

Sonlu Durum Makinelerinin Fourier Analizi Tabanlı Sınanması

Savaş Takan ve Tolga Ayav

İzmir Yüksek Teknoloji Enstitüsü
Bilgisayar Mühendisliği Bölümü, Urla, İzmir.
{savastakan,tolgaayav}@iyte.edu.tr

Özet Sonlu durum makinesi (FSM), devre ve yazılım testlerinde yaygın kullanıma sahip bir modelleme tekniğidir. FSM'lerin testi için literatürde çeşitli yöntemler bulunmakla birlikte, modellerin büyümesi sonucu test kümesinin büyüklüğü, hata yakalama oranı ve test üretim süresi gibi konularda yüksek başarıma sahip alternatif test yöntemlerine ihtiyaç bulunmaktadır. Bu çalışma ikili fonksiyonların Fourier analizine dayanan yeni bir test oluşturma yöntemi önermektedir. İlk sonuçlar, fonksiyonun frekans bileşenlerinden yararlanarak oluşturulan test takımının daha yüksek bir performansa sahip olduğunu göstermektedir. Önerilen yöntem, karakteristiği, maliyeti ve hata yakalama oranı üzerinden literatürden seçilen iki yöntemle karşılaştırılmıştır.

Anahtar Kelimeler: Sonlu durum makinesi, Test takımı üretimi, Fourier dönüşümü, F yöntemi, W yöntemi, UIO yöntemi

Fourier Analysis-Based Testing of Finite State Machines

Savaş Takan and Tolga Ayav

İzmir Institute of Technology
Dept. of Computer Engineering, Urla, İzmir

Abstract Finite State Machine (FSM), as a formal modeling technique to represent both circuits and software, has been widely used in testing. FSM testing is a well-studied subject and there are several test generation methods. However, the current increase in the demand for pervasive and safety critical systems as well as the increase in software size calls for more rigorous methods that can produce more effective test suites particularly in terms of size, time spent for test generation and fault detection ratio. In this study, we propose a new test generation method based on the Fourier analysis of Boolean functions. An analysis on the effects of the various frequency components of the function output allow us to generate test suites with better performance characteristics. We compare our F-method with the two existing methods.

Keywords: Finite State Machine, Test suite, Fourier transformation, F method, W method, UIO method

1 Giriş

Biçimsel modeller kullanarak test üretilmesi yazılım testinin aktif araştırma konularından birisidir [8]. Bu test yöntemleri, ara yüz tasarımından uçak yazılımlarına ve gerçek zamanlı sistemlere kadar birçok alanda kullanılmaktadır [1]. Biçimsel modeller kullanılarak oluşturulmuş testler, test edilen modelin gereksinimleri sağlayıp sağlamadığını sınamakta kullanılırlar. Yazılım alanında Finite State Machine (FSM), Petri Net ve UML gibi çeşitli modellerden yararlanılmaktadır. Bu makale FSM modelini ele almaktadır. Literatürde FSM'lerden test takımı yaratan birçok yöntem bulunmaktadır. Bunlar arasında en yaygın kullanılanları W, Wp, UIO, UIOv, DS, HSI ve H isimlendirmeleriyle anılanlardır [8] [13] [5]. Bu projede, ikili fonksiyonların Fourier analizine dayanan yeni bir test üretme tekniği önerilmektedir. Fourier dönüşümü matematikte, bilgisayar ve diğer mühendislik bilimlerinde geniş bir kullanım alanına sahiptir [12]. Fourier dönüşümüyle elde edilen katsayılar ilgili bileşenlerle fonksiyon çıkışı arasındaki ilintileri vermektedir. Buradan yola çıkarak, ileride gösterileceği gibi fonksiyona etkisi yüksek olan bileşenler belirlenerek bu bileşenlere bağlı geçişler seçilmiştir. Daha sonra aç gözlü algoritması ile bunlardan test takımı elde edilmiştir. Önerilen yöntem, mutasyon çözümlemesi kullanılarak literatürden alınan W ve UIO yöntemleri ile karşılaştırılmıştır.

