prod_{\substack{j=1 \\ j \neq i}}^N M_j)$$

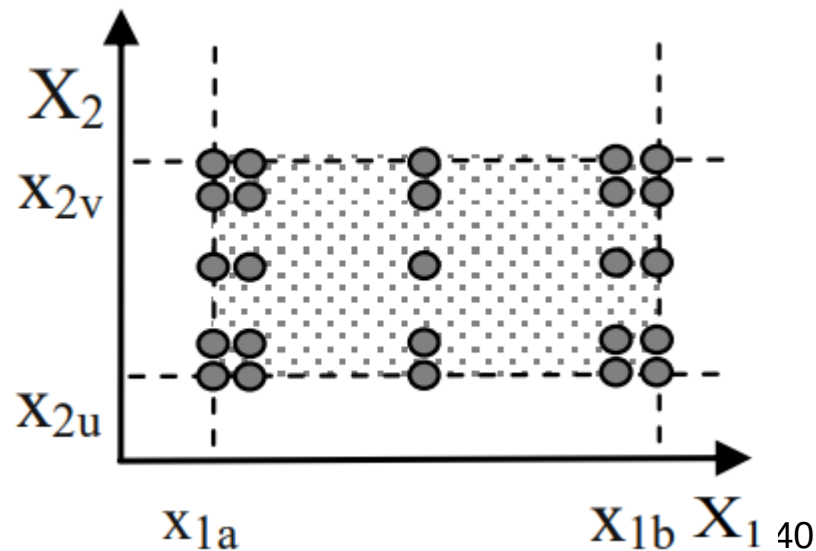
testpiemērus un ne vairāk kā

$$6N \prod_{i=1}^N M_i \text{ testpiemērus.}$$



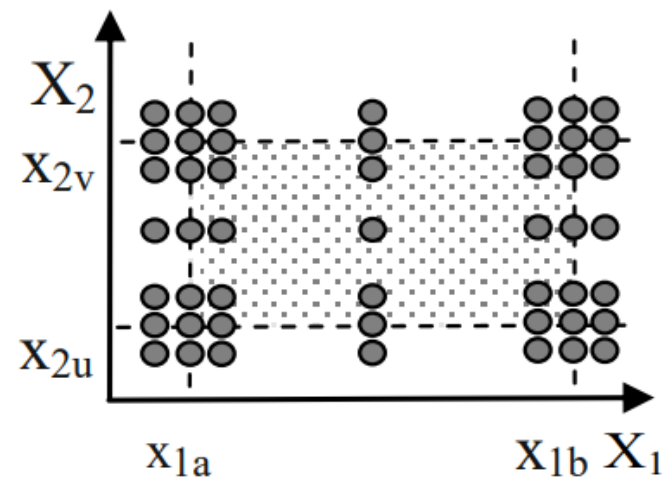
6. Sliktākā gadījuma robežvērtību testēšanas metode

- Ja programmai ir tikai viens parametrs X_1 , iegūst 5 testpiemērus no katras ekvivalences klases. Pēc tam izslēdz no testkomplekta liekos testpiemērus, kas radušies gadījumos, kad ekvivalences klasēm ir kopīgas robežvērtības.
- Tādējādi viena parametra gadījumā
- testkomplektā iegūst $5M_1 - L_1$ testpiemērus.



