Strukturované testování

Radek Mařík

CA CZ, s.r.o.

September 14, 2007

CA Labs

Transforming IT Management

Obsah

Testování bílé skříňky [KFN93]

Výhody:

Cílené testování: Tester může testovat program po částech.

Pokrytí: Lze určit, jakým testem je ta či ona část programu testována.

Řídicí tok: Tester ví, co by program měl provést v dalším kroku v závislosti na jeho současném stavu.

Integrita dat: Tester ví, které části programu modifikují (či by měla modifikovat) danou položku dat.

Vnitřní hranice: Tester může vidět vnitřní hranice kódu, které jsou úplně neviditelné z pohledu vnějšího testera.

Testování specifických algoritmů: Tester může využít speciální testovací techniky založena na dané implementaci.

Testování cest [KFN93]

- **Cesta** je sekvence operací, které se provedou od začátku běhu programu do jeho ukončení, tzv. úplná cesta
- **Část cesty** je sekvence operací z jednoho místa programu do jiného místa.
- **Kritéria pokrytí** specifikují třídu cest, které by se měly provést v rámci testování. Typicky redukují množství testů na rozumnou proveditelnou úroveň.
- Testování provedené podle těchto kritérií se nazývá **testování cest**.

Testování cest - příklad [KFN93]

```
IF (A>B and C==5)
  THEN do SOMETHING
SET D=5
```

```
(a) A<B  and C==5
       (SOMETHING is done, then D is set to 5)
(b) A<B  and C!=5
       (SOMETHING is not done,  D is set to 5)
(c) A>=B  and C==5
       (SOMETHING is not done, D is set to 5)
(d) A>=B  and C!=5
       (SOMETHING is not done, D is set to 5)
```

Kritéria pokrytí [KFN93]

Pokrytí řádek požaduje provedení každé řádky kódu alespoň jednou. Nejslabší kritérium.

- Tester může pokrýt všechny tři řádky kódu případem (a).

Pokrytí větví znamená, že podmínka každého větvení musí alespoň jednou být pravdivá a alespoň jednou nepravdivá. Toto pokrytí vyžaduje otestování všech řádek a všech větví.

- Tester může použít případ (a) a jakýkoliv další ze zbývajících tří.

Pokrytí podmínek zkontroluje všechny možné způsoby, za kterých daná podmínka je pravdivá či nepravdivá.

- Vyžaduje všechny čtyři případy.
- Požadováno americkou armádou a letectvím.

Úplné pokrytí cest vyžaduje provedení všech možných různých úplných cest.

- V praxi je neproveditelné.

Nástroje pokrytí [Kit95]

- Analyzátory pokrytí poskytují kvantitavní míru kvality testů.
- Nástroj produkuje zprávu o tom, které části produktu byly pokryty (provedeny) v rámci dané sady testů.
- Generace:
  1. předzpracování zdrojového kódu,
  2. instrumentace strojového kódu (object code instrumenting = OCI)
- Nástroje:
  - PureCoverage

Grafy a relace [Bei95]

- Grafy jsou hlavním koncepčním nástrojem testování:
  - grafy toku řízení,
  - grafy toku dat,
  - stromy závislosti volání funkcí,
  - grafy konečných automatů,
  - grafy toku transakcí.
- **Relace:** vybraná asociace mezi objekty,
  - A, B . . . objekty,
  - $\leftrightarrow$, $\leftarrow$, $\rightarrow$ . . . relace,

$$A \rightarrow B$$

  - Příklad: akce A je následovaná akcí B.
- **Uzel:** objekty grafu reprezentované kroužky,
- **Jméno uzlu:** každý uzel má jednoznačnou identitu nebo jméno,
- **Atributy uzlu:** vlastnosti uzlu.
  - Příklady:
    - stav programu,
    - hodnota proměnné.

Hrany grafu [Bei95]

- **Hrana:** šipka nebo čára spojující dva uzly vyjadřující danou relaci mezi těmito uzly.
- **Jméno hrany:** každá hrana má jednoznačnou identitu nebo jméno.
  - pár jmen obou uzlů,
  - speciální jméno,
- **Atributy hran:** vlastnosti hran.
  - Příklady:
    - exekuční čas programu podél specifické cesty,
    - pravděpodobnost provedení dané cesty,
    - fakt o výběru určitého datového objektu.
- **Orientovaná hrana:** používaná pro asymetrické relace.
  - Příklad: A je následováno B.
  - Většina modelů testování používá orientované hrany.
- **Neorientovaná hrana:** hrana označující symetrickou relaci.
- **Paralelní hrany:** dvě či více hran mezi jedním párem uzlů.

