BUDAPESTI MŰSZAKI ÉS GAZDASÁGTUDOMÁNYI EGYETEM
TÁVKÖZLÉSI ÉS TELEMATIKAI TANSZÉK

SZELEKTÍV AUTOMATIKUS TESZTGENERÁLÁS EFSM FORMÁLIS MODELLEKHEZ HIBA ÉS STRING SZERKESZTÉSI TÁVOLSÁG ALAPÚ MÓDSZEREKKEL

Kovács Gábor

Ph.D. Tézisfüzet

Tudományos vezető

Dr. Csopaki Gyula és Dr. Tarnay Katalin
Nagysebességű Hálózatok Laboratóriuma
Távközlési és Médiainformatikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Budapest
2009

1. Bevezető

A távközlésben használt protokollok, amelyek két vagy több eszköz, berendezés közötti kommunikációt adják meg, akárcsak napjaink szoftverrendszerei, egyre összetettebbé válnak, ugyanakkor velük szemben szabványok szigorú megbízhatósági követelményeket támasztanak. Számos módszertant dolgoztak ki távközlési protokollok fejlesztésére, amelyek közül néhány a szabványosításig is eljutott. Ezek a módszertanok képesek arra, hogy garantálják a protokollok minőségét a teljes életciklus folyamán.

Az SDL+ életciklus modellben [IT97] a követelményfelvételt a kétlépcsős tervezési fázis követi, amely egy vázlatos tervezésből és egy formális modellezésből áll. Ez utóbbi validáció után alapjául szolgál az implementáció és a teszt specifikáció tevékenységre.

Az értekezés ezt az életciklus modellt alapul véve vizsgálja meg a formális modellen alapuló teszt specifikáció automatizálhatóságának kérdéskörét építve a szakirodalom által javasolt megoldásokra. Kiterjesztett véges állapotgépekhez (EFSM) történő modell alapú teszteset generálásra több megközelítés is létezik. Az EFSM véges állapotgéppé alakítható, vagy lehetséges részleges állapottérbejáró módszerek alkalmazása.

A véges állapotgéppel (FSM) modellezett rendszer esetén tesztesetek automatikus generálása az állapotátmenet ellenőrző módszer lehetséges. Azonban a megvizsgálandó állapotátmenetek száma hatékonytalanná teszi az eljárást: nagy állapottér esetén végtelenül sok vagy végtelenül hosszú teszt eseteket generálhatunk [LY96, vBP94].

EFSM alapú tesztgenerálásra több megoldás született [SKGH97, BH89, BDA, DÜU04, Kim08], azokat szoftver eszközök valósítják meg. A szoftver eszközök, amelyek kritika tárgyát képezik a generált teszt készlet számossága és minősége miatt, a kimerítő keresést, illetve a véletlen séta módszerét támogatják.

A szakirodalom több állapottérbejáró heurisztikát ismer, mint például a mélység korlát, szétszóródásos keresés, költségfüggvény alapú keresés, valószínűség eloszlás alapú keresés, részleges rendezés és véletlen séta. Holzmann megállapítása szerint a véletlen séta, nemcsak a legegszerűbben implementálható módszer, hanem jó eséllyel ez biztosítja a legjobb minőségű keresést [Hol91].

2. Kutatási célkitűzések

A fő kutatási célkitűzésem az EFSM, illetve az abból származó SDL szemantikához egy formális alapokon nyugvó modell felállítása, és azon alapuló szelektív teszteset generáló módszerek definiálása. Mindeközben figyelembe veszem, hogy az e módszerekkel automatikusan generált absztrakt tesztkészlet által tartalmazott konformancia tesztesetek száma kisebb legyen, mint a nyers erő módszerével generált kimerítő tesztkészlet számossága, ugyanakkor a tesztkészletek minőségét jellemző metrikája jobb

legyen, mint a véletlen séta módszerével generált tesztkészlet metrikája. A számosság csökkentésének a tesztkészlet végrehajtási ideje szempontjából van jelentősége

A disszertációmat eredményei három téziscsoportra sorolhatók. Az első csoport egy formális keretrendszert és egy hibalefedettség alapú algoritmust ad a szelektivitás támogatására az automatikus teszteset generálás során, amely felhasználó beavatkozás nélkül képes megfelelni kisebb méretű és jó minőségű tesztkészlet követelményének.

A disszertáció második része kiterjeszti a string szerkesztési távolság alapú módszertant egy olyan algoritmussal, amellyel meghatározható a legnagyobb tömörítés egy adott tesztkészlet sűrűség paraméterhez.

A harmadik téziscsoport absztrakt algoritmusokat javasol, amelyekkel bármely teszteset válogató módszer egy tesztkészlet metrika segítségével szelektív automatikus teszteset generáló algoritmussá alakítható. A javasolt módszer és a konformancia implementáció relációból származtatott metrika csökkenti az első téziscsoportban definiált módszer számításigényét, és a tesztkészlet metrika lehetővé teszi tesztkészletek karbantartási költségének csökkentését spirális fejlesztési modellek alkalmazása esetén.

3. Kutatási módszerek

A kutatási célkitűzések elérésére analitikus és szimulációs kutatási módszert használtam fel. Az analitikus módszereket algoritmus-, gráf-, illetve komplexitáselméleti problémákra, a szimuláción alapuló módszereket pedig a javasolt modellek, algoritmusok tapasztalati értékelésére alkalmaztam.

Az első tézisben több különböző formális szemantikára, konkrétan a véges állapotgép modellre, a kiterjesztett véges állapotgép modellre és a processz algebrára építem. Definícióim segítségével átjárást biztosítok a processz algebra szemantikára értelmezett *formális keretrendszer konformancia tesztelésre* nevű modell [Tre00] és az SDL szemantikája között. A szelektív algoritmusom a mutáció elemzés [DMLS78] megoldására épít, annak működését kísérleti úton vizsgáltam meg.

A második tézis a gráfelméletből ismert párosítási problémát, illetve az algoritmuselméletből ismert string szerkesztési távolságot kiszámító algoritmusra épít. A javasolt módszer polinomiális időben visszavezeti a string távolság alapú teszteset válogatás módszerét az ún. k-párosítási problémára. A módszeremet hatékonyságát kísérleti úton ellenőriztem.

A harmadik tézis az evolúciós algoritmusok elméletére és a szoftverfejlesztésből ismert életciklusmodellekre épít. A javasolt eljárást szimulációs eredmények alapján értékeltem.