7. Robustā sliktākā gadījuma robežvērtību testēšanas metode

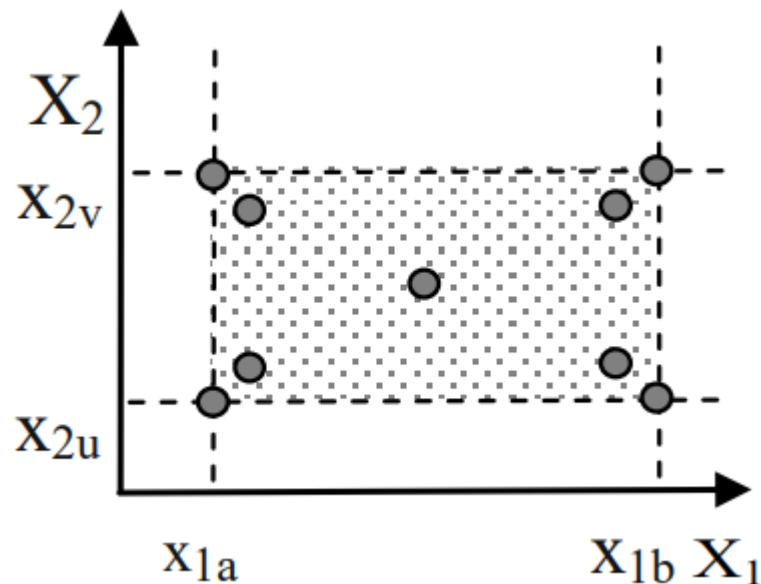
- Testkomplekta veidošanas algoritms ir analogisks kā sliktākā gadījuma robežvērtību testēšanas metodei, programmai ar N parametriem ieqūstot vismaz $\prod_{i=1}^N (7M_i - 2L_i)$ testpiemērus un ne vairāk kā $7^N \prod_{i=1}^N M_i$ testpiemērus.



8.Stūra vājā iekšējā robežvērtību testēšanas metode

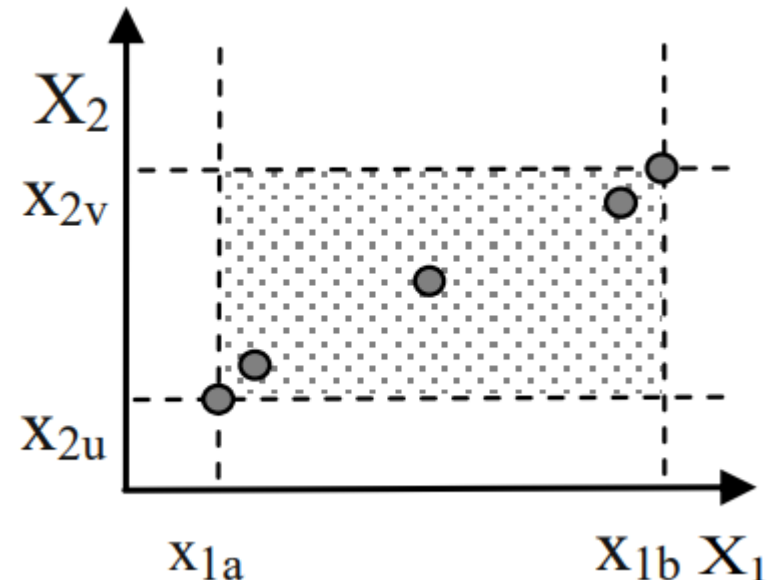
- Darbojas saskaņā ar daudzu kļūdu pieņēmumu.
- Tā testē gadījumus, kad visiem parametriem ir viena veida īpašās vērtības – robežvērtības, iekšējie OFF punkti vai nominālās vērtības.

$$(2^{N+1} + 1) \prod_{i=1}^N M_i - 2 \sum_{i=1}^N (L_i \prod_{\substack{j=1 \\ j \neq i}}^N M_j)$$



9. Vājā diagonālā iekšējā robežvērtību testēšanas metode

- Vājā diagonālā iekšējā RVT metode arī atbilst daudzu kļūdu pieņemumam, taču tā testē gadījumus, kad visiem programmas parametriem ir viena tipa speciālās vērtības, piemēram, ja tās ir robežvērtības, tad visiem parametriem ir maksimālās vērtības, līdzīgi arī ar iekšējiem OFF punktiem un nominālvērtībām