Terminologie teorie grafu [Bei95]

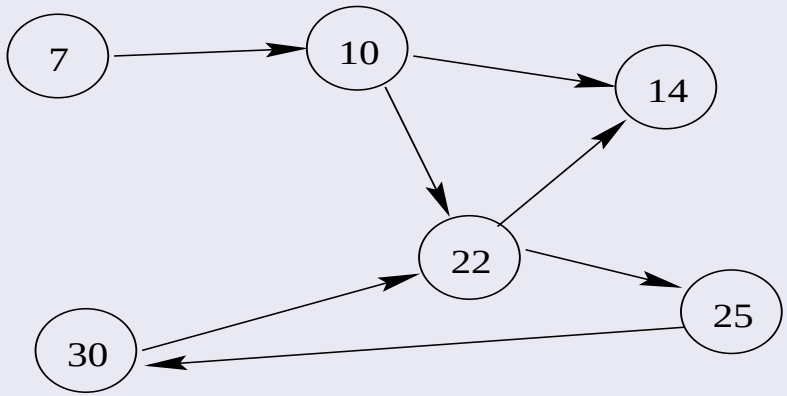
- **Graf:** je kolekce uzlů, jmen uzlů, atributů uzlů, hran, jmen hran, atributů hran a relací mezi uzly.
- **Orientovaný graf:** všechny hrany jsou orientované.
- **Neorientovaný graf:** všechny hrany jsou neorientované.
- **Vstupní hrana:** hrana, která míří do uzlu (hlava šipky).
- **Výstupní hrana:** hrana opouštějící uzel (ocas šipky).
- **Počáteční uzel:** uzel nemající vstupní hrany.
- **Koncový uzel:** uzel nemající výstupní hrany.
- **Uzel větvění:** uzel se dvěma a více výstupními hranami.
  - Příklad: CASE příkaz nebo IF-THEN-ELSE příkaz.
- **Cesta:** sekvence hran spojující dva uzly.
  - Příklad: Při testování chování se pracuje s cestami *modelem*, který popisuje chování softwaru. Takové cesty mohou nebo nemusí korespondovat cestám implementací programu.

Cesty grafu [Bei95]

- **Proveditelná cesta:** cesta, pro níž existují vstupní hodnoty takové, že běh programu bude sledovat danou cestu.
- **Neproveditelná cesta:** cesta, pro níž neexistují množina vstupních hodnot, která by umožňila postupovat programem podél této cesty.
- **Cesta start-stop:** cesta vedoucí od počátečního uzlu do koncového uzlu.
- **Segment cesty:** obvykle cesta, která není cestou start-stop.
- **Jméno cesty:**
  - jména uzlů podél cesty,
  - jména hran podél cesty,
- **Délka cesty:**
  - počet uzlů dané cesty,
  - počet hran dané cesty.
- **Smyčka:** jakákoliv cesta, která navštíví nějaký uzel alespoň dvakrát.
- **Cesta bez smyček:** cesta nemající smyčku.

Reprezentace grafů [Bei95]

Graf



Seznam

7:	10
10:	14, 22
14:	
22:	14, 25
25:	30
30:	22

Matice (tabulka)

	7	10	14	22	25	30
7	.	1	.	.	.	.
10	.	.	1	1	.	.
14	.	.	.	.	.	.
22	.	.	1	.	1	.
25	.	.	.	.	.	1
30	.	.	.	1	.	.

Obecné principy testování [Bei95]

Výstavba modelu (grafu)

1. objekty, o které se zajímáme (uzly).
2. relace, které by měly existovat mezi uzly.
3. které objekty mají vztah s jinými (hrany).
4. vlastnosti asociované s hranami: atributy hran.

Návrh testovacích případů podle modelu

1. definuj graf,
2. definuj relace,
3. navrhni testy pro **pokrytí uzlů** (testy potvrzující přítomnost uzlů).
4. navrhni testy pro **pokrytí hran** (testy potvrzující všechny požadované hrany a žádné jiné).
5. otestuj všechny atributy.
6. navrhni testy smyček.

Model toku transakcí [Bei95]

- **objekty:** kroky zpracování dané transakce. Pro každy krok transakčního procesu existuje jeden uzel.
  - Příklad: kontrola účtu nebo získání peněz z bankovního automatu.
- **relace:** "*je následován* - dalším krokem".
  - Příklad: **kalkulace_příjmu** *je následována* **kalkulací_daně**
- **hrany:** spojují po sobě následující kroky.
  - Příklad: relace *je následován* mezi dvěma uzly.