4. Eredmények

4.1. Hibalefedettség alapú szelektív automatikus tesztgenerálás SDL specifikációhoz

1. Tézis csoport. *[J2, C1, C6, C7] Kidolgoztam egy hibamodell alapú szelektív módszert, amely egy SDL specifikáció alapján tetszőleges állapottér bejáró algoritmus alkalmazásával automatikusan generál absztrakt konformancia teszteseteket MSC formában. A megoldásom hatékonyságát empirikus vizsgálattal ellenőriztem.*

Automatikus tesztgeneráló szoftverek, illetve szoftver komponensek kifejlesztését megalapozó kutatások közös megállapítása, hogy nagy protokoll modellek esetén a kimerítő keresés helyett szelektív módszereket érdemes alkalmazni [Hol91]. Az SDL alapú AutoLink [SKGH97] a nyers erő módszerét vagy a véletlen keresést támogatja automatikus teszteset generálásra, a szelektív megoldás, a kézi állapottér navigáció, félautomatikus, emberi beavatkozást igényel. A TorX [TB03] és a TGV [JJ02] nevű processz algebra szemantikára épülő [Tre00] eszköz automatikus teszteset generáló modulja a nyers erő módszerét támogatja miközben megjegyzi a szelektivitás szükségességét.

A kiindulási feltételek megegyeznek a fenti két kutatási projekt két alapelvével, és azt két további követelménnyel toldja meg:

- teljesen automatikus teszteset generálás
- fekete dobozos teszteset generálás
- szelektív, azaz korlátozott a generált tesztesetek száma, ami kevesebb, mint a nyers erő módszerével generált
- minőségileg jobb eredmény ad, mint a véletlen séta módszere

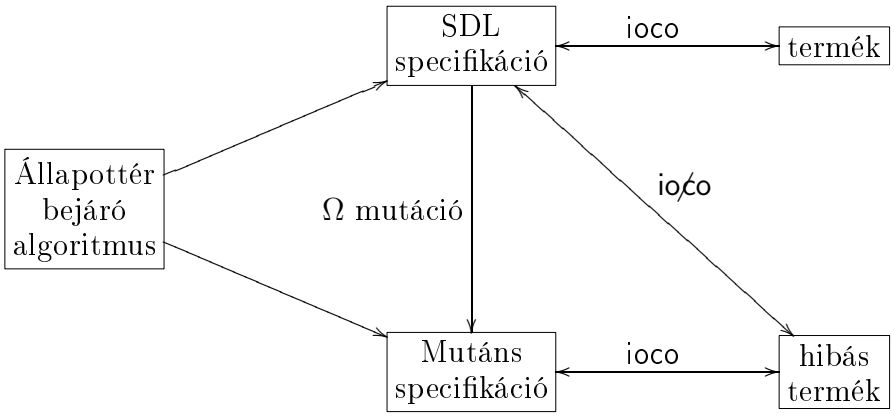
Az automatikus módszerekkel nagyszámú tesztesetet tartalmazó tesztkészletek állíthatók elő rövid idő alatt. Ez a téziscsoport egy olyan megoldást javasol, amely hibamodellt használ a generálandó teszt készlet méretének korlátozására. Az tézisben alkalmazott mutáció alapú technika jól ismert a szoftveres világból: tesztadat válogatásra javasolták eredetileg [DMLS78], ezen kívül fehér dobozos szoftver validációra használták. Az utóbbi évtized kutatási eredménye megmutatta, hogy alkalmazható különböző formalizmusokra is [PG91, WL93, FMDM94, SMFLDS00, FMM+95, FMSM99, ABM98, BOY00, AB99]. SDL specifikációk validációjára több javaslat is született, amely ezt a módszert használja, a 2004-es [1] épít a téziscsoport javaslataira, a 2008-as [WRQB08] független megoldás.

Az e tézisben megfogalmazott állításokkal összhangban lévő és azokat kiegészítő állítások találhatók Pap Zoltán Ph.D. disszertációjában [Pap06]: egy SDL nyelvhez

definiált atomi operátorokból álló, automatikusan alkalmazható hibamodell, amely 1.1 Altézisben alkalmazható, valamint egy automatikus tesztválogatási algoritmus, amely szekvenciálisan alkalmazható a 1.2 Altézisben javasolt algoritmussal.

1.1. Tézis. *Kidolgoztam és formalizáltam egy új, hibamodell alapú keretrendszert SDL nyelvhez, amely lehetővé teszi a fekete dobozos absztrakt teszteset generálást, és a hibamodellek automatikus alkalmazását.*

Kidolgoztam egy formális keretrendszert, amely SDL specifikációk alapján a Pap által javasolt [Pap06] hibamodell felhasználásával automatikusan teszteseteket generál.



1. ábra. A hibamodell alapú automatikus tesztgenerálás modellje SDL specifikációhoz

A tesztgeneráló keretrendszer alapja a *formális keretrendszer konformancia tesztelésre* nevű modell [Tre00] ioco konformancia és elutasítás szerinti előrendezés implementáció relációi. A konformancia reláció az ellenőrzi, hogy az implementáció képes-e a specifikáció által előírt viselkedést produkálni. Az egy hibát tartalmazó specifikációhoz konform implementáció és a hiba nélküli specifikáció között ugyanakkor feltétlenül áll fent ez a reláció.

A 1. ábra mutatja ezt az esetet. A konformancia reláció fennáll mind az SDL specifikáció és a termék, mind a mutáns specifikáció és a hibás termék között. Mutáns specifikációt egy $\omega \in \Omega$ operátor hozza létre az eredeti specifikációból. Az operátor alkalmazása tervezési szempontból vagy új követelményt ad a rendszerhez, és/vagy elvesz egy követelményt vagy módosít egy követelményt. Ezek a követelmények a termék szempontjából azt eredményezik, hogy az egy kötelező követelményt nem valósít meg, ugyanakkor megvalósít egy opcionális követelményt. Ez lehetővé teszi az eredeti

és a mutáns specifikáció relációjának formalizálását, ami elutasítás szerint előrendezés reláció, mert

- a kötelező követelményeknek a mutáns specifikációban is jelen kell lenniük, valamint
- a konformancia reláció nincs tekintettel az opcionális követelményekre, ezért azokat figyelmen kívül kell hagyni.

A reláció fennállásának bizonyításához az i/o ábécéből generálható összes szekvenciát ellenőriznünk kell azt, azonban a reláció hiányát egyetlen teszteset igazolni tudja. Ez a teszteset automatikusan generálható, és egyben alkalmas a konformancia reláció ellenőrzésére.

1.2. Tézis. *Definiáltam egy fekete dobozos szelektív absztrakt konformancia teszteset generáló algoritmust, amely generálási időben egy hibamodell alapján korlátozza a generált tesztkészlet méretét úgy, hogy közben megőrzi annak hibadetektáló képességét. Az algoritmust a 1.1. altézisben megfogalmazott formális keretrendszeren alapuló szoftver eszközzel végzett szimulációval megvizsgáltam.*

A 1. Algoritmus tetszőleges SDL specifikációhoz automatikusan tesztkészletet generál, amely maximális hibalefedettséget biztosít az adott Ω automatikusan alkalmazható hibamodellre. A generált tesztkészlet mérete nem nagyobb, mint a hibamodell által generált hibás rendszerek száma.

Szimulációval megvizsgáltam a fenti algoritmust és összehasonlítottam az INRES, a Conference Protocol, a WAP WTP (Wireless Application Protocol – Wireless Transaction Protocol) és az SS7 MTP2 (Signalling System No. 7 – Message Transfer Part level 2) esetében a véletlen séta alapú módszerrel és a nyers erő módszerével.