10. Vājā diagonālā iekšējā robežvērtību testēšanas metode

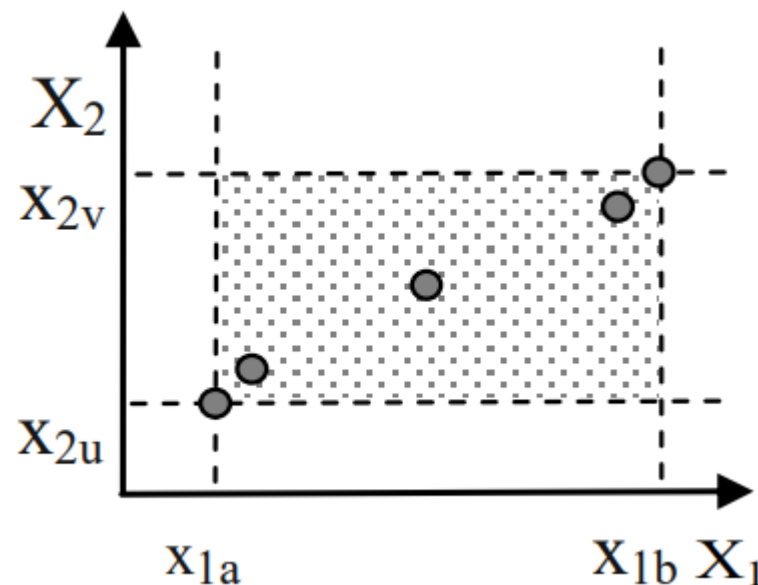
- N parametru gadījumā vājā diagonālā iekšējā RVT metode testkomplektā dod vismaz

$$5 \prod_{i=1}^N M_i - \prod_{i=1}^N L_i$$

testpiemērus

un ne vairāk kā

$$5 \prod_{i=1}^N M_i$$

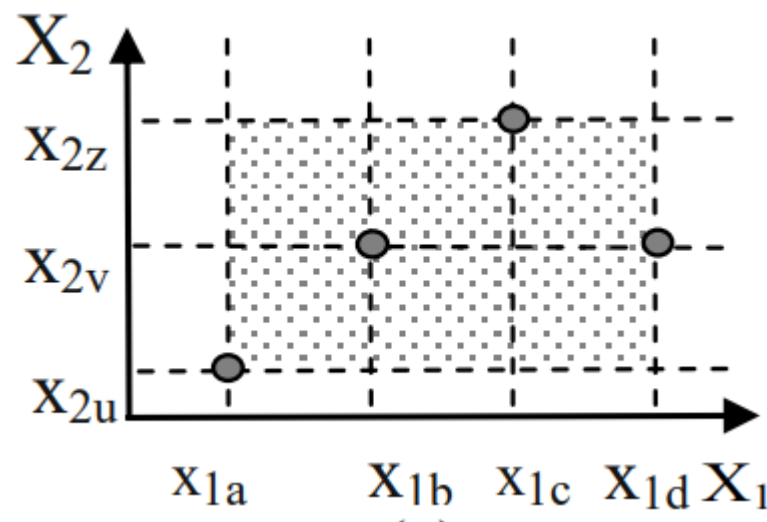


11. Daudzdimensiju robežvērtību testēšanas metode

- N parametru gadījuma testkomplektu veidos testpiemēri jeb ne vairāk kā

$$\max_{i=1}^N (2M_i - L_i)$$

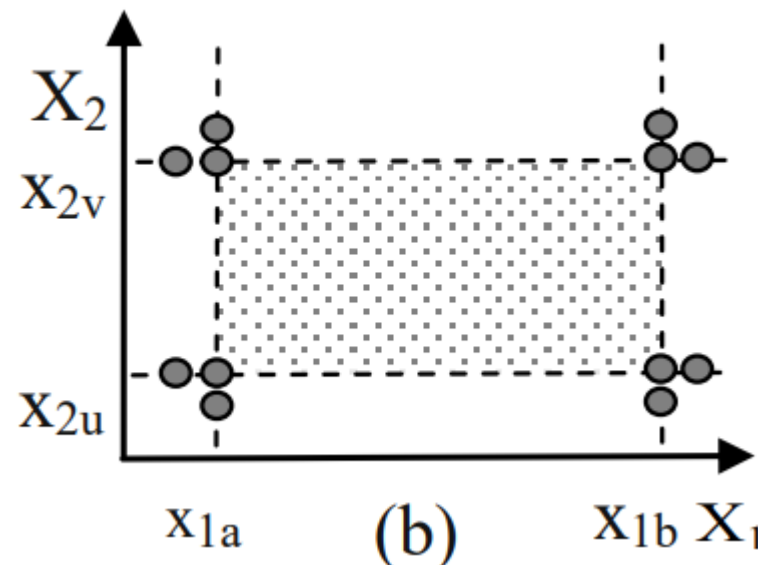
$\max_{i=1}^N (2M_i)$ testpiemēri, ja
nav pārkāpušos robežvērtību.