2 Sonlu Durum Makinesi Testi

Bir sonlu-durum makinesi $M = (I, O, S, f, g, \sigma)$ 6-lısı'ndan oluşur. Burada I, giriş sembolleri sonlu kümesini, O, çıkış sembolleri sonlu kümesini, S, sonlu durumlar kümesini, $f: S \times I \rightarrow S$ bir sonraki durumu belirleyen geçiş fonksiyonunu, $g: S \times I \rightarrow O$, çıkış fonksiyonunu, σ ise makinenin başlangıç durumunu göstermektedir.

Test ile ilgili bazı önemli tanımlar aşağıda verilmiştir.

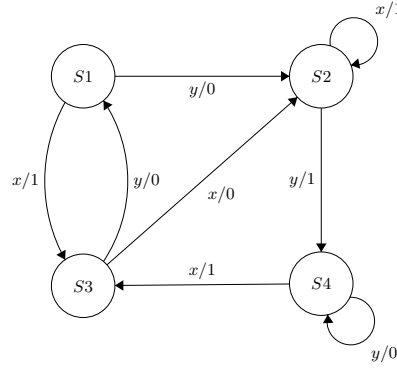
Test Senaryosu: Belirli bir program yolunu çalıştırmak ya da bir gereksinim ile uyumluluğunu doğrulamak gibi belirli bir amaç veya test koşulu için geliştirilen, bir dizi girdi değeri, test öncesi yürütülmesi gereken önkoşullar, test sonrası oluşması beklenen sonuçlar ve koşullar bütünü.

Test Takımı: Bir sistem veya bileşeni test etmek için oluşturulmuş test senaryoları kümesi. Öyle ki, bir test senaryosu için ardkoşul olan bir durum bir diğeri için ön koşuldur.

Genel anlamda, FSM test süreci aşağıdaki temel adımlardan oluşmaktadır [1]:

- Transfer dizisi $T(s_i)$, FSM'yi istenilen bir duruma götürmek için kullanılır.
- Sınama dizisi VER_j FSM üzerinde uygulanır ve istenilen çıkışların üretilip üretilmediği kontrol edilir.
- Sıfırlama dizisi sayesinde s_0 ilk duruma geri gidilir.

Bu bölümde, Unique Input Output (UIO) ve W yöntemleri açıklanacaktır. Tüm açıklamalar Şekil 1'de görülen örnek FSM üzerinden yapılacaktır.

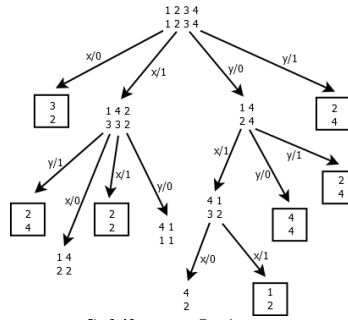


Şekil 1: örnek FSM

2.1 UIO Yöntemi

X / Y , X girişine yanıt olarak başka bir durum çıktı dizisi Y üretmezse, bu bir UIO dizisidir. UIO yöntemi, UIO dizilerini kullanarak FSM'nin doğruluğunu araştırır. Detaylar için literatürdeki çalışmalara başvurulabilir [9] [4].

Örnek üzerinde bakacak olursak; ilk aşamada, tüm durum grupları UIO Metodu'nda kök olarak seçilir. Dallanmayı önleyen 2 durum vardır. İlki, grupların tekil ya da aynı durumlardan oluşmasıdır. Diğeri ise durumların ağacın yukarısında görülebilmesidir. UIO ağacı Şekil 2'de görülmektedir. Dallanmanın sonunda, yukarıdan aşağıya doğru durumlar seçilir. Her bir durumun kökten kendisine kadar oluşturduğu yol, test senaryolarını meydana getirir. Elde edilen test senaryoları Tablo 1'de verilmiştir.