A javasolt módszer és az azt megvalósító szoftver eszköz előnye egy bemeneti SDL specifikációhoz adott hibamodell alapján történő korlátos méretű nagy hibadetektáló képességgel bíró tesztkészlet automatikus generálásának képessége. Gyengesége a szükséges nagy számítási igény.

E tézis eredményeit a disszertáció 3. fejezetében mutatom be részletesen, és a [C6, C7, J2, C2] cikkekben publikáltam, amelyekre négy független [1, 2, 3, 4] és egy függő [5] hivatkozás van.

4.2. Optimális string szerkesztési távolság alapú teszteset válogatás

2. Tézis csoport. *[J3, C13] Kiegészítettem a string szerkesztési távolság alapú teszteset válogatás módszertanát úgy, hogy az megtalálja egy bemeneti tesztkészlethez az*

Algoritmus 1: Tesztesetek automatikus generálása EFSM specifikációból adott hibamodell alapján

input : $m = (S, V, I, O, T)$, egy EFSM/SDL specifikáció; *stop* feltétel
output: $\Sigma = \{\ldots, \sigma, \ldots\}$ tesztkészlet, ahol σ egy IOTS.

```
1  for t ∈ T do
2      for ω ∈ Ω do
3          Legyen m' = (S, V, I, O, T'), ahol t' = ω(t), t' ∈ T', t ∈ T;
4          σ = null;
5          repeat
6              if stop then return;
               /* Egy tetszőleges kereső algoritmus bejárja m
                  állapotterét és közben inkrementálisan létrehozza σ-t
                  */
7              σ := derive(m);
8          until out(m, σ) ≠ out(m', σ) ;
9          Σ := Σ ∪ {σ};
10     endfor
11 endfor
12 return Σ
```

adott approximációs küszöbszámhoz tartozó legkisebb méterű és lehető leginkább eltérő teszteseteket tartalmazó részhalmazát. Szimulációval megvizsgáltam a módszer hatékonyságát, és összevetettem azt más módszerekkel.

A tézis a Vuong és Alilovic-Curgus által javasolt [VAC92] és Feijs és társai által kiegészített [FGMT02] módszerre épül. A módszertan alapja, hogy a teszteseteket stringnek tekintve azok különbözősége string szerkesztési távolsággal mérhető. Az előbbi cikk egy string szerkesztési távolság alapú állapottér lefedési metrikát definiál normalizált szerkesztési súlyok meghatározásával. Bevezeti az ε-lefedés fogalmát, vagyis egy teszteset akkor tekinthető redundánsnak, ha távolsága egy másik tesztesettől kisebb, mint egy adott ε paraméter.

Az utóbbi cikk két heurisztika bevezetésével általánosítja a módszert. A jelölt útvonal (marked trace) fogalmának bevezetésével állapotgépbeli hurkok közötti normalizált szerkesztési távolságot definiál (cycling), és korlátozza az egy állapotba való visszatérést (reduction).

Az idézett cikkekben bemutatott módszer el tudja dönteni, hogy két teszt eset redundáns-e. Azt azonban nem állapítja meg, hogy melyiket kell megtartani.

2.1. Tézis. *Konstruktív eljárást definiáltam, amely visszavezeti a bemeneti tesztkészlet adott approximációs küszöbszámhoz tartozó legkisebb részhalmazának string szerkesztési távolságon alapuló megtalálását egy páros gráfbeli párosítási problémára. Bebizonyítottam, hogy a lehető legkisebb méret $O(|\Sigma|^3)$ időben meghatározható, ahol $|\Sigma|$ a tömörítendő tesztkészlet (mint halmaz) számossága.*

A [VAC92, FGMT02] szerinti távolság metrika alapján definiáltam a távolság mátrix fogalmát: $\mathbf{D} = [d_{ij}]$, ahol $d_{ij} = d(\mu(t_i), \mu(t_j))$ és $1 \leq i, j \leq |\Sigma|, i, j \in \mathbb{N}$, valamint μ a tesztesetet jelölt útvonallá alakító operátor.

A 2. Algoritmus bemenete a Σ tesztkészlet távolságmátrixa és egy ε küszöbszám. A kimenet a legtömörebb részhalmaz mérete. A Σ tesztkészlet két diszjunkt halmazra osztható: azon tesztestek Σ'' halmaza, amelyek ε-lefedhetők egymással, illetve azon $\Sigma'_0 \subseteq \Sigma$ halmaz, amelyek elemei nem fedhetők le. Σ'_0 elemeit Σ'-nek mindenképpen tartalmaznia kell, és emellett Σ''-ből a minimális számosságú tesztesetet kell kiválasztanunk, tehát Σ'' maximális redukciója adja a legtömörebb Σ'-t.

Az algoritmus két boolean értékeket tartalmazó mátrixot, $\mathbf{A}$ és $\mathbf{A}_\psi$, és egy páros G' gráfot használ. Az algoritmus előrször meghatározza mely teszt esetek ε-lefedik egymást, és $\mathbf{A}$-ben 1-gyel jelöli azokat. Utána a le nem fedett tesztesetek Σ'_0 halmazát, majd az azonos lefedést biztosító teszteseteket állapítja meg. Ezután a G' páros gráfot konstruálja meg az algoritmus, amelyen a Hopcroft-Karp algoritmussal [CLRS01] meghatározza a ψ maximális párosítást, amit megjelöl az $\mathbf{A}_\psi$ mátrixban. A visszaadott minimális k méret a Σ'_0 méretének és az $\mathbf{A}_\psi$ mátrix alsó- vagy felsőháromszög mátrixa rangjának összege.

Algoritmus 2: String távolság alapú teszteset válogatás visszavezetése páros gráfban maximális párosításra

input : $\mathbf{D}$ távolság mátrix Σ tesztkészlet; ε küszöbszám
output: k, a maximális számú redundáns tesztesetek száma adott ε-hoz

1 $\mathsf{data}(\mathbf{A} = [a_{ij}], a_{ij} \in \{0,1\}; \mathbf{A}_\psi = [a_{\psi ij}], a_{\psi ij} \in \{0,1\}$*;*
2 $G' = (N', E')$ *páros gráf, ahol* $N' = N'_R \cup N'_C$)