12. Robustā stūra ārējā robežvērtību testēšanas metode

- Vismaz testpiemēri.
- Augšējā robeža ir

$$\sum_{i=1}^N (2M_i \prod_{\substack{j=1 \\ j \neq i}}^N (2M_j - L_j)) + \prod_{i=1}^N (2M_i - L_i)$$

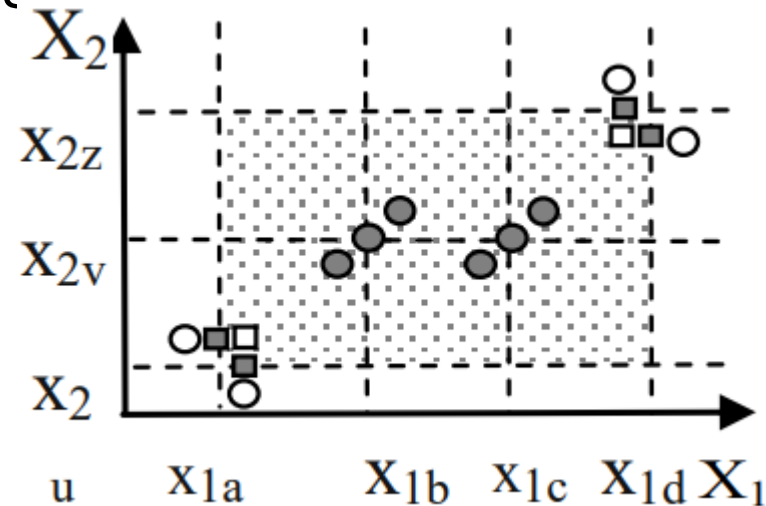
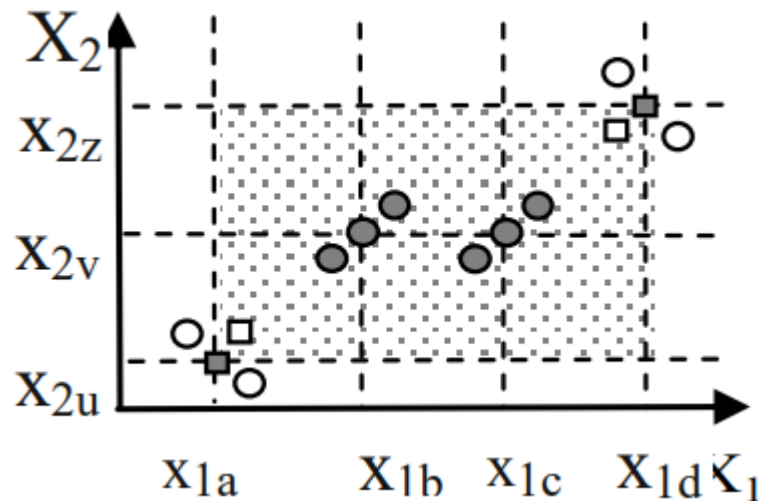


$$\prod_{i=1}^N 2M_i + \sum_{i=1}^N (2M_i \prod_{\substack{j=1 \\ j \neq i}}^N 2M_j) = 2^N \prod_{i=1}^N M_i + \sum_{i=1}^N (2^N \prod_{j=1}^N M_j) = 2^N (N+1) \prod_{i=1}^N M_i$$

