

Szoftverellenőrzési technikák (vimim148)

Struktúra alapú tesztervezési módszerek

Majzik István, Micskei Zoltán

<http://www.inf.mit.bme.hu/>

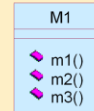
1

Utolsó módosítás: 2010.10.18.

Teszttervezés módszerei

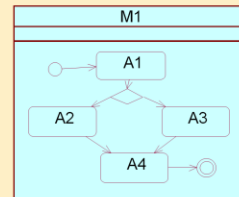
I. Specifikáció alapú

- a rendszer mint „fekete doboz” adott
- csak a külső viselkedés (funkció) ismert, a belső felépítés (pl. forráskód) nem
- tesztelés alapja: **specifikált funkciók léte**; extra funkciók hiánya



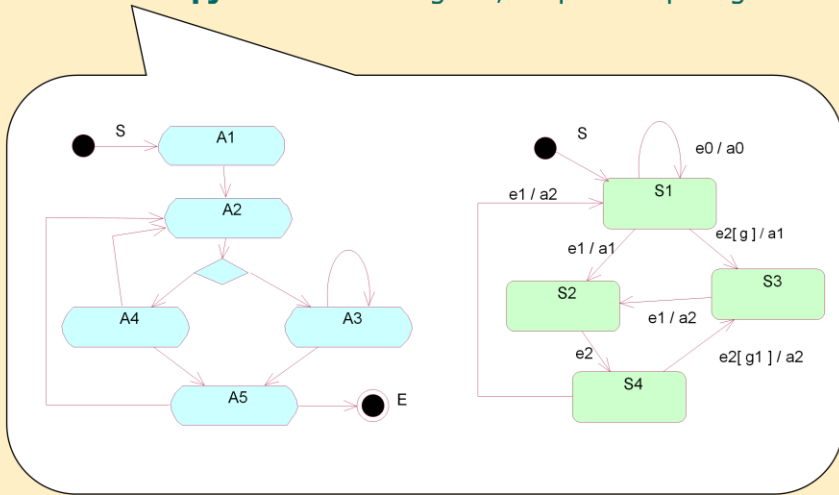
II. Struktúra alapú

- a rendszer mint „üvegdoboz” adott
- a belső struktúra is ismert
- tesztelés alapja: **belső működés**: programgráf bejárása



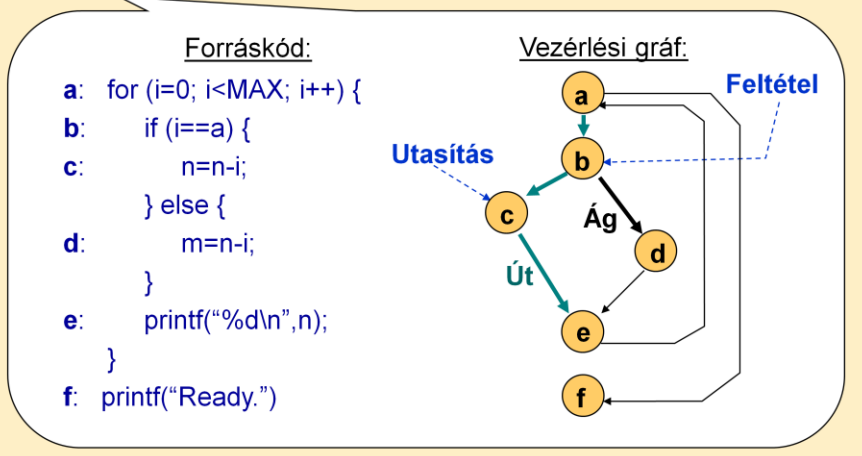
A belső struktúra

- Jól kezelhető modell:
 - **Modell alapján:** aktivitás diagram, állapottérkép diagram



A belső struktúra

- Jól kezelhető modell:
 - Modell alapján: aktivitás diagram, állapottérkép diagram
 - **Forráskód alapján:** vezérlési gráf (programgráf)



Tesztminőségi mértékszámok

A tesztelés hatékonyságának számszerű jellemzése:

A tesztelhető elemek mekkora részét teszteltük

- | | |
|-------------------------|------------------------|
| 1. Utasítások | → Utasítás fedettség |
| 2. Döntési ágak | → Döntési ág fedettség |
| 3. Feltétel kombinációk | → Feltétel fedettség |
| 4. Végrehajtási utak | → Út fedettség |

Ez **nem** a hibafedés!

