Optimalizace sady testů

Radek Mařík

ČVUT FEL, K13132

20. října 2016

Obsah

Příprava testovacích skriptů - volání funkcí/metod/vlastností

- podobné systému "Expect" (GNU),
- předpoklad bezestavového chování funkce,
- potřebné prostředí:
  - příprava struktur parametrů,
  - inicializace objektu,
  - (jednoduché nastavení stavu objektu),
- hodnoty parametrů,
- odvození úplné sady parametrů může být založeno na teorii rámců (UI),
- testovací skript jako parametrizovaná zafixovaná sekvence funkčních volání, tzv. volání *super*-funkce
  1. testScript0(A, B, C)
     1. function1(A, B)
     2. function2(B, C)
     3. function3(C, C)

Testování kombinací parametrů funkce

- testovaná funkce,
- množina parametrů funkce,
- množina významných hodnot pro každý parametr,
    - hraniční podmínky.

- **Idealní testovací plán**
    - všechny možné kombinace

- **Praktický testovací plán**
    - selhání jsou způsobena interakcí pouze několika parametrů,
    - testování kombinacemi pokrývající všechny možné k-tice,
    - $k \in \{1, 2, 3\}$
    - k-tice komprimovány do plných kombinací.

Příklad

- PLC (Programable Logic Controller) instrukce *sbr*,
- 5 parametrů,
- 17 hodnot pro každý parametr odpovídající různým adresním modům, I:500, O:[N7:0], #L[B3:0]:[N7:0], atd.

Testovací plán:

- Ideální testovací plán
  - $1.419.857 = 17^5$ kombinací
  - 230 dnů testování
- Praktický testovací plán:
  - Nezávislé parametry: 17 kombinací
  - Párová závislost parametrů:
    - $10 = 4 + 3 + 2 + 1$ parametrových párů,
    - $289 = 17^2$ kombinací hodnot pro každý pár,
    - $2890 = 289 \times 10$ párů parametrových hodnot,
    - 289 kombinací obsahující všechny páry,
    - 30 minut

Cíle OO testování [SPK+99]

- Prvním cílem je redukce počtu testovacích případů, které se musí vytvořit.
- Druhým cílem je minimalizace počtu testovacích případů, které se musí provést a jejichž výsledky musí být ověřeny.
- Testovací případy objektově orientovaných systému mají tvar: <*initial state, message(s), final state*>

Počet OO testovacích případů

- volající objekt
  - protokol chování objektu,
  - předpokládané vstupní podmínky,
  - ověřované vstupní podmínky - zpracování vyjímek.
- volaný objekt
  - protokol chování objektu,
  - splnění výstupních podmínek.
- členy třídy
- parametry
- stavy

PŘÍLIŠ MNOHO TESTŮ

- selekce testovacích případů tak, aby se získalo maximální pokrytí s minimem počtu testovacích případů.

Návrh testovacího plánu

- **Brutální síla**
    - kombinatorická exploze,
    - nemožné vypočítat v rozumném čase.
- **Ortogonální pole**
    - obtíže s neplatnými kombinacemi,
    - testovací případy nemusí být balancované,
        - Každý pár je pokryt stejným počtem testů.
    - vhodné pro návrh s 2-3 hodnotami parametrů,
    - pole je obtížné najít
        - katalogy,
        - simulované žíhání,
        - kritérium maximální entropie.
- **Projektivní rovina**
    - pojem Latinského čtverce,
    - pojem konečné projektivní roviny,

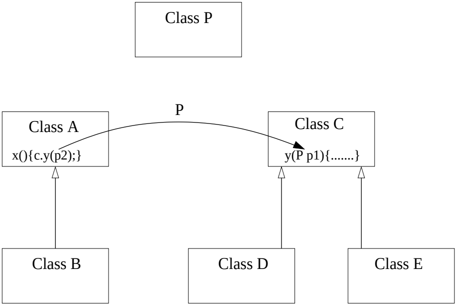
Ortogonální pole [Mon91, Ros88, LW89, SPK+99]