13. Robustā stiprā robežvērtību testēšana

- Izmanto pieļaujamo vērtību ekvivalences klašu visas robežvērtības, iekšējos un ārējos OFF punktus un sadala tās divās kopās:

visas pieļaujamās vērtības ; visas nepieļaujamās vērtības



13. Robustā stiprā robežvērtību testēšana

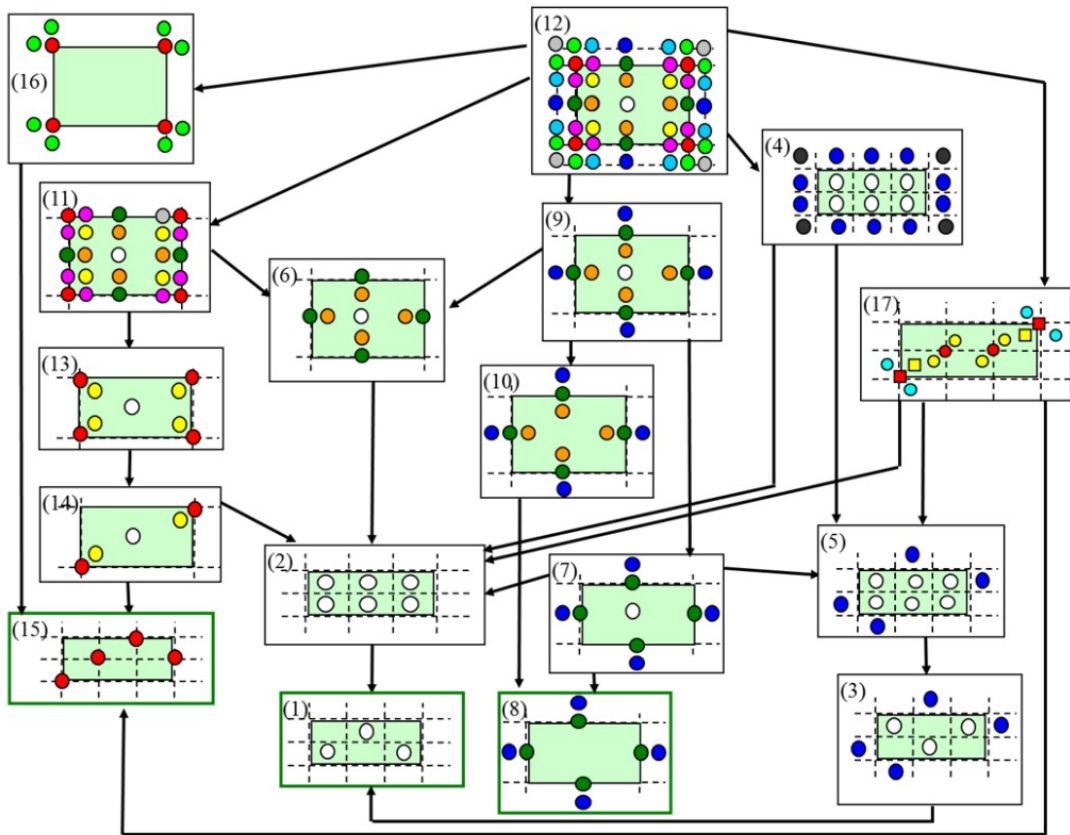
- Apakšējā sarežģītības robeža

$$\begin{aligned} & \max_{i=1}^N (2M_i + L_i) + \sum_{i=1}^N (2M_i - 2L_i) + \max_{i=1}^N (2M_i - 2L_i) = \\ & = \max_{i=1}^N (2M_i + L_i) + 2 \sum_{i=1}^N (M_i - L_i) + 2 \max_{i=1}^N (M_i - L_i) \end{aligned}$$

- Augšējā robeža

$$\max_{i=1}^N (2M_i + L_i) + \sum_{i=1}^N (2M_i - 2L_i) + \sum_{i=1}^N (2M_i - 2L_i) = \max_{i=1}^N (2M_i + L_i) + 4 \sum_{i=1}^N (M_i - L_i) .$$