Şekil 2: UIO Ağacı

States	Input	Output
S1	YXX	011
S2	Y	1
S3	X	0
S4	YY	00

Tablo 1: Yarattılan Test Senaryoları

2.2 W Metodu

FSM'deki her bir çift durumun davranışını ayırt edebilen bir girdi dizisine W-kümesi denir. W-metot, W-kümelerini kullanarak FSM'nin doğruluğunu araştırır. Örnek vermek gerekirse, W-metodunda ilk önce Tablo 2 oluşturulur. Bu tabloda ikili olarak kontroller yapılır. Elde edilen ayrıma göre girişler seçilir. Örneğin S1, S2 satırlarına bakılacak olursa, y girişinin farklı çıkış ürettiği görülmektedir. Bu nedenle S1 ve S2 için y girişi alınmıştır. S2 ve S3 için y çıkışının farklı durumlara götürdüğü görülmüştür. Dolayısıyla burada y girişi seçilmiştir. S3 ve S4 için ise, farklı çıkışlar oluşturması nedeniyle x girdisi alınmıştır. Son olarak S4 ve S1 için, ileriki durumu farklılaştırması nedeniyle y girişi seçilmiştir. Elde edilen her değer eklenerek aşağıdaki gibi yazılmıştır. Son olarak, giriş dizileri ters sırada alınmış ve test senaryoları aşağıdaki adımlarla oluşturulmuştur [11][7][13]:

- S1,S2 = y
- S2,S3 = yy
- S3,S4 = yyx
- S4,S1 = yyxy
- W = {yyxy,yxy,xy,y}

Current States	x	y	Next States	Next States
S1	1	0	S3	S2
S2	1	1	S2	S4
S3	0	0	S2	S1
S4	1	0	S3	S4

Tablo 2: FSM Tablosu

3 Önerilen F Yöntemi

F yöntemi, test senaryolarını elde etmek için Fourier katsayılarından yararlanarak en sık kullanılan geçişleri bulmayı amaçlamaktadır. Bunu yapmak için FSM ilk olarak ikili bir forma dönüştürülür. Örneğin, A durumu 00, B durumu 01, x

0, y 1'dir. Katsayılar, giriş ve durum bilgileri kullanılarak hesaplanır. Kendi belirlediğimiz bir parametre değeri oluşturulur. Parametre değeri oluşturulduktan sonra, bu katsayılar arasından eşğin üstünde kalanlar seçilir. Seçilen katsayılar kullanılarak uygun geçişler belirlenir. Bu işlemten sonra, belirlenen geçişleri birleştiren aç gözlü algoritma ile test senaryoları oluşturulur. F yöntemiyle test senaryosu üretimi ileride örnek FSM üzerinden detaylıca anlatılacaktır. Öncesinde (takip eden alt bölümde), Fourier analizinin temelleri ele alınmaktadır.

3.1 Fourier Analizi

İkili fonksiyonların Fourier analizi son on yılda matematik ve mühendislik alanlarında ilgi çeken ve çok araştırılan bir konu olmasına rağmen, çok az pratik uygulaması bulunmaktadır. Fourier analizinin bir gereği olarak, 0 ve 1 yerine 1 ve -1 *yanlış* ve *doğru* değerler olarak kullanılmaktadır. Fourier dönüşümü, ikili fonksiyonları $f : \{-1, 1\}^n \rightarrow \mathbb{R}$ şeklinde sözde ikili fonksiyonlara dönüştürür. Temel tanım ve teorem, kanıtlar sunulmaksızın aşağıda verilmiştir. Bu konuda daha fazla açıklama, örnek ve teorem O'Donnell [12] ve Wolf'un [6] çalışmalarında bulunabilir.

Teorem 1 (Fourier Açılımı). *Her bir fonksiyon $f : \{-1, 1\}^n \rightarrow \mathbb{R}$ Fourier açılımı ile benzersiz bir şekilde ifade edilebilir.*

$$f(x) = \sum_{S \subseteq [n]} \hat{f}(S) \chi_S$$

Burada $\hat{f}(S)$ Fourier katsayılarını, $\chi_S = \prod_{i \in S} x_i$ ise Parity fonksiyonunu ifade etmektedir.