Szabványok előírása lehet (DO-178B, MSZ EN 50128,...)

- 100% utasítás fedettség általában alapkövetelmény

Tartalom

- Vezérlési folyamat alapú kritériumok
- Adatfolyam alapú kritériumok
- Összefoglalás

Alapfogalmak

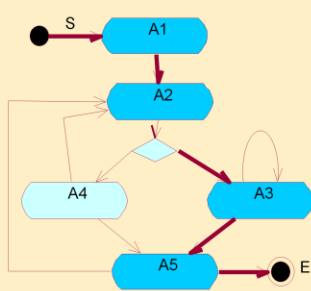
- Utasítás (statement)
- Blokk (block)
 - egybefüggő utasítás, amik között nincs elágazás vagy függvényhívás
- Feltétel (condition)
 - egyszerű logikai kifejezés, amiben nincs Boolean operátor
- Döntés (decision)
 - Feltételekből álló kifejezés nulla vagy több Boolean operátorral
- Út (path)
 - utasítások sorozata, tipikusan a modul be és kilépési pontja között

1. Utasítás lefedettség (Statement coverage)

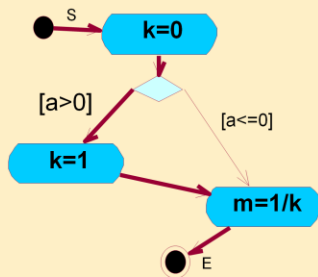
Definíció:

$$\frac{\text{Tesztelés során végrehajtott utasítások száma}}{\text{Összes utasítás száma}}$$

Utasítások kihagyási feltételeit nem veszi figyelembe!



Utasítás fedettség: 80%



Utasítás fedettség: 100%

- Előkerül még a line coverage elnevezés is, ami nem szerencsés, mert egy sorban lehet több utasítás is.

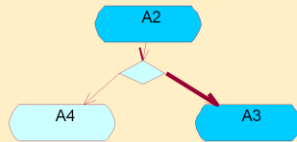
2. Döntés lefedettség (Decision coverage)

Definíció:

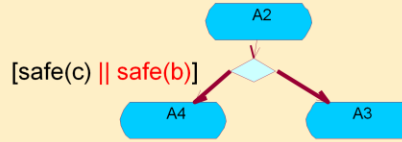
Tesztelés során végrehajtott döntési ágak száma

Összes lehetséges döntési ág száma

Nem vesz figyelembe minden feltétel-kombinációt!



Döntési ág fedettség: 50%



Döntési ág fedettség: 100%

-Használják még a branch coverage elnevezést is, azonban az például repülő környezetben nem ugyanaz mint a decision coverage, lásd:

FAA, Certification Authorities Software Team, „What is a "Decision" in Application of Modified Condition/Decision Coverage (MC/DC) and Decision Coverage (DC)?”, 2002.
URL:

http://www.faa.gov/aircraft/air_cert/design_approvals/air_software/cast/cast_papers/media/cast-10.pdf

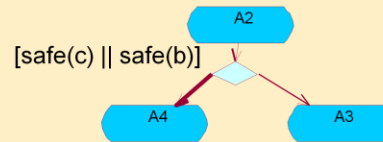
3. Feltétel lefedettség (Condition coverage)

Definíció:

Feltételekben a tesztelt bemeneti kombinációk száma
Feltételek összes bemeneti kombinációinak száma

100%-os feltétel lefedettséghez :

1. $\text{safe}(c) = \text{true}, \text{safe}(b) = \text{false}$
2. $\text{safe}(c) = \text{false}, \text{safe}(b) = \text{true}$



- Nem feltétlen eredményez 100%-os döntés lefedettséget!
- *Más definíció:* minden feltételt igaznak és hamisnak is kiértékelünk (ez nem ugyanaz, mint a fenti a lusta kiértékelés miatt!)