Modely konečných automatů [Bei95]

Příklad: **konečný automat položek menu**

- **objekty:** Menu, které se objevují na obrazovce. Pro každé menu existuje jeden uzel.
- **relace:** "*Vede přímo na.*". Výběr položky menu způsobí přechod programu na zobrazení dalšího menu.
- **hrany:** pro každý výběr polozky existuje jedna hrana mezi příslušnými uzly.

Model datového toku [Bei95]

- **objekty:** vybrané datové objekty. Pro každý datový objekt a jeho instanci (různé hodnoty) existuje jeden uzel.
- **relace:** "*je použit pro výpočet hodnoty čeho*".
  - Příklad: ve vzorci $X = 2Y + Z$, Y a Z jsou použity pro výpočet hodnoty X.
- **hrany:** nakresli hranu z A do B, pokud hodnota A je použita pro výpočet hodnoty B.

Časové modely [Bei95]

- **objekty:** sekvence příkazů programu bez větvění (přímé sekvence).
- **relace:** "*je následován příkazem*".
  - Příklad: **READ filename** *je následován* **FOR**.
- **hrany:** spojují příkazy, které následují po sobě. **Příkazy větvění**, (např. **IF, CASE**) mají jednu výstupní hranu pro každou větev.
- **vlastnosti:**
  - očekávaná čas provedení hrany,
  - pravděpodobnost, že program provede danou větev.

Predikáty řízení toku [Bei95]

Příklady: výpočet daně z příjmu fyzických osob (smíchané USA a ČR formuláře)

- **logický predikát:** věta nebo výraz, který nabývá logické hodnoty TRUE nebo FALSE.
  - Příklad: Váš rodič Vás může prohlásit za vyživované dítě na přiznání k dani z příjmů fyzických osob.
- **výběrový predikát:** výraz, který nabývá více než dvou hodnot určených k výběru jedné z mnoha variant.
  - Příklad: Zaškrtni pouze políčko - (1) svobodný, (2) ženatý vyplňující společné přiznání, (3) ženatý vyplňující oddělené přiznání, (4) hlava domácnosti, (5) vdovec.
- **složený predikát:** logický výraz zahrnující více logických predikátů spojených spojkami **AND, OR,** nebo **NOT**.

Závislost řídicích predikátů [Bei95]

- **nezávislé predikáty:** dva nebo více predikátů podél cesty jsou nezávislé, pokud jejich pravdivostní hodnoty mohou být nastaveny nezávisle.
- **korelované predikáty:** dva nebo více predikátů podél cesty jsou korelované, pokud pravdivostní hodnota jednoho z nich omezuje pravdivostní hodnoty ostatních predikátů podél cesty.
  - Příklad: daný predikát se objeví dvakrát podél cesty.
- **komplementární segmenty cest:** dva segmenty, pokud ty samé predikáty nacházející se na těchto segmentech mají na jednom segmentu hodnotu TRUE a na druhém FALSE.

Graf toku řízení [Bei95]

- **objekty (uzly):** sekvence kroku zpracování taková, že pokud se zpracuje jeden krok sekvence, budou provedeny i všechny zbývající kroky této sekvence (předpoklad nepřítomnosti pochybení)
  - Příklad: Tax Form 1040 (USA)

    7: (enter) příjmy, výplaty, spropitné.
    8a: (enter) daněné úroky.
    9: (enter) příjmy z dividend
- **relace (hrany):** *je přímo následován čím*
  - Příklad: Tax Form 1040

    7: 8a
    8a: 9

Predikátový uzel [Bei95]

Definice

Uzel s dvěma či více výstupními hranami, každá hrana nesoucí atribut s hodnotou predikátu.

Příklad

Tax Form 1040: Jestliže Vás Váš rodič může prohlásit za vyživované dítě, pak zaškrtněte políčko 33b, jinak políčko 33b nezaškrtávejte.

33b	Váš rodič Vás může prohlásit za závislého (predikátový uzel)
33b'	zaškrtni políčko 33b (procesní uzel)
33b:	33b'(T), 34(F)
33b':	34

Další speciální uzle [Bei95]

- **výběrový uzel:** uzel s více než dvěma výstupními hranami každou ohodnocenou hodnotou výběru.
- **spojovací uzel:** uzel s dvěma a více vstupními hranami.
- Příklad (řádka 34, formulář 1040): Zadej větší z následujících hodnot:
  - buď srážku daně z Rozvahy A řádky 29
  - NEBO standardní srážku podle rodinného stavu.