Tanım 1 (İç Çarpım). *İç çarpım aşağıdaki formülle bulunur:*

$$\langle f, g \rangle = \frac{\sum_{x \in \{-1, 1\}^n} f(x)g(x)}{2^n}$$

Fourier katsayıları aşağıdaki gibi hesaplanır.

$$\hat{f}(S) = \langle f, \chi_S \rangle$$

Aşağıda basit bir ikili fonksiyon üzerinden Fourier açılımının nasıl hesaplandığı gösterilmektedir. Fonksiyonumuz aşağıdaki gibi olsun:

$$f = a \vee b \wedge c$$

f 'in doğruluk yöneyinin aşağıdaki gibi olduğu kolayca görülebilir:

$$[FFFTTTTT]$$

Fourier açılımının genel yapısı şu şekildedir:

$$f = \hat{f}(\emptyset) + \hat{f}(1)a + \hat{f}(2)b + \hat{f}(3)ab + \hat{f}(4)c + \hat{f}(5)ac + \hat{f}(6)bc + \hat{f}(7)abc$$

Tamam 1 uyarınca, ilk Fourier katsayısı $\hat{f}(\emptyset)$ aşağıdaki şekilde hesaplanır:

$$\hat{f}(\emptyset) = \frac{1}{2^3}(1 + 1 + 1 - 1 - 1 - 1 - 1 - 1) = 0.25$$

Benzer şekilde ikinci katsayı aşağıdaki gibi bulunur:

$$\hat{f}(1) = \frac{1}{2^3}(1 \cdot 1 + 1 \cdot 1 + 1 \cdot 1 - 1 \cdot 1 - 1 \cdot -1 - 1 \cdot -1 - 1 \cdot -1 - 1 \cdot -1) = 0.75$$

Tüm katsayılar hesaplandığında Fourier açılımı aşağıdaki şekilde elde edilmiş olur:

$$f = 0.25 + 0.75a + 0.25b + 0.25ab + 0.25c + 0.25ac - 0.25bc - 0.25abc$$

3.2 F Yöntemiyle Test Takımı Yaratma

Fourier dönüşümü ikili fonksiyonlar üzerinde tanımlı olduğundan, FSM öncelikle bu forma dönüştürülür. Bu dönüşüm, temelde durumların (state) ikili kodlanmasından ibarettir. Örneğin, S1, 00 olarak; S2, 01 olarak kodlanır. Girdi değerleri x ve y, sırasıyla 1 ve 0'a dönüştürülür.

Çıkış fonksiyonu Denklem 1'deki gibidir. Çıkış fonksiyonu giriş ve durumlardan oluşmuştur.

$$O = (\bar{S}_1 \wedge \bar{S}_2 \wedge X) \vee (\bar{S}_1 \wedge S_2 \wedge X) \vee (\bar{S}_1 \wedge S_2 \wedge \bar{X}) \vee (S_1 \wedge S_2 \wedge X) \quad (1)$$

Doğruluk tablosu çıktı fonksiyonu kullanılarak elde edilir. Girdiler ve durumlar Fourier katsayılarını elde etmek için birlikte kullanılır. Fourier açılımı Denklem 2'de verilmiştir:

$$O = 0 - 0.5S_1 + 0.5S_2 + 0 + 0.5X + 0 + 0 + 0.5S_1S_2X \quad (2)$$

Örnekte görüldüğü gibi, bazı katsayılar sıfırdır. Bu nedenle, ilgili bileşenlerin fonksiyon üzerinde etkisi yoktur. Bu bileşenlerle ilgili olan test senaryolarını silmek, test takımı boyutunu ve test takımı maliyetini azaltır. Bu nedenle bir eşik değeri belirlenerek eşik altına kalan bileşenler ihmal edilir. Testlerin belirlenmesinde ve test senaryo sayısının azaltılmasında rol oynayan iki parametre belirlenmiştir. İlki katsayıları büyük olan bileşenlerin seçilmesini sağlayan eşik değeri, diğeri ise seçilen bileşenlere ilişkin geçişlerin sayısını sınırlamakta kullanılan sıfır ile bir arasında bir orandır.