`/* Inicializáció, a távolságmátrix meghatározása */`
3 $k := 0, N'_R := \emptyset, N'_C := \emptyset, E' := \emptyset, \mathbf{A} := 0, \mathbf{A}_\psi := 0;$
`/* Az` $\mathbf{A}$ `mátrix meghatározása */`
4 **foreach** i, j, $1 \le i, j \le |\Sigma|$ **do**
5 **if** $d_{ij} < \varepsilon$ **then** $a_{ij} = 1$;
6 **else** $a_{ij} = 0$;
7 **endfch**
`/*` Σ'_0 `méretének meghatározása */`
8 **foreach** i **do**
9 **if** $\sum_j a_{ij} = 0$ **then** $k := k + 1$;
10 **endfch**
`/* Azonos tesztesetek keresése */`
11 **foreach** k, l, $1 \le k \le |\Sigma| - 1, k < l \le |\Sigma|$ **do**
  if $\sum_j (a_{kj} \,\mathsf{xor}\, a_{lj}) = 0$ **then**
12 **foreach** j, $1 \le j \le |\Sigma|$ **do** $a_{lj} := 0$, $a_{jl} := 0$;
13 **endif**
14 **endfch**
15 **endfch**
`/*` G' `gráf konstrukciója */`
16 **foreach** $\sigma_i \in \Sigma$ **do**
17 $N'_R := N'_R \cup \sigma_i$;
18 $N'_C := N'_C \cup \sigma_i$;
19 **endfch**
20 **foreach** i, j, $1 \le i, j \le |\Sigma|$ **do**
21 **if** $a_{ij} > 0$ **then** $E' := E' \cup \{(n'_i, n'_j)\}$;
22 **endfch**
23 Legyen $G'_\psi = (N'_\psi, E'_\psi)$ a Hopcroft-Karp algoritmus által kiválasztott párosítás;
`/* A maximális párosítás megjelölése` $\mathbf{A}_\psi$`-ben */`
24 **foreach** i,j, $1 \le i, j \le |\Sigma|$ **do**
25 **if** $(n'_i, n'_j) \in E'_\psi$ **then** $\mathbf{A}_{\psi\,\mathbf{ij}} = \mathbf{1}$;
26 **else** $\mathbf{A}_{\psi\,\mathbf{ij}} = \mathbf{0}$;
27 **endfch**
28 **return** $k := k + \mathsf{rank}(A^L_\psi)$

Adott távolságmátrix esetén a páros gráf konstruálási költsége $O(|\Sigma|^2)$. A Hopcroft-Karp algoritmus komplexitása [CLRS01] a $G' = (N', E')$ páros gráfra alkalmazva $O(\sqrt{|N'|}|E'|)$. Mivel $|N'| \leq 2|\Sigma|$ és $|E'| \leq |\Sigma|^2$, $O(|\Sigma|^{5/2})$. Ezért az eredő komplexitást az azonos lefedésű teszt esetek keresése határozza meg, ami $O(\Sigma^3)$. [1]

2.2. Tézis. *A legkisebb méretű, és egyben lehető leginkább eltérő teszteseteket tartalmazó tesztkészlet megtalálásának problémáját polinom időben visszavezettem a Dell'Amico és Martello által definiált polinom időben megoldható k-párosítási problémára.*

A tesztesetek közötti redundancia a legkisebb, ha a redukált teszt készletben a tesztesetek páronkénti távolságainak összege maximális. Ha több, mint egy minimális számosságú Σ' megoldás létezik, a legnagyobb belső távolságösszegű preferálandó, ami polinom időben azonosítható. Ez a optimalizálási problémát definiálja:

$$\max \sum_{\sigma_i \in \Sigma', \sigma_j \in \Sigma'} d(\mu(\sigma_i), \mu(\sigma_j)), \tag{1}$$

ahol $\forall i, j : d(\mu(\sigma_i), \mu(\sigma_j)) > \varepsilon$.

Ezt az optimalizálási problémát a gráf párosítási probléma minimális költségű maximális folyam problémává alakításával visszavezettem a Dell'Amico és Martello által definiált polinom időben megoldható k-párosítási problémára:

$$\max \sum_{i=1}^{|\Sigma|} \sum_{j=1}^{|\Sigma|} c_{ij} d_{ij}, \tag{2}$$

$$\sum_{j=1}^{|\Sigma|} c_{ij} \leq 1, i = 1, ..., |\Sigma|, \tag{3}$$

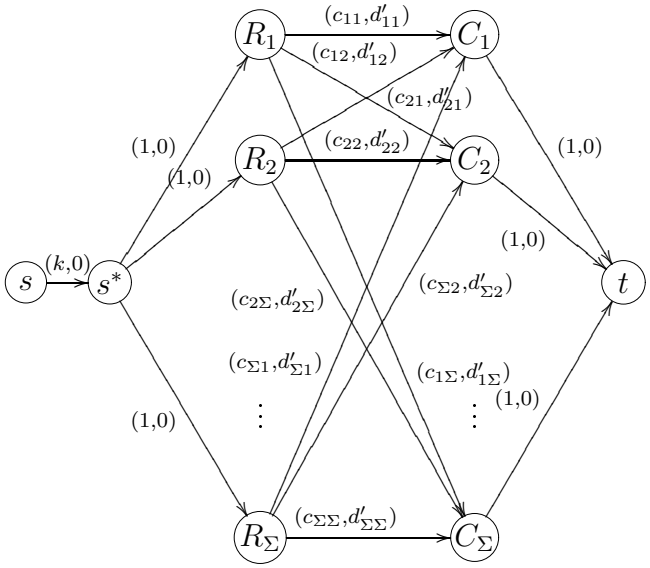
$$\sum_{i=1}^{|\Sigma|} c_{ij} \leq 1, j = 1, ..., |\Sigma|, \tag{4}$$

$$\sum_{i=1}^{|\Sigma|} \sum_{j=1}^{|\Sigma|} c_{ij} = k, \tag{5}$$

ahol $c_{ij} \in \{0, 1\}$ a költség.

Az INRES és a Conference Protocol példájára alkalmazott manuálisan és automatikusan generált teszteseteken végrehajtott kétféle tesztkövetelmény alapú [HGS93]

[1] Azonos sorok vizsgálatára a hash alapú keresést végző Rabin-Karp algoritmust [CLRS01] használva az eredő komplexitást a Hopcroft-Karp algoritmus komplexitása határozza meg, vagyis $O(|\Sigma|^{5/2})$.



2. ábra. A párosítási problémának megfelelő folyamprobléma

teszteset válogatási módszerrel hasonlítottam össze a javasolt string szerkesztési távolság alapú módszert: egy hibalefedettség alapú módszerrel, amelyhez Pap [Pap06] által javasolt módszert és hibamodellt használtam, és egy állapotátmenet lefedés alapú módszerrel.

A 1. táblázat alapján megállapítható, hogy a módszerek közül a string szerkesztési távolság alapú módszer, ami a legkisebb futási időt igényelte a három közül, és közel azonos eredményt adott, mint az FSM alapú, és ezért csak manuálisan alkalmazható állapotátmenet lefedés alapú módszer. A legnagyobb tömörítést elérő, és legjobb hibadetektálási képességgel bíró hibalefedettség alapú módszer állapotátmenet lefedés területén kevésbé hatékony.

A javasolt módszer előnye, hogy szemben más lefedettségi metrika alapú módszerekkel, polinom időben képes teszt készletek redukálására. Hátránya, hogy a ε közelítés miatt nem képes hasonló mértékű tömörítésre.

A módszerhez nyitott kutatási témák kapcsolódnak, mint például teszt esetek hatékony stringgé történő leképezésének vizsgálata.