- **Faktor:** entita, parametr, na kterém závisí testovací případ.
- **Úroveň:** každý faktor má konečný počet možných hodnot nazývaných úrovněmi.
- *Příklad:*
  - tři faktory A, B, C a každý faktor má tři úrovně $1, 2, 3$.
  - existuje $27 = 3^3$ možných kombinací (testovacích případů)
  - existuje pouze 9 párových kombinací testující interakce vzájemné interakce faktorů.

trial	A	B	C
1	1	1	3
2	1	2	2
3	1	3	1
4	2	1	2
5	2	2	1
6	2	3	3
7	3	1	1
8	3	2	3
9	3	3	2

Testování interakce tříd [SPK+99]

- Otestuj interakci tříd "*A* hierarchie" se třídami "*C* hierarchie" při zprávě *y* s parametrem třídy *P*. Sama *C* třída je abstraktní.



Třída	A	B	C	D	E	P
# stavů	2	3	2	3	3	2

Testování pomocí ortogonálních polí I [SPK+99]

1. Orthogonal Array Testing (OATS)
2. Zobrazení hierarchií tříd do ortogonálních polí:
   - Vysílající hierarchie je jeden faktor.
   - Přijímací hierarchie je druhý faktor.
   - Každý parametr zprávy generuje další faktor.
   - Možné stavy faktoru zdvojnásobují počet faktorů.
   - *Příklad:* 6 faktorů = **A** hierarchie, **P** hierarchie, a **C** hierarchie a stav objektu pro každou třídu.
3. Výběr úrovní:
   - Maximální počet možných hodnot každého faktoru definuje počet úrovní.
   - *Příklad:*
     - 1 faktor má 1 úroveň (P),
     - 2 faktory mají maximálně 2 úrovně (A, C),
     - 3 faktory mají maximálně 3 úrovně (stavy).

Testování pomocí ortogonálních polí II [SPK+99]

1. Použití standardních polí:
    - Vyber nejmenší standardní pole, které řeší problém.
    - *Příklad:*
        - 6 faktorů s maximálně 3 úrovněmi,
        - standardní pole L_{18}, které má 1 faktor s 2 úrovněmi a 7 faktorů se 3 úrovněmi.
        - $2^1 \times 3^7$
2. Vytvoř zobrazení mezi standardním polem a daným problémem:
    - Úrovně každého faktoru jsou zobrazeny do čísel standardního pole.
    - *Příklad:*
        - Hierarchie třídy A:

Hodnoty domény	A	B
Hodnoty pole	1	2

        - Stavy hierarchie třídy P

Hodnoty domény	P,State1	P, State2	P, State2
Hodnoty pole	1	2	3

Vzorky ortogonálních polí [Mon91, Ros88, LW89, SPK+99]

$L_4(2^3)$

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

$L_8(2^7)$

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

Ortogonální pole $L_{18}(2^1 \times 3^7)$ [Mon91, Ros88, LW89, SPK+99]

$L_{18}(2^1 \times 3^7)$

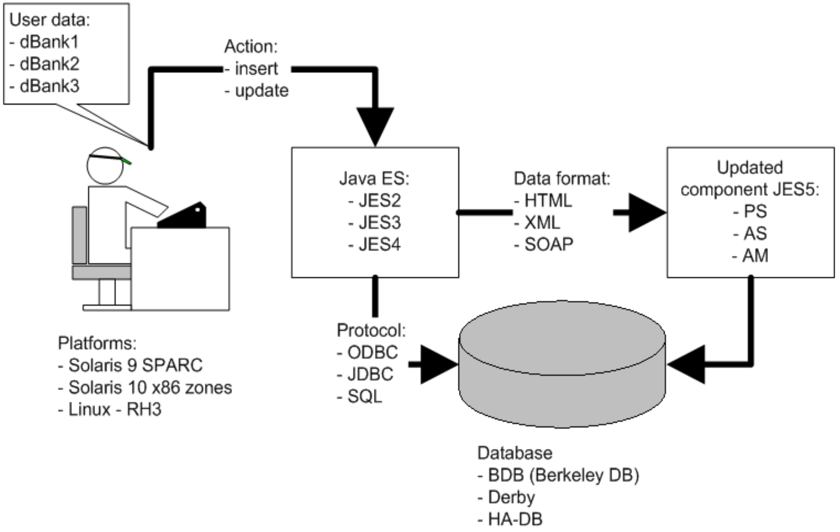
	1	2	3	4	5	6	7	8
1	1	1	1	1	1	1	1	1
2	1	1	2	2	2	2	2	2
3	1	1	3	3	3	3	3	3
4	1	2	1	1	2	2	3	3
5	1	2	2	2	3	3	1	1
6	1	2	3	3	1	1	2	2
7	1	3	1	2	1	3	2	3
8	1	3	2	3	2	1	3	1
9	1	3	3	1	3	2	1	2
10	**2**	**1**	**1**	**3**	**3**	**2**	**2**	**1**
11	2	1	2	1	1	3	3	2
12	2	1	3	2	2	1	1	3
13	2	2	1	2	3	1	3	2
14	2	2	2	3	1	2	1	3
15	2	2	3	1	2	3	1	3
16	2	3	1	3	2	3	2	1
17	2	3	2	1	3	1	2	3
18	2	3	3	2	1	2	3	1