Fourier açılımına eşik değeri uygulanmasıyla eşik altındaki bileşenler ihmal edildiğinde O fonksiyonu Denklem 3'te görüldüğü şekle indirgenir.

$$O = 0.5S_1 + 0.5S_2 + 0.5X + 0.5S_1S_2X \quad (3)$$

Denklem 3'te gösterildiği gibi, katsayılar belirlendikten sonra uygun geçişler seçilir. Tablo 3'de ilgili algoritmalar çalıştırıldıktan sonra seçilmiş olan geçişler gösterilmektedir.

S1	S2	X	S1	S2	Output
0	0	0	1	0	0
0	1	0	1	1	1
1	1	1	0	1	1
0	0	1	0	1	1
0	1	1	1	0	1
1	0	0	0	1	0

Tablo 3: Bütün katsayılar uygulandıktan sonra

Fourier terimleriyle ilintili geçişleri seçmek için Şekil 3 ve 4'te görülen algoritmalar geliştirilmiştir. İlk algoritmada katsayılar, geçişler, birinci ve ikinci parametre değeri olmak üzere dört giriş bulunmaktadır. Bu algoritmanın çıktısı, terimler tarafından seçilen geçişler oluşturmaktadır. 'termSize' parametresi, kaç Fourier teriminin seçileceğini söylemektedir. 'transitionSize' parametresi ise kaç tane girişin alınacağını söylemektedir. Bu iki parametre sayesinde, test takımının boyutu ve etkinliği belirlenebilmektedir.

```

Input: transitions ,term,transitionSize,termSize
sort(term);
n = 0;
for  $i \leftarrow termSize$  to 0 do
    term = term.removeLast();
    if  $term.coefficient \neq 0$  then
        for  $transition$  in transitions do
            if  $check(term, transition)$  then
                temp.add(transition);
                if  $temp.size() == transitionSize$  then
                    | return temp;
                end
                termSet.add(coeffcient);
            end
        end
    end
    if  $termSet.size() == termSize$  then
        | return temp;
    end
end
return temp;

```

Şekil 3: Geçişlerin seçimini gerçekleştiren sözde kod

Diğer algoritma ise (Şekil 4), Fourier değişkenlerini bir şablon olarak alır. Geçişin uygun olup olmadığı, bu şablondan kontrol edilir. Desen bir ise ve herhangi bir değişiklik yoksa, program yanlış döner. Diğer yandan, eğer desen sıfır

ise ve bir deęişiklik varsa, program yine yanlış döner. Programın sonunda doğru döner.

```
Input: term, transition
Output: boolean
for  $i \leftarrow 0$  to  $transition.state.length()$  do
  if  $term.variables[i] == '1'$  then
    if  $transition.state[i] == transition.nextState[i]$  then
      return false;
    end
  else
    if  $transition.state[i] != transition.nextState[i]$  then
      return false;
    end
  end
end
return true;
```

Şekil 4: Check(...) sözde kodu

Seçilen geçişler, ağgözlü algoritması kullanılarak birleştirilir. Bunu yapmak için ilk geçiş bir çözüm olarak kabul edilir. Yeni geçişler her zaman önceki çözümlere eklenmeye çalışır. Mevcut çözümlere dahil olamayan geçişler, yeni bir çözüm olarak kabul edilir.

4 Deęerlendirme

Karşılaştırma yapmak için iki adet algoritma kullanılmıştır. Dolayısıyla yaratılan algoritmanın verdiği sonuçlar, karşılaştırılan bu iki algoritma ile sınırlıdır. Ancak bu alanda pek çok algoritma bulunmaktadır. Mevcut algoritmaların hepsi ile kıyas yapabilmek mümkün olmadığı için, karşılaştırma yapmak üzere en temel algoritmalar seçilmiştir.