Domēstestēšanas metožu hierarhija saskaņā ar iekļautības principu



Npk.	Metode	Sarežģītība	
		Apakšējā robeža	Augšējā robeža
1.	Vājā EKT metode	$\max_{i=1}^N (M_i)$	Sakrīt ar apakšējo robežu
2.	Stiprā EKT metode	$\prod_{i=1}^N M_i$	Sakrīt ar apakšējo robežu
3.	Robustā vājā EKT metode	$\max_{i=1}^N (M_i) + \sum_{i=1}^N Q_i$	Sakrīt ar apakšējo robežu
4.	Robustā stiprā EKT metode	$\prod_{i=1}^N (M_i + Q_i)$	Sakrīt ar apakšējo robežu
5.	Robustā jauktā EKT metode	$\prod_{i=1}^N M_i + \sum_{i=1}^N Q_i$	Sakrīt ar apakšējo robežu
6.	Vājā iekšējā RVT metode	$(4N+1) \prod_{i=1}^N M_i - \sum_{i=1}^N (L_i \prod_{\substack{j=1 \\ j \neq i}}^N M_j)$	$(4N+1) \prod_{i=1}^N M_i$
7.	Vājā ārējā RVT metode	$(4N+1) \prod_{i=1}^N M_i - \sum_{i=1}^N (L_i \prod_{\substack{j=1 \\ j \neq i}}^N M_j)$	$(4N+1) \prod_{i=1}^N M_i$
8.	Vājā vienkāršā ārējā RVT metode	$4N \prod_{i=1}^N M_i - \sum_{i=1}^N (L_i \prod_{\substack{j=1 \\ j \neq i}}^N M_j)$	$4N \prod_{i=1}^N M_i$
9.	Robustā vājā RVT metode	$(6N+1) \prod_{i=1}^N M_i - 2 \sum_{i=1}^N (L_i \prod_{\substack{j=1 \\ j \neq i}}^N M_j)$	$(6N+1) \prod_{i=1}^N M_i$

Npk.	Metode	Sarežģītība	
		Apakšējā robeža	Augšējā robeža
10.	Robustā vienkāršā vājā	$6N \prod_{i=1}^N M_i - 2 \sum_{i=1}^N (L_i \prod_{\substack{j=1 \\ j \neq i}}^N M_j)$	$6N \prod_{i=1}^N M_i$
11.	Sliktākā gadījuma RVT metode	$\prod_{i=1}^N (5M_i - L_i)$	$5^N \prod_{i=1}^N M_i$
12.	Robustā sliktākā gadījuma RVT metode	$\prod_{i=1}^N (7M_i - 2L_i)$	$7^N \prod_{i=1}^N M_i$
13.	Vājā stūra iekšējā RVT metode	$(2^{N+1} + 1) \prod_{i=1}^N M_i - 2 \sum_{i=1}^N (L_i \prod_{\substack{j=1 \\ j \neq i}}^N M_j)$	$(2^{N+1} + 1) \prod_{i=1}^N M_i$
14.	Vājā diagonālā iekšējā RVT metode	$5 \prod_{i=1}^N M_i - \prod_{i=1}^N L_i$	$5 \prod_{i=1}^N M_i$
15.	Daudzdimensionālā RVT metode	$\max_{i=1}^N (2M_i - L_i)$	$2 \max_{i=1}^N (M_i)$
16.	Robustā stūra ārējā RVT metode	$\sum_{i=1}^N (2M_i \prod_{\substack{j=1 \\ j \neq i}}^N (2M_j - L_j)) + \prod_{i=1}^N (2M_i - L_i)$	$2^N (N+1) \prod_{i=1}^N M_i$
17.	Robustā stiprā RVT metode	$\max_{i=1}^N (2M_i + L_i) + 2 \sum_{i=1}^N (M_i - L_i) + 2 \max_{i=1}^N (M_i - L_i)$	$\max_{i=1}^N (2M_i + L_i) + 4 \sum_{i=1}^N (M_i - L_i)$

Kopsavilkums

- Programmatūras testēšanas tehnoloģiskā sarežģītība ir tik liela, ka atbilstoši domēntestēšanas metodei nav iespējams pilnībā notestēt vairumu praksē sastopamo programmatūru.
- Darbā tiek piedāvāts līdz šim testēšanas teorijā nebijis domēntestēšanas metožu apkopojums un to iekļautības hierarhija.