  Avšak jestliže jste zaškrtl jedno z políček na řádce 33a nebo b, pokračujte v určení standardní srážky podle instrukcí. Jestliže jste zaškrtl políčko 33c, Vaše standardní srážka je nula.
  1. svobodný = $3,800,
  2. ženatý/vdaná vyplňující společně = $6,350,
  3. vdovec =$6,350,
  4. ženatý/vdaná vyplňující odděleně = $3,175,
  5. hlava domácnosti = $5,600.

Příklad výběrového a spojovacího uzlu [Bei95]

34	Standardní nebo položkovaná srážka?
34.1	Rozvaha A, řádka 29
34.2	zaškrtni 33a nebo 33b
34.4	zaškrtnuto 33c?
34.5	vyběr variantu (výběrový uzel)
34.8	srážka = \$6,350 (spojkový uzel)
34:	34.1(položky), 34.2(standard)
34.2:	34.3(T), 34.4(F)
34.4:	34.5(F), 34.6(T)
34.5:	34.7(svobodný), 34.8(společně, vdovec), ...

Složené predikáty [Bei95]

- Složené predikáty jsou ošidné, neboť skrývají komplexitu.
- Je správné je testovat, protože v nich programátoři dělají často chyby.
- Vždy lze složený predikát expandovat, aby se odhlalila složitost.
- n pomocných predikátoru vede na 2^n větví.
- Musí se uvažovat všech 2^n případů.
- Případy nejsou redudantní, protože složený predikát bude pracovat jenom tehdy, pokud v implementace neobsahuje chybu.
- Příklad: $A\&B$ OR C

A:	B.1(T), B.2(F)
B.1:	C.1(T), C.2(F)
B.2:	C.3(T), C.4(F)
C.1:	TRUE(T,F)
C.2:	TRUE(T), FALSE(F)
C.3:	TRUE(T), FALSE(F)
C.4:	TRUE(T), FALSE(F)

Návrh testu a jeho provedení [Bei95]

1. Pověř požadavky, zkontroluj jejich úplnost a konzistenci.
2. Přepiš specifikace do sekvence krátkých vět. Pozornost věnujte predikátům.
3. Jména uzlů: jendoznačně očísluj věty.
4. Vystav model.
5. Ověř model.
6. Vyber testovací cesty.
7. Dourči vybrané cesty, tj. urči vstupní hodnoty takové, že program poběží danou cestou pokud neobsahuje chyby.
8. Odhadni a zaznamenej očekávané výsledky pro každý test.
9. Definuj validační kritéria pro každý test.
10. Proveď testy.
11. Zkontroluj výsledky.
12. Zkontroluj cestu.

Ověření testovacích cest [Bei95]

Ověření, zda

- všechny objekty mají očekávané relace s jinými objekty.
- uzly vykazují předpokládané hodnoty atributů.
- hrany jsou tam, kde se očekávají v rámci daných relací.
- hrany mají správné hodnoty atributů.
- všechny orientované hrany mají očekávaný směr.
- všechny symetrické relace jsou vskutku symetrické.
- hrany jsou orientované, pokud tak mají být.

Výběr cest testů [Bei95]

- Cesta se staví segment po segmentu počínaje počátečním uzlem a konče koncovým uzlem.
- Vyber segmenty, které začínají v bodech větvení a pokračuj to místa, kde se řízení opět spojí v jednom uzlu.
- Vystav cesty jako kombinace takto nalezených segmentů.

Příklad výběru cest testů [Bei95]

Model

32:	vstup celkový_příjem	33a1
33a1:	nastav čítač na nulu	33a2
33a2:	Má 65 nebo je starší?	33a3(T), 33a4(F)
33a3:	inkrementuj čítač	33a4
33a4:	Je slepý?	33a5(T), 33a6(F)
33a5:	inkrementuj čítač	33a6
33a6:	Má partner 65 nebo je starší	33a7(T), 33a8(F)

Testy

A1:	32, 33a1, 33a2, 33a3(T), 33a4
A2:	32, 33a1, 33a2, 33a4(F)
A1:	32, 33a1, 33a2, 33a3(T), 33a4, 33a5(T), 33a6, ...
A2:	32, 33a1, 33a2, 33a4(F), 33a6(F), ...

33a4 je nezávislé na 33a2

Kombinace segmentů testů [Bei95]

- Odstraň kombinace, které nelze provést vzhledem k vzájemně se blokujícím predikátům.
- Pokud je možné si vybrat, vyber kombinace na základě chyb v minulosti, obtížnosti analýzy kombinací, rozsahu analýzy situace, Vaší znalosti odpovědného programátora, ...