Bu bölümde F yöntemi, W ve UIO yöntemleriyle karakteristik, maliyet, etkinlik gibi üç farklı kriter üzerinden karşılaştırılmaktadır. Sıfırlama sayısı ve test senaryolarının ortalama uzunluğu, test takımının karakteristiğini belirlemektedir. Yöntemlerin maliyetini hesaplamak için test takımının uzunluğuna bakılmıştır. Yöntemlerin etkinliği ise hata yakalama oranlarıyla ölçülmüştür. Bu değerler, 'durum, girdi ve çıktı' ile incelenmiştir. Grafikler 3 tip konfigürasyon üzerinden oluşturulmuştur:

- Giriş sayısı deęişkendir, çıkış sayısı 4 ve durum sayısı 6'dır.
- Çıkış sayısı deęişkendir, giriş sayısı 4 ve durum sayısı 6'dır.
- Durum sayısı deęişkendir, giriş ve çıkış sayıları 4'tür.

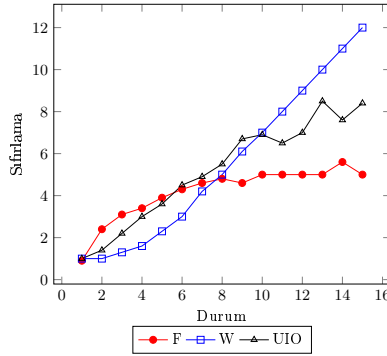
Datalar, yani FSM'ler, sonuçların yanlış olması engellenmek için, belli başlı kriterler kullanılarak rastgele biçimde oluşturulmuştur. Bu kriterler, oluşturul-

lacak FSM'nin durum, sonuç ve giriş sayılarının ayrı ayrı değiştirilmesiyle yaratılmıştır. Böylece, araştırılan özelliğin kriterlerin değişimine göre nasıl sonuçlar ürettiği incelenmiştir. Araştırılan özellikler ise test durumlarının ortalama uzunluğu, test takımlarının uzunluğu ve reset miktarıdır. Rastgele oluşturulan FSM'lerin sayısı, 12 adet grafik için yaklaşık 640 adettir. Algoritmalar dört defa çalıştırılarak, grafikler test edilmiştir. Yani inceleme, sonuçların genelleştirilmesi adına yaklaşık 2560 adet rastgele FSM ile araştırılmıştır. Yöntemin verimliliği mutasyon analizi ile gösterilmiştir [10][2][3]. Mutasyon operatörleri olarak "transfer", "ekstra durum" ve "eksik durum" hataları kullanılmıştır.

Mutasyon testinin, pahalı bir test olduğu bilinmektedir. W ve UIO metotları state'lere ulaşmak için geçiş ağacını kullanmaktadır. Fakat bu durum, çok fazla giriş verisinin oluşmasına neden olmaktadır. Çalışma kapsamında geliştirilen algoritmaya da geçiş ağacı kullanılabilir; durumlara elle veri gönderebileceği için ve mutasyon testinin maliyetini azaltmak için geçiş ağacı tüm algoritmalarından çıkarılmıştır. Böylece, hatalı testin oluşması engellenmeye çalışılmıştır.

Sıfırlama sayısı ve test senaryolarının ortalama uzunluğu test takımının karakteristiğini belirler. Sıfırlama sayısı test senaryo sayısı ile aynıdır. Test senaryo sayısının fazla olması, sıfırlama sayısının da fazla olmasından dolayı maliyetlidir. Yani sıfırlama sayısı karakteristiği belirlerken, maliyeti de belirlemektedir. Burada, değeri küçük olanın sıfırlama sayısı ve dolaylı olarak da maliyeti küçüktür. Test senaryosunun ortalama uzunluğu genel olarak ne kadar fazlaysa, maliyeti o kadar düşüktür. Çünkü uzun olması durumunda sıfırlama sayısı daha düşük olacağından maliyet düşecektir. Ancak, her iki değer de başka değişkenlerin etkisinde olduğu göz önünde bulundurulmalıdır. Dolayısıyla, bu değerler tahmini sonuç verecektir.

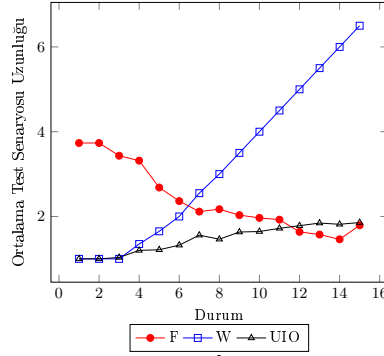
Şekil 5 6 7 8'te sadece durum ile ilgili sıfırlama, ortalama test senaryosu uzunluğu, test takımı uzunluğu ve hata yakalama oranı grafiklerine yer verilmiştir. Giriş ve çıkış ile ilgili testler sayfa sınırı gereği, bu çalışmaya koyulmamıştır.