Példa arra, hogy 100%-os feltétel lefedettség nem eredményez 100%-os döntés lefedettséget:

- Döntés: $((a \text{ AND } b) \text{ OR } (c \text{ AND } d))$

- T1: $a=1, b=0, c=1, d=0$

- T2: $a=0, b=1, c=0, d=1$

- Minden feltétel (a,b,c,d) szerepel igaz és hamis értékkel is, de a döntés mindkét esetben hamisra értékelődik ki, így a döntés lefedettség csak 50%-os

Forrás: Y. T. Yu and M. F. Lau, "A comparison of MC/DC, MUMCUT and several other coverage criteria for logical decisions," *Journal of Systems and Software*, vol. 79, no. 5, pp. 577-590, May. 2006.

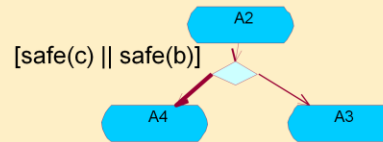
4. Condition/Decision Coverage (C/DC)

Definíció:

- Minden feltétel felvette az összes lehetséges kimenetét legalább egyszer, és
- minden döntés felvette az összes lehetséges kimenetét egyszer

100%-os C/DC lefedettséghez :

1. $\text{safe}(c) = \text{true}, \text{safe}(b) = \text{true}$
2. $\text{safe}(c) = \text{false}, \text{safe}(b) = \text{false}$



Nem vizsgálja meg, hogy a feltételnek tényleg van-e hatása a döntés eredményére!

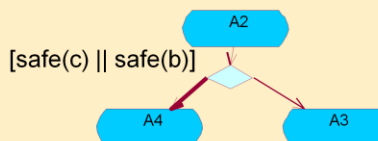
5. Modified Condition/Decision Coverage (MC/DC)

Definíció:

- Minden feltétel felvette az összes lehetséges kimenetét legalább egyszer, és
- minden döntés felvette az összes lehetséges kimenetét egyszer, és
- minden feltétel a többitől függetlenül befolyásolta a hozzá tartozó döntés kimenetelét

100%-os MC/DC lefedettséghez :

1. $\text{safe}(c) = \text{true}, \text{safe}(b) = \text{false}$
2. $\text{safe}(c) = \text{false}, \text{safe}(b) = \text{true}$
3. $\text{safe}(c) = \text{false}, \text{safe}(b) = \text{false}$



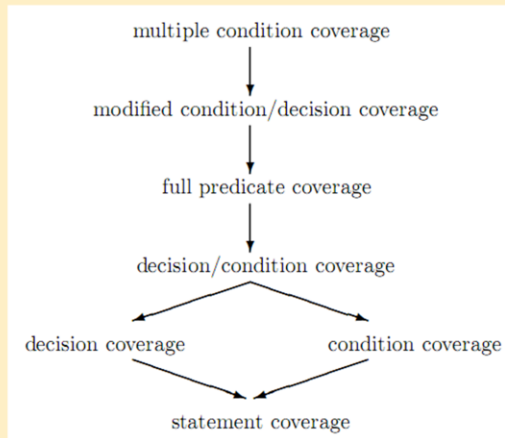
Feltétel és döntési lefedettségek összefoglalása

Table 1. Types of Structural Coverage

| Coverage Criteria | Statement Coverage | Decision Coverage | Condition Coverage | Condition/Decision Coverage | MC/DC | Multiple Condition Coverage |
|--|--------------------|-------------------|--------------------|-----------------------------|-------|-----------------------------|
| Every point of entry and exit in the program has been invoked at least once | | • | • | • | • | • |
| Every statement in the program has been invoked at least once | • | | | | | |
| Every decision in the program has taken all possible outcomes at least once | | • | | • | • | • |
| Every condition in a decision in the program has taken all possible outcomes at least once | | | • | • | • | • |
| Every condition in a decision has been shown to independently affect that decision's outcome | | | | | • | • ⁸ |
| Every combination of condition outcomes within a decision has been invoked at least once | | | | | | • |