E tézis eredményeit, amelyeket a [J3, C13, C4] cikkekben publikáltam, a disszertáció 4. fejezetében mutatom be.

Tesztkészlet	Rendszer	Módszer	Redukált méret	Detektált hibák száma	Ellenőrzött átmenetek
Automatikus (8)	INRES	String	3/8	19/25	11/13
		Átmenet	3/8	19/25	11/13
		Hiba	4/8	23/25	11/13
Manuális (13)	INRES	String	6/13	19/25	6/13
		Átmenet	8/13	19/25	10/13
		Hiba	4/13	23/25	8/13
Automatikus (40)	Conference	String	14/40	59/78	39/55
		Átmenet	11/40	59/78	40/55
		Hiba	6/40	60/78	25/55

1. táblázat. Teszteset válogatási kísérletek eredményei

4.3. Iteratív szelektív tesztgenerálás tesztkészlet metrikával

3. Tézis csoport. *[J1, C9] Definiáltam egy az evolúciós algoritmusok mintájára épülő absztrakt módszert, amely csökkenti a fekete dobozos szelektív tesztgeneráló eljárások számítási költségét, valamint iterációt tartalmazó fejlesztési életciklus modell esetén lehetővé teszi az alacsony költségű tesztkészlet karbantartást. Bevezettem a tesztkészlet metrika fogalmát, amelyet iterációs kritériumként alkalmaztam. Tapasztalati úton megvizsgáltam a módszer hatékonyságát.*

Hibalefedettség alapú tesztgenerálás tapasztalata, hogy a hibafelderítés blokkos, vagyis sok teszteset fedezi fel ugyanazt a hibahalmazt. A végső tesztkészletet azok az injektált hibák határozzák meg, amelyek csak egy-egy teszteset fedez fel. Ezért a teljes párosítási probléma mérete, vagyis a teljes futási idő csökkenthető, ha ezeket a teszteseteket keressük, es a többi közül nagy lefedést biztosító teszteseteket választunk ki.

Két absztrakt algoritmust definiáltam formálisan, mindkettő azonos absztrakt műveletekhalmazból építkezik, amelyek a következők: új teszteset generálása (derive), a párosítási probléma költségmátrixának meghatározása (compute), a párosítási probléma megoldása (reduce), párosítási problémák összefűzése (append). Az algoritmusok bemenete egy EFSM specifikáció és egy leállási feltétel, kimenete egy tesztkészlet.

Az 3. algoritmus minden ciklusban új tesztesetet generál, és ellenőrzi, hogy az új teszteset javít-e a tesztkészlet a párosítási probléma kiértékelésekor használt metrikáján, ami a $\mathbf{\Phi}$ mátrixból származtatható. Ezt addig ismétli, amíg egy leállási feltétel nem teljesül.

A 4. algoritmus az iterációs ciklusban azonnal tömöríti az aktuális tesztkészletet valamely teszteset válogatási módszerrel, amely párosítási problémáját a $\mathbf{\Phi}$, illetve a $\mathbf{C}$ mátrix definiálja.

Algoritmus 3: Iteratív szelektív automatikus tesztgenerálás

input : m, F, *stop*
output: Σ
`/* tesztkészlet a` k. `iterációban, egész értékeket tartalmazó mátrix a` k. `iterációban */`
1 data(Σ, $\mathbf{\Phi}$);
`/* Inicializáció */`
2 $k := 0$;
3 $\Sigma[0] := \emptyset$;
4 $\mathbf{\Phi}[0] := 0$
`/* A` k^{th}. `ciklus: */`
5 **repeat**
6 **switch** *véletlen választás* **do**
7 **case** létező teszteset módosítása
8 **foreach** $\sigma \in \Sigma[k-1]$ **do** $\sigma :=$ derive(m, σ);
9 **case** új teszteset generálása
10 $\sigma :=$ derive(m);
11 $\Sigma[k] := \Sigma[k-1] \cup \{\sigma\}$;
12 **endsw**
13 $\mathbf{\Phi}[k]$ meghatározása az aktuális $\Sigma[k]$ és $F \cup \{m\}$ alapján;
14 **if** $\mathbf{\Phi}[k] \overset{(F)}{>} \mathbf{\Phi}[k-1]$ **then**
15 $\Sigma := \Sigma[k]$;
16 **endif**
17 **until** *stop* = false ;

Algoritmus 4: Iteratív szelektív tesztgenerálás azonnali optimalizációval

input : m, F, *stop*
output: σ
`/* tesztkészlet a` k`. iterációban, egész értékeket tartalmazó mátrix a` k`. iterációban boolean értékeket tartalmazó mátrix a` k`. iterációban */`
1 data(Σ,$\mathbf{\Phi}$, $\mathbf{C}$);

`/* Inicializáció */`
2 $k := 0$;
3 $\Sigma[0] := \emptyset$;
4 $\mathbf{\Phi}[0] := 0$;
5 $\mathbf{C}[0] := 0$;

`/* A` k. `ciklus: */`
6 **repeat**
7 **switch** *véletlen választás* **do**
8 **case** létező teszteset módosítása
9 **foreach** $\sigma \in \Sigma[k-1]$ **do** $\sigma :=$ derive(m, σ);
10 **case** új teszteset generálása
11 $\sigma :=$ derive(m);
12 $\Sigma[k] := \Sigma[k-1] \cup \{\sigma\}$;
13 **endsw**
14 $\mathbf{\Phi}[k]$ meghatározása az aktuális $\Sigma[k]$ és $F \cup \{m\}$ alapján;
15 $\mathbf{C}[k]$ meghatározása $\mathbf{\Phi}[k]$ alapján;
16 Legyen
$\Sigma\,[k] :=$ reduce (append ($\Sigma\,[k-1]$, $\Sigma\,[k]$), select (append($\mathbf{C}\,[k-1]$, $\mathbf{C}\,[k]$)));
17 Legyen
$\mathbf{\Phi}[k] :=$ reduce(append($\mathbf{\Phi}[k-1], \mathbf{\Phi}[k]$), select(append($\mathbf{C}[k-1], \mathbf{C}[k]$)));
18 **until** $stop = false$;

Az algoritmusokat két metrikával konkretizáltam: bevezettem hibalefedettség alapú és string szerkesztési távolság metrikákat.

Hibalefedettég alapú teszt készlet metrika bevezetésével iteratívvá tettem – a Pap által javasolt – hibalefedettség alapú tesztválogató algoritmust. A visszavezetés alapját a tesztgenerálás során alkalmazott $\mathsf{fails\_after}$ függvény adja: Legyen $\mathsf{fails\_after} : \Sigma \times M \to \mathbb{N}$ a $\mathsf{fails\_after}$ függvény. Legyen $\forall \sigma \in \Sigma, m \in M : \mathsf{fails\_after}(\sigma, m) = l \iff \mathsf{obs}(\mathsf{substring}(\sigma, l), m) = \mathsf{fail}, \mathsf{obs}(\mathsf{substring}(\sigma, l-1), m) = \mathsf{pass}$, ahol $l \leq \mathsf{length}(\sigma)$, $l \in \mathbb{N}$. Ebből kifolyólag $\mathsf{obs}(\sigma, f) = \text{pass} \iff \mathsf{fails\_after}(\sigma, m) = \mathsf{length}(\sigma)$.