Příklad kombinace segmentů testů [Bei95]

Model

33b:	Vyživuje Vás rodič?	33b1(T), 33c(F)
33b1:	zaškrtni 33b	33c
33c:	ženatý/vdaná vyplňují společně	33c1

Testy

B1:	33b, 33b1(T), 33c
B2:	33b, 33c(F)
A1B1:	32, 33a1, 33a2, 33a3(T), 33a4, 33a5(T), 33a6, ..., 33b, 33b1(T), 33c
A2B2:	32, 33a1, 33a2, 33a4(F), 33a6(F), ..., 33b, 33c(F)
A1B2:	32, 33a1, 33a2, 33a3(T), 33a4, 33a5(T), 33a6, ..., 33b, 33c(F)
A2B1:	32, 33a1, 33a2, 33a4(F), 33a6(F), ..., 33b, 33b1(T), 33c

Dourčení hodnot [Bei95]

- Nalezení vstupních hodnot tak, aby za předpokladu bezchybné implementace program provedl danou cestu podle modelu.
- Obvykle lze ze znalosti funkce aplikace vybrat množinu cest pokrytí a určit přímo potřebné hodnoty.

Dourčení logiky řízení I [Bei95]

1. Rozděl model na segmenty, které začínají a končí jedním uzlem.
2. Vytvoř seznam závislých (korelovaných) predikátů.
3. Pro každý segment vytvoř seznam všech možných podcest tohoto segmentu, které neobsahují smyčky.
4. Spoj jakékoliv dva segmenty sdílející silně závislé predikáty, i když nejsou přímo napojeny.

Dourčení logiky řízení II [Bei95]

1. Eliminuj nedosažitelné odporující si cesty.
2. Pokračuj v kroku 1 spojováním segmentů a eliminováním kombinací.
3. Vznikne množina větších segmentů, které jsou vzájemně nezávislé. Vyber dostatek cest tak, aby všechny segmenty byly pokryté. Napojovaní těchto segmentů je zvolit libovolně.
4. Spoj segmenty kombinací cest a dourči hodnoty kombinací.
5. Opakuj pokud pokrytí nevyhojue zvolenému kritériu.
6. Dourčení vstupních hodnot by nyní mělo být přímočaré. Stačí sledovat cestu.

Dourčení algebraické [Bei95]

1. Interpretace predikátu: predikát je intepretován dosazením výrazů vstupních hodnot.
2. Interpretace predikatu závisí na dané cestě.
3. Je ekvivalentní predikátu, který získáme postupným výpočtem podél cestu k daném predikátu.
4. Vyjádřete predikátový výraz pomocí symbolické substituce.
5. Nalezněte množinu vstupních hodnot, která vyhovuje všem interpretovaným predikátům podél vybrané cesty (soustava rovnic).
6. Obvykle však lze vystačit s postupným výběrem a dosazovaním hodnot podvýrazů predikátů.

Predikce výsledků [Bei95]

- predikce očekávaných výsledků pro každou vybranou cestu.
- Nedělejte to manuálně!
- Použijte:
  - existující testy.
  - starý program.
  - předešlou verzi.
  - prototypy.
  - modelovací program (tabulkové výpočty, BASIC)
  - vynucené jednoduché případy (Realismus je potíž bránící dobrému testování).
  - aktualní program (verifikován použitím mezivýsledků).

Aplikační možnosti [Bei95]

- základní technika,
- Nenaleznete chybějící požadavky.
- Nenaleznete sporné nebo nevyžádané vlastnosti, které byly implemtovány, aniž byly specifikovány požadavky.
- Je velmi nepravděpodobné, že budou nalezeny chybějící cesty a vlastnosti, pokud program a model pro testy byly vytvořeny jednou a tutéž osobou.
- Náhodná správnost, i když nepraděpodobná, může mařit výsledky této metody, pokud není možné zkontrolovat všechny mezivýsledky a hodnoty predikátů.
- Vaše testy nebudou lepší než je Váš orakulus.

Literatura I

Boris Beizer.

Black-Box Testing, Techniques for Functional Testing of Software and Systems.
John Wiley & Sons, Inc., New York, 1995.

Cem Kaner, Jack Falk, and Hung Quoc Nguyen.

Testing Computer Software.
International Thomson Computer Press, second edition, 1993.

Edward Kit.

Software Testing in the Real World.
Addison-Wesley, 1995.