Forrás: Kelly J. Hayhurst et al. „A Practical Tutorial on Modified Condition/Decision Coverage”, NASA/TM-2001-210876, 2001

Feltétel és döntési lefedettségek összefoglalása



Vezérlési folyamat
alapú kritériumok
viszonya

Forrás: S. A. Vilkomir and J. P. Bowen, "From MC/DC to RC/DC: formalization and analysis of control-flow testing criteria," *Formal Aspects of Computing*, vol. 18, no. 1, pp. 42-62, 2006.

Példa: vezérlési folyamat alapú kritériumok

```
Product getProduct(String name, Category cat){
    if (name == null || ! cat.isValid)
        throw new IllegalArgumentException();

    Product p = ProductCache.getItem(name);

    if (p == null){
        p = DAL.getProduct(name, cat);
    }

    return p;
}
```

15

- Statement coverage:
 - name: null, cat: valid
 - name: valid, cat valid, product not in cache
 - Problem: getting from cache not tested
- Decision:
 - name: null, cat: valid
 - name: valid, cat valid, product not in cache
 - name: valid, cat valid, product in cache
 - Problem: cat.isValid call is not tested
- C/DC coverage:
 - name: null, cat: valid
 - name: valid, cat: invalid
 - name: valid. cat valid, product not in cache
 - name: valid. cat valid, product in cache

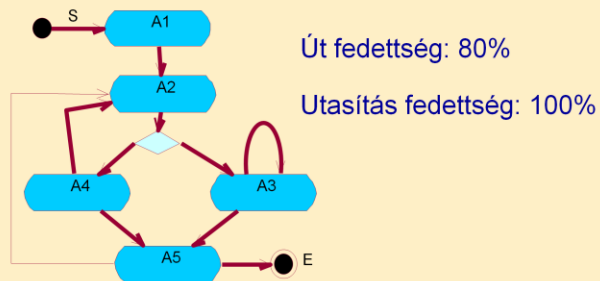
6. Alap út lefedettség (Basis path coverage)

Definíció:

$$\frac{\text{Tesztelés során bejárt független utak száma}}{\text{Összes független út száma}}$$

100% út lefedettség együtt jár:

- 100% utasítás lefedettség, 100% döntés lefedettség
- feltétel lefedettség nem garantált



A teljes út lefedettség (full path coverage) a ciklusok miatt általában nem praktikus.

Egy strukturális tesztelési módszer

- Tipikus cél: **Független utak bejárása**
 - Független utak a tesztelés szempontjából:
Van olyan utasítás vagy elágazás,
ami a másokban nincs meg
- A független utak maximális száma:
 - CK , ciklomatikus komplexitás
 - Szabályos vezérlési gráf alapján meghatározható:
 $CK(G)=E-N+2$, ahol
 - E : élek száma
 - N : csomópontok száma a G vezérlési gráfban
(vezérlési gráf összekötött, 1-1 kimenet és bemenet)
- A független utak halmaza nem egyedi

Egy strukturális tesztelési módszer

- Tipikus cél: Fü

- Független utak
- Van olyan utasítás ami a másikban

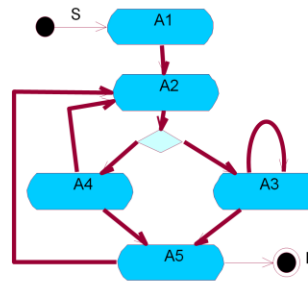
- A független ut

- CK, ciklomat
- Szabályos vezér
- $CK(G) = E - N + 2$, ahol

E: élek száma

N: csomópontok száma a G vezérlési gráfban
(vezérlési gráf összekötött, 1-1 kimenet és bemenet)

- A független utak halmaza nem egyedi



N=5,
E=8,
CK=5
max. 5
független út!