A $\mathsf{fails\_after}$ függvény felhasználásával származtatható a $\mathbf{\Phi}$ tulajdonság mátrix: $\mathbf{\Phi}_{ij} := \mathsf{fails\_after}\ (\sigma_i, \phi_j), \forall i, j \in \mathbb{N}, 1 \leq i \leq |\Sigma|, 0 \leq j \leq |\Phi|$. Ebből Pap algoritmusának [Pap06] inputja a következő módon állítható elő: $\forall i, j \in \mathbb{N}, 1 \leq i \leq |\Sigma|, 1 \leq j \leq |\Phi|$:

$$\mathbf{C}_{ij} = \begin{cases} 0 \iff \mathbf{\Phi}_{i0} = \mathbf{\Phi}_{ij} \iff \mathsf{fails\_after}(\sigma_i, m) = \mathsf{fails\_after}(\sigma_i, \phi_j). \\ 1 \iff \mathbf{\Phi}_{i0} \neq \mathbf{\Phi}_{ij} \iff \mathsf{fails\_after}(\sigma_i, m) \neq \mathsf{fails\_after}(\sigma_i, \phi_j). \end{cases}$$

A Σ tesztkészlet hibalefedettségi metrikáit a $\mathbf{\Phi}_\Sigma$ és $\mathbf{C}_\Sigma$ mátrixok alapján definiáltam. Ezek rendre a Σ tesztkészlet hibadetektáló képességét, Σ teszteseteinek átlagos hosszát, Σ méretét, valamint Σ hibadetektáló képességének egyenletességét veszik figyelembe.

A következő többszintű fitnesz függvények alkalmasak a Σ teszt készlet minősítésére:

1. A Σ teszt készlet által felfedezett hibák száma alapján legyen $cr_1 : \mathbf{C} \to \mathbb{N}$, ahol $\mathbf{C}$ egy boolean értékeket tartalmazó mátrix:

$$cr_1(\mathbf{C}_\Sigma) = \sum_{j=1}^{|\Sigma|} \mathsf{sgn}(\sum_{i=1}^{|F|} c_{\Sigma ij}),$$

   ahol $\mathsf{sgn}(x)$ az előjel függvény: $x < 0 \Rightarrow \mathsf{sgn}(x) = -1, \mathsf{sgn}(0) = 0, x > 0 \Rightarrow \mathsf{sgn}(x) = 1$.

2. Legye a $cr_2 : \mathbf{\Sigma} \to \mathbb{R}$ egy függvény, amely a Σ által tartalmazott teszt esetek átlagos hosszát számítja ki:

$$cr_2(\mathbf{\Phi}_\Sigma) = \frac{\sum_{i=0}^{|F|} \sum_{j=1}^{|\Sigma|} \phi_{\Sigma ij}}{|\Sigma|}$$

3. Legyen $cr_3 : \mathbf{\Sigma} \rightarrow \mathbb{R}$ egy fitnesz függvény, amely a Σ teszt készlet méretét veszi figyelembe: $cr_3(\Sigma) = |\Sigma|$.
4. Legyen $cr_4 : \mathbf{C}_\Sigma \rightarrow \mathbb{R}$ egy függvény, amely a Σ teszt készlet hibadetektáló képességének egyenletességét veszi figyelembe:

$$cr_4(\mathbf{C}_\Sigma) = \frac{\sum_{j=1}^{|\Sigma|}(\sum_{i=1}^{|F|}\mathbf{c}_{\Sigma\, ij})^2}{|\Sigma|}$$

Szimuláció alapú kísérletekkel megvizsgáltam a hibalefedettség alapú iteratív tesztteset generáló algoritmusok hatékonyságát. A vizsgálatokat a Conference Protocol, az INRES, a WAP WTP és az SS7 MTP2 rendszerek modelljeire végeztem el. Megállapítottam, hogy a hibalefedettség alapú iteratív algoritmusok csökkentik a teszt készlet generálási időt, miközben hasonló méretű és hasonló – vagy néhány esetben valamivel jobb – hibalefedettségi mutatóval rendelkeznek, mint az a két lépésből álló nem iteratív módszer, amely először egy nagyméretű teszt készletet generál, majd abból azon hibalefedettség alapú teszt eset válogató algoritmust futtat. A protokoll modell komplexitásának növekedésével arányos volt a futási idő csökkenés.

A string szerkesztési távolság alapú iteratív automatikus tesztgenerálás az INRES példáján mutattam meg, ahol a Σ tesztkészlet minősítésére a következő metrikát definiáltam: A $(\mu(\Sigma), dd)$ pár egy metrikus teret definiál, ahol Σ az $I \cup O$ i/o ábécé feletti tesztesetek halmaza, μ egy esemény-string leképezés, és $dd : \Sigma \times \Sigma \rightarrow \mathbb{R}$ egy távolságfüggvény, amelyre $\forall \Sigma_1, \Sigma_2 \subseteq \Sigma$:

$$dd(\Sigma_1, \Sigma_2) = |\sum_{\sigma_i \in \Sigma_1, \sigma_j \in \Sigma_1} d(\mu(\sigma_i), \mu(\sigma_j)) - \sum_{\sigma_i \in \Sigma_2, \sigma_j \in \Sigma_2} d(\mu(\sigma_i), \mu(\sigma_j))|$$

A módszer előnye, hogy a szakirodalom által ismert lefedettségi metrikák felhasználásával képes már eleve redukált teszt készleteket generálni automatikusan. Hátránya, hogy a redukált teszt készlet mérete nagyobb lehet, mint az evolúciós ciklus nélkül generált, majd optimalizált készlet mérete.

E tézis eredményeit, amelyeket a [C9, J1, C13, C4] cikkekben publikálta, a disszertáció 5. fejezetében mutatom be.

5. Gyakorlati hasznosítás

A disszertációban bemutatott eredmények a tervezési fázisban történő formális modellezés tevékenységet tartalmazó fejlesztési életciklus modellek használatát és ezáltal

a számítógéppel támogatott tervezést mozdíthatja elő az automatikus teszteset generálás hatékonyabbá tételével.

Mindhárom tézis a tesztspecifikáció tevékenységre fokuszál, és azt a célt valósítja meg, hogy az automatikusan generált tesztkészlet redundanciája alacsony legyen, ugyanakkor a teszkészlet megőrizze hatékonyságát. Az első tézisben javasolt megoldás felhasználásával egy SDL-ben adott specifikáció alapján absztrakt konformancia tesztesetek generálhatók egyetlen gombnyomással. A második tézis megközelítése hatékony automatikus teszteset válogatást tesz lehetővé, vagyis csökkenthető általa egy tesztkészlet redundanciája. További a teszt készletek string reprezentációját célzó kutatás eredményeivel kiegészítve valós problémák megoldására is alkalmassá tehető. A harmadik tézisben javasolt algoritmusok tetszőleges tesztkészlet a szakirodalomban metrikából szelektív automatikus tesztgeneráló eljárást készítenek.