Şekil 5: Durum sayısına göre sıfırlama sayıları

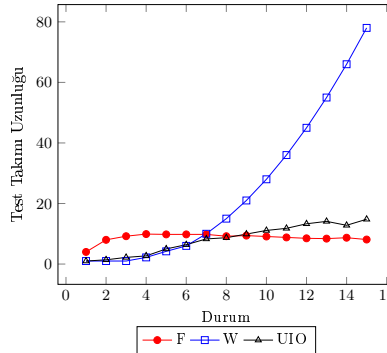
Sıfırlama ve durum grafiğinde, sıfırlama sayısı açısından W metodu doğrusal biçimde ilerlerken; F ve UIO metotlarının azalarak artan biçimde ilerlediği görülmektedir. Sıfırlama sayısının az olması, eğer test takımı uzunluğu değişimi-

yorsa, test maliyeti açısından başarıyı yüksek bir sonuç vermektedir. Şekil 5'te, F metodunun en düşük sıfırlama değerine sahip olması dolayısıyla, diğerlerine göre en yüksek başarıya ulaştığı görülmektedir.



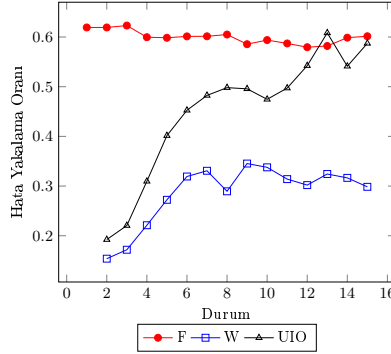
Şekil 6: Durum sayısına göre ortalama test senaryosu uzunluğu

Test senaryosu uzunluğu arttıkça sıfırlama sayısı azalacağından, test takımı uzunluğu sabit olma koşuluyla, başarı olarak değerlendirilmektedir. Durum ve ortalama test senaryosu uzunluğu grafiğinde (Şekil 6), W metodunun doğrusal biçimde arttığı görülmektedir. Diğer yandan F ve UIO metotları, düz doğrusal biçimde ilerleme göstermiştir. Bu nedenle, diğerlerine göre W metodunun başarımının daha yüksek olduğu söylenebilir.



Şekil 7: Durum sayısına göre test takımı uzunluğu

Test takımı uzunluğu maliyeti belirler. Test takımı uzunluğu ve durum grafiğine (Şekil 7) baktığımızda, W metodunun üstsel biçimde arttığı görülmektedir. Diğer yandan F ve UIO metodları, düz doğrusal ilerleme göstermiştir. Buradan hareketle, UIO ve F metodlarının maliyeti düşük; dolayısıyla başarıyı yüksektir.



Şekil 8: Durum sayısına bağlı hata yakalama oranı

Hata bulma oranı, maliyetten sonra en önemli parametrelerden biridir. Hata bulma oranı ve durum grafiğine (Şekil 8) bakıldığında, UIO ve W metodlarının azalarak artan bir yol izlediği görülmektedir. Diğer yandan F metodu, düz doğrusal bir grafik çizmektedir. F metodunun hata bulma oranı diğerlerine göre daha yüksek olduğundan, gösterdiği başarıyı da diğer metotlara oranla daha yüksektir.