Strukturális tesztek generálása

- Algoritmus:
 - Max. CK számú független út kiválasztása
 - Bemenetek generálása egy-egy út bejárásához
- Problémák:
 - Nem minden út bejárható (ld. feltételek).
 - Generálható-e az úthoz bemeneti szekvencia?
 - Lehetséges-e a belső változók beállítása?
 - Ciklusok: Korlátozni (minimalizálni) kell a bejárást!
- Teljesen automatikus módszerek nem léteznek

Tartalom

- Vezérlési folyamat alapú kritériumok
- **Adatfolyam alapú kritériumok**
- Összefoglalás

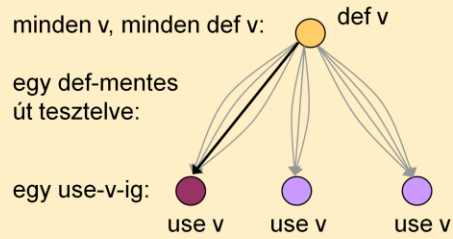
Alapfogalmak

- Változók értékadásának és felhasználásának viszonyát vizsgáljuk
- **def(v)**: adott helyen a v változónak értéket adunk
- **use(v)**: adott helyen a v változót felhasználjuk
 - p-use(v): v értékét feltételben használjuk
 - c-use(v): v értékét számolásban használjuk
- **definíciómentes út (definition clear path)**: v-nek nem adunk értéket az út csomópontjaiban

Adatfolyam alapú tesztelési kritériumok (kitekintés)

- All-defs:

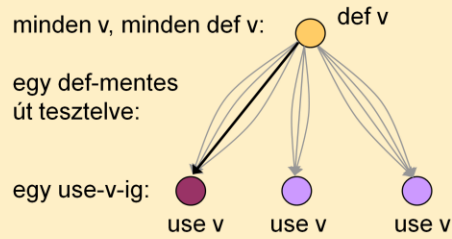
- def v
- use v



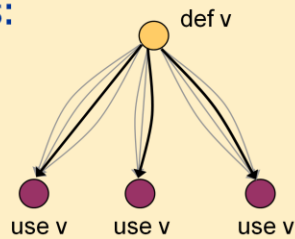
Adatfolyam alapú tesztelési kritériumok (kitekintés)

- All-defs:

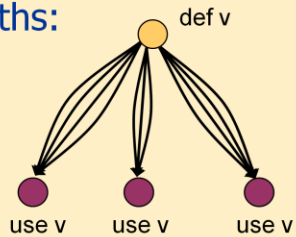
- def v
- use v



- All-uses:



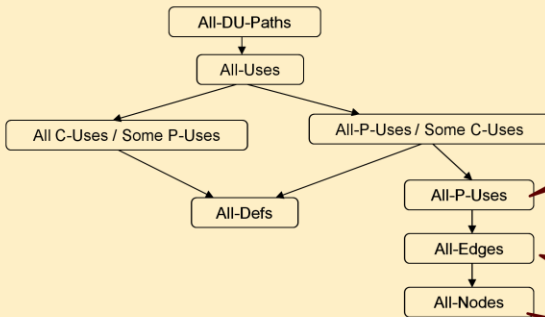
- All-paths:



Tartalom

- Vezérlési folyam alapú kritériumok
- Adatfolyam alapú kritériumok
- **Összefoglalás**

Strukturális fedettség kritériumok a gyakorlatban



Bonyolultabb feltétel és döntés fedés a biztonságkritikus szabványokban

Ezt már csak a jobb eszközök tudják

A projektek nagy része ezt sem éri el, 90%-os lefedettség már egy jó cél

ÖSSZEFOGLALÁS: Teszttervezési módszerek

- Specifikáció és struktúra alapú módszerek
 - Sok technika
 - Mindegyik alkalmazásához gyakorlat kell
- Nagy szakadék az elméleti és a gyakorlatban alkalmazott módszerek között
- Technikák kombinációja hatásos általában:
 - (példa mérések egy MS tanulmányból)
 - specifikáció alapú: 83%-os kódfedés
 - + exploratory: 86%-os kódfedés
 - + strukturális: 91%-os kódfedés