Příklad použití OATS [SPK+99]

- $L_{18}(2^1 \times 3^7)$
- Zobrazení (sloupce ... faktory):
  1. Hierarchie třídy A,
  2. Stavy hierarchie třídy A,
  3. Stavy třídy P,
  4. Hierarchie třídy C,
  5. Stavy hierarchie třídy C,
  6. poslední tři sloupce jsou ignorovány.
- 10: 2 1 1 3 3 2 2 1
- použij instanci třídy B ve stavu 1 k poslání zprávy s instancí třídy P ve stavu 1 instanci třídy E ve stavu 3.

Testování kompatibility

- Kompatibilita komponent, protokolů, databazí, atd.:



Příklad testování kompatibility - zakódování

1. **Akce uživatele:**

Domain value	Insert	Update
Array value	1	2

2. **Vstupní data:**

Domain value	dBank1	dBank2	dBank3
Array value	1	2	3

3. **Platformy:**

Domain value	Solaris 9 Sparc	Solaris 10 x86 zones	Linux - RH3
Array value	1	2	3

4. **Java ES:**

Domain value	JES2	JES3	JES4
Array value	1	2	3

5. **Formát přenášených dat:**

Domain value	HTML	XML	SOAP
Array value	1	2	3

6. **Modifikovaná komponenta JES5:**

Domain value	PS	AS	AM
Array value	1	2	3

7. **Protokol s databazí:**

Domain value	ODBC	JDBC	SQL
Array value	1	2	3

8. **Databáze:**

Domain value	BDB	Derby	HA-DB
Array value	1	2	3

Příklad testování kompatibility - testovací případ

- 1 faktor s 2 úrovněmi, 7 faktorů s 3 úrovněmi
- $L_{18}(2^1 \times 3^7)$
- Mapování (sloupce . . . faktory) dle prezentovaného pořadí
- **Abstraktní testovací případ 10:**

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

- **Testovací případ 10:**
  1. Uživatel koná akci "Update"
  2. s hodnotami dat v "dBank1"
  3. v prostředí platformy "Solaris 9 Sparc"
  4. se systémem "JES4"
  5. která komunikuje data ve formátu "SOAP"
  6. s obnovenou komponentou "AS"
  7. používající databazový protokol "JDBC"
  8. a databázi "BDB".

Latinský čtverec

- **Latinský čtverec** je matice s n řádky a n sloupci taková, že každé element obsahuje celé číslo od 1 do n tak, že se v žádném sloupci nebo řádku nevyskytuje žádné číslo vícekrát než jednou.

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \\ 3 & 1 & 2 \end{bmatrix}$$

- **Párově ortogonální Latinské čtverce**
  - Každý element v daném čtverci vystupuje v relaci s každým elementem druhého čtverce právě jedenkrát.
  - Pro jakékoli mocninu prvočísla N existuje $N-1$ párově ortogonálních Latinských čtverců velikosti $N \times N$.