5 Sonuç

FSM'ler kullanılarak test takımları yaratan metotlara alternatif olarak, daha iyi başarıyı sağlamak amacıyla geliştirilen F metodunda, ikili fonksiyonlar üzerine Fourier dönüşümü yöntemi kullanılmıştır. Fourier dönüşümünde, gürültü karakteristikleri ve çeşitli frekans bileşenlerinin fonksiyon çıkışı üzerindeki etkilerine yönelik analiz, yüksek performanslı testlerin oluşturulmasına olanak vermektedir. Matematik, bilgisayar bilimleri ve mühendislik gibi geniş bir alanda çalışılan Fourier dönüşümlerinde her bir katsayı, değişkenlerin değişimlerinin fonksiyon üzerindeki etkisini bildirmektedir. Buradan yola çıkarak, fonksiyon için etkisi en fazla olan değişken değişimleri çıkarılmıştır. Değişkenlerdeki değişimlere bakılarak geçişler seçilmiştir. Daha sonra aç gözlü algoritması ile bunlardan test takımı elde edilmiştir. Test takımları yaratan metotlara alternatif olarak, daha iyi başarıyı sağlamak amacıyla geliştirilen F metodu; karakteristiği, maliyeti ve hata bulma oranı bakımından, mutasyon analiz yöntemi ile mevcut metotlarla (UIO ve W) karşılaştırılmıştır.

Karşılaştırmalar sonucunda, sıfırlama sayısı bakımından F metodunun en düşük sıfırlama değerine sahip olması dolayısıyla, diğerlerine göre en yüksek başarıya ulaştığı görülmüştür. Diğer yandan, durum ve ortalama test senaryosu uzunluğu grafiğinde, W metodunun diğerlerine oranla başarıyı yüksek bir sonuç verdiği ortaya çıkmıştır. Test takımı uzunluğu ve durum grafiğine baktığımızda ise UIO ve F metodlarının maliyetlerinin düşük olması dolayısıyla, başarılarının yüksek olduğu görülmüştür. Maliyetten sonra en önemli parametrelerden biri olan hata bulma oranına bakıldığında ise F metodunun diğer metotlara oranla daha yüksek başarıyı gösterdiği sonucuna ulaşılmıştır. Bunların hepsi değerlendirilmiştir.

dirildiğinde, F metodunun, diğer metotlara alternatif olabilecek yüksek başarımlı sağladığını söylemek mümkündür.

Kaynaklar

1. Ainapure, B.S.: Software testing and quality assurance. Technical Publications (2009)
2. Belli, F., Beyazit, M.: A formal framework for mutation testing. In: 2010 Fourth IEEE International Conference on Secure Software Integration and Reliability Improvement. pp. 121–130. IEEE (2010)
3. Belli, F., Beyazit, M., Takagi, T., Furukawa, Z.: Model-based mutation testing using pushdown automata. IEICE TRANSACTIONS on Information and Systems 95(9), 2211–2218 (2012)
4. Broy, M., Jonsson, B., Katoen, J.P., Leucker, M., Pretschner, A.: Model-based testing of reactive systems. In: Volume 3472 of Springer LNCS. Springer (2005)
5. Damasceno, C.D.N., Masiero, P.C., Simao, A.: Evaluating test characteristics and effectiveness of fsm-based testing methods on rbac systems. In: Proceedings of the 30th Brazilian Symposium on Software Engineering. pp. 83–92. ACM (2016)
6. De Wolf, R.: A brief introduction to fourier analysis on the boolean cube. Theory of Computing, Graduate Surveys 1(1-20), 15 (2008)
7. Dorofeeva, R., El-Fakih, K., Maag, S., Cavalli, A.R., Yevtushenko, N.: Fsm-based conformance testing methods: A survey annotated with experimental evaluation. Information and Software Technology 52(12), 1286–1297 (2010)
8. Endo, A.T., Simao, A.: Evaluating test suite characteristics, cost, and effectiveness of fsm-based testing methods. Information and Software Technology 55(6), 1045–1062 (2013)
9. Gargantini, A.: 4 conformance testing. In: Model-based testing of reactive systems, pp. 87–111. Springer (2005)
10. Jia, Y., Harman, M.: An analysis and survey of the development of mutation testing. IEEE transactions on software engineering 37(5), 649–678 (2011)
11. Mathur, A.P.: Foundations of software testing, 2/e. Pearson Education India (2013)
12. O'Donnell, R.: Analysis of boolean functions. Cambridge University Press (2014)
13. Ural, H.: Formal methods for test sequence generation. Computer communications 15(5), 311–325 (1992)