Az értekezésben bemutatott szoftver eszköz használható ismert, kereskedelmi forgalomban kapható termékek, konkrétan például a Telelogic Tau [Tau] kiegészítőjeként az azzal automatikusan generált tesztkészlet redundanciáinak felderítésére.

Köszönetnyilvánítás

Abban a szerencsében volt részem, hogy sok ember segítségére és támogatására számíthattam az utóbbi néhány év során. Segítségük nélkül nem jöhetett volna létre ez a disszertáció. Legelőször is köszönetet szeretnék mondani témavezetőimnek, Csopaki Gyulának, aki hallgató korom óra felügyelte munkámat, és Tarnay Katalinnak a iránymutatásukért és támogatásukért. Szeretném megköszönni a Nagysebességű Hálózatok Laboratóriumnak, hogy hosszú éveken át támogatta a kutatási tevékenységem, és a Távközlési és Médiainformatikai Tanszéknek a háttér biztosítását. Különös köszönettel tartozom protokoll tervezés és tesztelés témakörön dolgozó kollégáimnak, szerzőtársaimnak konstruktív javaslataikért, támogatásukért és a közös munka lehetőségéért.

Hivatkozások

[AB99] P.E. Ammann and P.E. Black. *A Specification-based Coverage Metric to Evaluate Test Sets.* In *Proceedings of Fourth IEEE International High-Assurance Systems Engineering Symposium (HASE 99)*, pages 239–248, 1999.

[ABM98] P.E. Ammann, P.E. Black, and W. Majurski. *Using Model Checking to Generate Tests from Specifications.* In *Second IEEE International Conference on Formal Engineering Methods*, pages 46–54, 1998.

[BDA] C. Bourhfir, R. Dssouli, and E.M. Aboulhamid. *Automatic Test Generation for EFSM-based Systems.* `http://citeseer.nj.nec.com/114451.html`.

[BH89] L. Bromstrup and D. Hogrefe. Tesdl: Experience with generating test cases from sdl specifications. In *Fourth SDL Forum*, pages 267–279, 1989.

[BOY00] P.E. Black, V. Okun, and Y. Yesha. *Mutation Operators for Specifications.* In *The Fifteenth IEEE International Conference on Automated Software Engineering, Proceedings ASE 2000*, pages 81–88, 2000.

[CLRS01] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to Algorithms.* MIT Press and McGraw-Hill, second edition, 2001.

[DMLS78] R.A. De Millo, R.J. Lipton, and F.G. Sayward. *Hints on Test Data Selection: Help for the Practicing Programmer. IEEE Computer*, 11(4):34–41, April 1978.

[DÜU04] Ali Y. Duale and M. Ümit Uyar. A method enabling feasible conformance test sequence generation for efsm models. *IEEE Transactions on Computers*, 53(5):614–627, 2004.

[FGMT02] L.M.G. Feijs, N. Goga, S. Mauw, and J. Tretmans. Test selection, trace distance and heuristics. In *Testing Communication Systems XIV*, pages 267–282, Berlin, Germany, 2002.

[FMDM94] S.C.P.F. Fabbri, J.C. Maldonado, M.E. Delamaro, and P.C. Masiero. *Mutation Analysis Testing for Finite State Machine.* In *Proc. ISSRE'94 - Fifth International Symposium on Software Reliability Engineering*, pages 220–229, California, USA, 1994.

[FMM+95] S.C.P.F. Fabbri, J.C. Maldonado, P.C. Masiero, M.E. Delamaro, and E. Wong. *Mutation Testing Applied to Validate Specifications Based on Petri Nets. FORTE'95 - 8th International IFIP Conference on Formal Description Techniques for Distributed Systems and Communications Protocol*, 1995.

[FMSM99] S.C.P.F. Fabbri, J.C. Maldonado, T. Sugeta, and P.C. Masiero. *Mutation Testing Applied to Validate Specifications Based on Statecharts.* In *Proc. ISSRE'99 - 10th International Symposium on Software Reliability Engineering*, pages 210–219, Florida, USA, 1999.

[HGS93] M.J. Harrold, R. Gupta, and M.L. Soffa. A methodology for controlling the size of a test suite. *Transactions on Software Engineering and Methodology*, 2(3):270–285, July 1993.

[Hol91] G.J. Holzmann. *Design and validation of computer protocols.* Prentice Hall, 1991.

[IT97] ITU-T. *Recommendation Z.100 – Supplement 1: SDL+ methodology, Use of MSC and SDL (with ASN.1)*, 1997.

[JJ02] C. Jard and T. Jeron. TGV: Theory, principles and algorithms. In *6th World Conference on Integrated Design and Process Technology (IDPT 2002)*, June 2002.

[Kim08] T-H. Kim. Test generation for a protocol specified in sdl with complex loops by event-based efsm modeling. *IJCSNS International Journal of Computer Science and Network Security*, 8(3), March 2008.

[LY96] D. Lee and M. Yiannakakis. *Principles and Methods of Testing Finite State Machines – A Survey. Proc. of the IEEE*, 43(3):1090–1123, 1996.

[Pap06] Z Pap. *Detection, diagnosis and correction of faults in telecommunications software development based on formal methods.* PhD thesis, Budapest University of Technology and Economics, 2006.

[PG91] R.L. Probert and F. Guo. *Mutation Testing of Protocols: Principles and Preliminary Experimental Results.* In *Proc. Protocol Test Systems, III.*, pages 57–76, 1991.

[SKGH97] M. Schmitt, B. Koch, J. Grabowski, and D. Hogrefe. Autolink - a tool for the automatic and semi-automatic test generation. In *Seventh GI/ITG Technical Meeting on Formal Description Techniques for Distributed Systems*, 1997.

[SMFLDS00] S.R.S. Souza, J.C. Maldonado, S.C.P.F. Fabbri, and W. Lopes De Souza. *Mutation Testing Applied to Estelle Specifications. Software Quality Journal*, 8(04), 2000.

[Tau] Telelogic Tau. `http://www.telelogic.com`.

[TB03] G.J. Tretmans and H. Brinksma. Torx: Automated model-based testing. In *First European Conference on Model-Driven Software Engineering*, pages 31–43, Nuremberg, Germany, December 11-12 2003.

[Tre00] J. Tretmans. *Specification Based Testing with Formal Methods: A Theory.* In A. Fantechi, editor, *FORTE / PSTV 2000 Tutorial Notes*, Pisa, Italy, October 10 2000.

[VAC92] Son T. Vuong and Jadranka Alilovic-Curgus. On test coverage metrics for communication protocols. In *Proceedings of the IFIP TC6/WG6.1 Fourth International Workshop on Protocol Test Systems IV*, pages 31–45, 1992.

[vBP94] G. v. Bochmann and A. Petrenko. *Protocol Testing: Review of Methods and Relevance for Software Testing.* In *ISSTA*, pages 109–124, 1994.