$$\begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 & 0 \\ 2 & 3 & 4 & 0 & 1 \\ 3 & 4 & 0 & 1 & 2 \\ 4 & 0 & 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 0 & 1 \\ 4 & 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 4 & 0 \\ 3 & 4 & 0 & 1 & 2 \end{bmatrix} \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 3 & 4 & 0 & 1 & 2 \\ 1 & 2 & 3 & 4 & 0 \\ 4 & 0 & 1 & 2 & 3 \\ 2 & 3 & 4 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 4 & 0 & 1 & 2 & 3 \\ 3 & 4 & 0 & 1 & 2 \\ 2 & 3 & 4 & 0 & 1 \\ 1 & 2 & 3 & 4 & 0 \end{bmatrix}$$

Návrh testovacího plánu - Postup I - příklad

- 3 parametry,
- 5 významných hodnot A, B, C, D, E pro každý parametr,
- 3 párové ortogální Latinské čtverce 5×5:
  - pro každou pozici hodnoty ze všech čtverců
  - nefektivní

$$\begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 & 0 \\ 2 & 3 & 4 & 0 & 1 \\ 3 & 4 & 0 & 1 & 2 \\ 4 & 0 & 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 0 & 1 \\ 4 & 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 4 & 0 \\ 3 & 4 & 0 & 1 & 2 \end{bmatrix} \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 3 & 4 & 0 & 1 & 2 \\ 1 & 2 & 3 & 4 & 0 \\ 4 & 0 & 1 & 2 & 3 \\ 2 & 3 & 4 & 0 & 1 \end{bmatrix}$$

- **Testovací plán**:
  - 25 kombinací (místo 125)

$$\begin{array}{ccc} 0,0,0 & \longmapsto & A,A,A \\ 1,2,3 & \longmapsto & B,C,D \\ 2,4,1 & \longmapsto & C,E,B \\ \ldots & \ldots & \ldots \\ 3,2,1 & \longmapsto & D,C,B \end{array}$$

Návrh testovacího plánu - Postup II - příklad

- 4 parametry,
- 5 významných hodnot A, B, C, D, E pro každý parametr,
- 2 párové ortogální Latinské čtverce 5×5:
  - řádek, sloupec, pro každou pozici hodnoty ze všech čtverců
  - možné pokrýt až $n + 1$ parametrů/hodnot

$$\begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 & 0 \\ 2 & 3 & 4 & 0 & 1 \\ 3 & 4 & 0 & 1 & 2 \\ 4 & 0 & 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 0 & 1 \\ 4 & 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 4 & 0 \\ 3 & 4 & 0 & 1 & 2 \end{bmatrix}$$

- **Testovací plán**:
  - 25 kombinací (místo 125)

$$\begin{array}{lcl} 0,0,0,0 & \longmapsto & A,A,A,A \\ 0,1,1,1 & \longmapsto & A,B,B,B \\ 0,2,2,2 & \longmapsto & A,C,C,C \\ \ldots & \ldots & \ldots \\ 4,4,3,2 & \longmapsto & E,E,D,C \end{array}$$

Ortogonální Latinské čtverce

- Latinský čtverec velikosti N
- N je prvočíslo, $i \in [0, N-1]$, $j \in [0, N-1]$, $k \in [1, N-1]$:

$$A_k(i,j) = (ki + j) \mod N$$

- N je mocnina prvočísla:
  - konečné (Galoisovo) těleso řádu $N = p^n$
    - $x^N - x$ polynom je rozložen v $GF(p) = T_0^\star$

$$x^{p^n} - x = p_1(x) p_2(x) \cdots p_{k_0}(x)$$

    - Jestliže $k_0 < p^n$, konstrukce $T_1^\star = T_0^\star / p_i(x)$
    - $p_i(x)$ má řád větší nebo roven 2,
    - opakuj dokud není $k_0 = p^n$
  - konečná projektivní rovina řádu N

Konečná projektivní rovina

Definice:

- Konečné těleso $GF(N)$ řádu N
- $GF(N)$ má elementy $a_1, a_2, \cdots, a_n$
- Systém množin $(X, \mathcal{P})$:

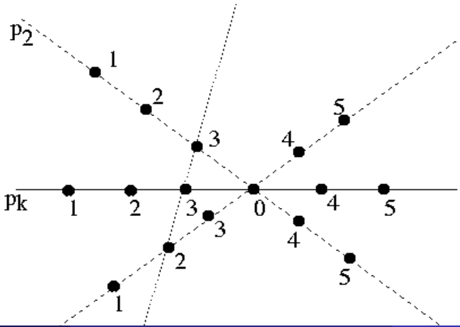
$$
\begin{aligned}
X &= \{(i,j); i,j = 1, \cdots, N\} \\
&\quad \cup \{(i,0); i = 1, \cdots, N\} \cup \{(0,0)\} \\
\mathcal{P} &= \{P(i,j); i,j = 1, \cdots, N\} \\
&\quad \cup \{P(0,j); j = 1, \cdots, N\} \\
P(i,j) &= \{(i,0)\} \cup \{(l,k); a_l = a_k a_i + a_j, k = 1, \cdots, N\} \\
P(0,j) &= \{(0,0)\} \cup \{(i,j); i = 1, \cdots, N\}
\end{aligned}
$$

Konečná projektivní rovina - vlastnosti

- soubor $N^2 + N + 1$ přímek $P(i, j) \in \mathcal{P}$
- soubor $N^2 + N + 1$ bodů $(i, j) \in X$
- každá přímka obsahuje $N + 1$ bodů,
- každý bod leží na $N + 1$ přímkách,
- kterékoliv dvě různé přímky se protínají v právě jednom bodu,
- jakékoliv dva různé body leží právě na jedné přímce

Konstrukce Latinských čtverců

1. Vyběr bod a $N+1$ přímek procházející tímto bodem.
2. Vyběr dvě přímky z těchto $N+1$ přímek:
   - index bodu na první přímce určuje řádkový index,
   - index bodu na druhé přímce určuje sloupcový index,
3. Každá přímka z $N-1$ zbývajících přímek určuje jeden Latinský čtverec.
   - Nalezni index bodu na přímce pro každý pár bodů na zvolených dvou přímkách.



Příklad testování kompatibility - kódování II

1. **Platformy:**

Domain value	Array value
Solaris 9 Sparc	0
Solaris 10 Sparc	1
Solaris 10 x86 zones	2
Linux - RH3	3

2. **Java ES:**

Domain value	JES2	JES3	JES4	JES5
Array value	0	1	2	3

3. **Obnovená komponenta JES5:**

Domain value	PS	AS	AM	DS
Array value	0	1	2	3

4. **Databáze:**

Domain value	BDB	Derby	HA-DB	$\star$
Array value	0	1	2	3

Příklad testování kompatibility - testovací případ II

- 4 parametry,
- 4 významné hodnoty pro každý parametr,
- 2 párově ortogonální latinské čtverce 4×4:

$$\begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & 0 & 3 & 2 \\ 2 & \mathbf{3} & 0 & 1 \\ 3 & 2 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 2 & 3 \\ 2 & 3 & 0 & 1 \\ 3 & \mathbf{2} & 1 & 0 \\ 1 & 0 & 3 & 2 \end{bmatrix}$$

- 16 kombinací (místo 256)
- **Abstraktní testovací případ 10:**
  - $2, 1, 3, 2$ (řádek, sloupec, hodnota1, hodnota2)
- **Testovací případ 10:**
  1. na platformě "Solaris 10 x86 zones"
  2. se systémem "JES3"
  3. s obnovenou komponentou "DS"
  4. a databází "HA-DB".

Literatura I

N. Logothetis and H. P. Wynn.
Quality through Design, Experimental Design, Off-line Quality Control and Taguchi's Contributions.
Clarendon Press, Oxford, 1989.

Douglas C. Montgomery.
Design and Analysis of Experiments.
John Wiley and Sons, third edition, 1991.

Phillip J. Ross.
Taguchi Techniques for Quality Engineering, Loss Function, Orthogonal Experiments, Parameter and Tolerance Design.
McGraw-Hill Book Company, 1988.

Hans Schaefer, Martin Pol, Tim Koomen, Gualtiero Bazzana, Lee Copeland, Hans Buwalda, Geoff Quentin, Mark Fewster, Lloyd Roden, Ruud Teumissen, and Erik Jansen.
Software testing training week.
SQE Europe, Tuesday 28 September to Friday 1 October 1999, Residence Fontaine Royale, Amstelween, Netherlands, Oct 1999.