[WL93] C.-J. Wang and M. T. Liu. *Generating Test Cases for EFSM with Given Fault Models.* In *INFOCOM 93*, volume 2, pages 774–781, 1993.

[WRQB08] W.E. Wong, A. Restrepo, Y. Qi, and Choi B. *An EFSM-based Test Generation for Validation of SDL specifications.* In *Automation of Software Test 2008*, pages 25–32, Leizpig, Germany, May 2008.

Publikációk

Külföldön megjelent idegen nyelvű folyóirat cikkek

[J1] G. Kovács, Z. Pap, G. Csopaki, and K. Tarnay, "Iterative automatic test generation method for telecommunication protocols," *Computer Standards and Interfaces*, vol. 28, no. 4, pp. 412–427, 2006.

Magyarországon megjelent idegen nyelvű folyóiratcikkek

[J2] G. Kovács, Z. Pap, and G. Csopaki, "*Automatic Test Selection based on CEFSM Specifications*," *Acta Cybernetica*, vol. 15, pp. 583–599, 2002.

Magyar nyelvű folyóiratcikkek

[J3] G. Kovács, "Teszteset válogatás távolság metrikával," *Híradástechnika*, vol. LXI, pp. 41–43, January 2006.

Lecture Notes in Computer Science cikk

[C1] G. Csopaki, G. Horváth, and G. Kovács, "*Communication Protocol Implementation in Java*," in *Lecture Notes in Computer Science 1905: Interactive Distributed Multimedia Systems and Telecommunication Services*, (Enschede, The Netherlands), pp. 254–265, October 17-20 2000.

[C2] G. Kovács, Z. Pap, D. Le Viet, A. Wu-Hen-Chang, and G. Csopaki, "*Applying Mutation Analysis to SDL Specifications*," in *11th International SDL Forum: SDL 2003: System Design* (R. Reed and J. Reed, eds.), (Stuttgart, Germany), pp. 269–284, 2003.

[C3] Z. Pap, M. Subramaniam, G. Kovács, and G. Á. Németh, "A bounded incremental test generation algorithm for finite state machines," in *19th IFIP Testing of Communicating Systems (TestCom/FATES)*, (Tallinn, Estonia), pp. 244–259, June 26-29 2007.

[C4] G. Kovács, G. Á. Németh, M. Subramaniam, and Z. Pap, "*Optimal string edit distance based test suite reduction for SDL specifications*", in *LNCS 5719: 14th International SDL Forum: SDL 2009* (R. Reed, A. Bilgic and R. Gotzhein, eds.), (Bochum, Germany), pp. 82-97, September 22-24 2009.

Nemzetközi konferencia-kiadványban megjelent idegen nyelvű előadások

[C5] G. Gordos, G. Csopaki, Z. Werner, G. Horvath, G. Kovacs, and T. Kerecsen, "*Automatic converion of SDL into modern oop languagues with a Java-based compiler*," in *International Workshop on Intelligent Communication Technologies and Applications*, (Neuchatel, Switzerland), May 5-7 1999.

[C6] G. Kovács, Z. Pap, and G. Csopaki, "*Automatic Test Selection based on CEFSM Specifications*," in *7th Symposium on Programing Languages and Software Tools* (T. Gyimothy, ed.), (Szeged, Hungary), pp. 84–97, June 15-16 2001.

[C7] G. Kovács, Z. Pap, and G. Csopaki, "*Automatic test selection method applied to WAP*," in *EUNICE 2001*, (Paris, France), September 3-5 2001.

[C8] G. Csopaki, T. Kasza, G. Kovács, and M. Szücs, "*Applicability of UML in the protocol development process*," in *Polish-Czech-Hungarian Workshop on Circuit Theory, Signal Processing and Telecommunication Services*, (Budapest), September 14-17 2001.

[C9] G. Kovács, D. Le Viet, and A. Wu-Hen-Chang, "*Iterative automatic test generation*," in *EUNICE 2003*, (Balatonfüred), September 8-10 2003.

[C10] C. Lukovszki, L. Kovács, G. Kovács, A. Foglar, E. Areizaga, and Z. Ghebretebsaé, "Revolutionary ipv6 optimized access solution," in *11th European Conf. on Networks and Optical Communications (NOC)*, (Berlin, Germany), July 11-13 2006.

[C11] H. Mickelsson, A. van Neerbos, P. Nooren, M. Prins, K. Oberle, D. Jocha, B. Radier, G. Kovács, M. Thakur, I. Pinilla, E. Areizaga, A. Gamelas, and A. Sitek, "Nomadism/fmc use cases and aaa impact," in *BroadBand Europe*, (Geneva, Switzerland), December 12-14 2006.

[C12] Z. Kardkovács, E. Lejtovicz, and G. Kovács, "Context identification: A relational database approach," in *The 3rd Language & Technology Conference* (Z. Vetulani, ed.), (Poznan, Poland), pp. 211–215, October 1-5 2007.

[C13] G. Kovács, G. Á. Németh, Z. Pap, and M. Subramaniam, "Deriving compact test suites for telecommunication software using distance metrics," in *Software, Telecommunications and Computer Networks (SoftCOM)*, (Split-Dubrovnik, Croatia), September 25-27 2008.

Egyéb közlemények

[O1] D. Jocha and G. Kovács, "Introduction to the MUSE FMC architecture," in *MUSE Summer School*, (Budapest), July 5 2007.

Hivatkozások

[1] T. Sugeta, J. Maldonado, and W. Wong, "Mutation testing applied to validate sdl specifications," in *Testing of Communicating Systems. 2004.03.17-2004.03.19 193-208. 2004.*, (Oxford, England), pp. 193–208, March 17-19 2004.

[2] M. Hong and Z. Lu, "A framework for testing web services and its supporting tool," in *SOSE 2005: IEEE International Workshop on Service-Oriented System Engineering*, (Beijing, China), pp. 199–206, October 20-21 2005.

[3] Y. Jiang, S. Hou, J. Shan, Z. L., and B. Xie, "Contract-based mutation for testing components," in *ICSM 2005: 21st IEEE International Conference on Software Maintenance*, (Budapest, Hungary), pp. 483–492, September 26-29 2005.

[4] Y. Jiang, S. Hou, J. Shan, L. Zhang, and B. Xie, "An approach to testing black-box components using contract-based mutation," *INTERNATIONAL JOURNAL OF SOFTWARE ENGINEERING AND KNOWLEDGE ENGINEERING 18: (1) 93-117 (2008)*, vol. 18, no. 1, pp. 93–117, 2008.

[5] Z. Andriska, G. Bátori, D. Le Viet, A. Wu-Hen-Chang, and G. Csopaki, "Tool for automatic test selection based on formal specification," *Híradástechnika*, vol. LVII, pp. 37–43, December 2002.

[6] K. Oberle, S. Wahl, and A. Sitek, "Enhanced methods for SIP based session mobility in a converged network," in *16th Mobile and Wireless Communications Summit*, (Budapest, Hungary), pp. 1–5, July 1-5 2007.