

Vežbe: GUI testiranje

Teorija

Finite State Machine (FSM) i VFSM, pristupi koji su više od akademskog značaja i kada se radi manuelno.

EFG/EIG pristupi za automatsko testiranje:

Postoje 2 različita posla u razvoju softvera:

- Dizajn, implementacija i testiranje programa
- Dizajn, implementacija i testiranje korisničkog interfejsa (UI)

Pretpostavimo da UI može odvojeno da se razvija, jer zahteva određene veštine, a možda i različite tehnike od razvoja samog softvera. Dizajn je deo posla koji zahteva dobro razumevanje korisničkih zahteva. Implementacija podrazumeva znanje sa tehničkog aspekta - znanje programskog jezika, platform, itd. Testiranje zahteva dobro poznavanje oba.

Grafički korisnički interfejsi (GUI) postali su sve više popularni i uobičajni korisnički interfejsi u računarskim sistemima. Sa druge strane, testiranje GUI je jako težak i izazovan zadatak iz mnogih razloga: prvo, ulazni prostor poseduje veliki, potencijalno neograničeni broj kombinacija inputa i događajima koji se javljaju kao output sistema; spoljni događaji mogu da interaguju sa tim inputima. Drugo, čak i jednostavan GUI poseduje veliki broj stanja, takođe zbog inputa. Poslednje, mnoge složene zavinosti mogu postojati između različitih stanja GUI sistema i/ili između stanja i inputa.

Test slučajevi uglavnom zahtevaju određivanje smislenih test ulaza i izlaza sistema očekivanih za ove ulaze. Shodno tome, za generisanje test slučajeva za GUI, moraju se identifikovati objekti testa i test ciljevi. Test objekti su ulazni elementi, npr. ekrani, prozori, ikonice, meniji, pointeri, komande, funkcijski tasteri, itd. Cilj testa je da generiše očekivano ponašanje sistema (željeni događaj) kao izlaz pomoću dobro definisanih ulaza testa. U širem smislu, test objekat je softver koji se testira (eng. Softver under Test, SUT). Cilj testa je da dobije poverenje SUT. Robusni sistemi imaju i dobar mehanizam rukovanja sa izuzecima.

Zadatak 1 - Konstruisanje FSM za GUI

FSM se može predstaviti kao:

- skup ulaza, skup izlaza i skup stanja
- izlazna funkcija koja mapira parove ulaza i stanja u odnosu na izlaze
- funkcija za sledeće stanje koja mapira parove ulaza i stanja do narednih stanja

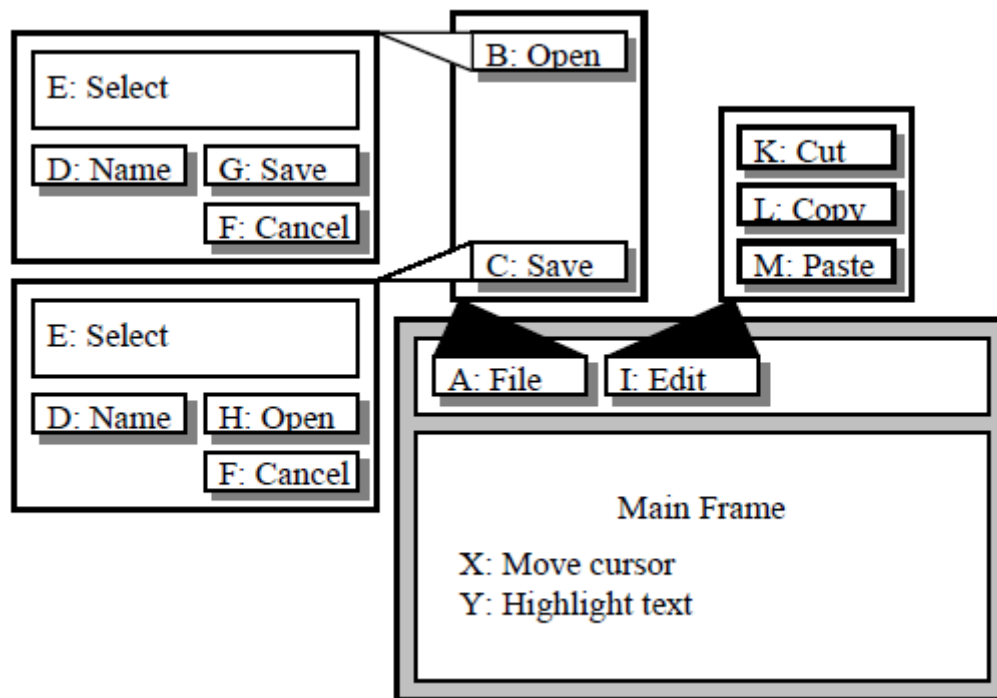
Za predstavljanje GUI mi ćemo tumačiti elemente FSA:

- Ulazni skup: identifikovati objekte koji mogu biti shvaćeni i kontrolisani od strane ulazno-izlaznih uređaja, elementi tzv. WIMP (Windows, Icons, Menus, and Pointers)
- Izlazni skup ima dve podgrupe:
 - o Željeni događaji - ishodi koje korisnik želi da ima, ispravn tj. prave odgovore
 - o Neželjeni događaji - ishodi koje korisnik ne želi da ima, neispravan rezultat ili nepčekivani rezultat, koji je iznenađenje za korisnika

Uvešćemo sledeće pretpostavke:

- Kao sinonime koristićemo FSA i STD (state transition diagram)
- STD su usmereni grafovi, imaju ulazni čvor i izlazni čvor i bar jednu granu od ulaza do izlaza.
- Izlazi su zapostavljeni u smislu Moore automata
- Mi ćemo spojiti ulaze i stanja, dodeljujući ih temenima STD u FSA
- Funkcija za sledeće stanje će se tumačiti shodno tome

Neka je dat mali deo programa MS WordPad.



Finite-State Machine

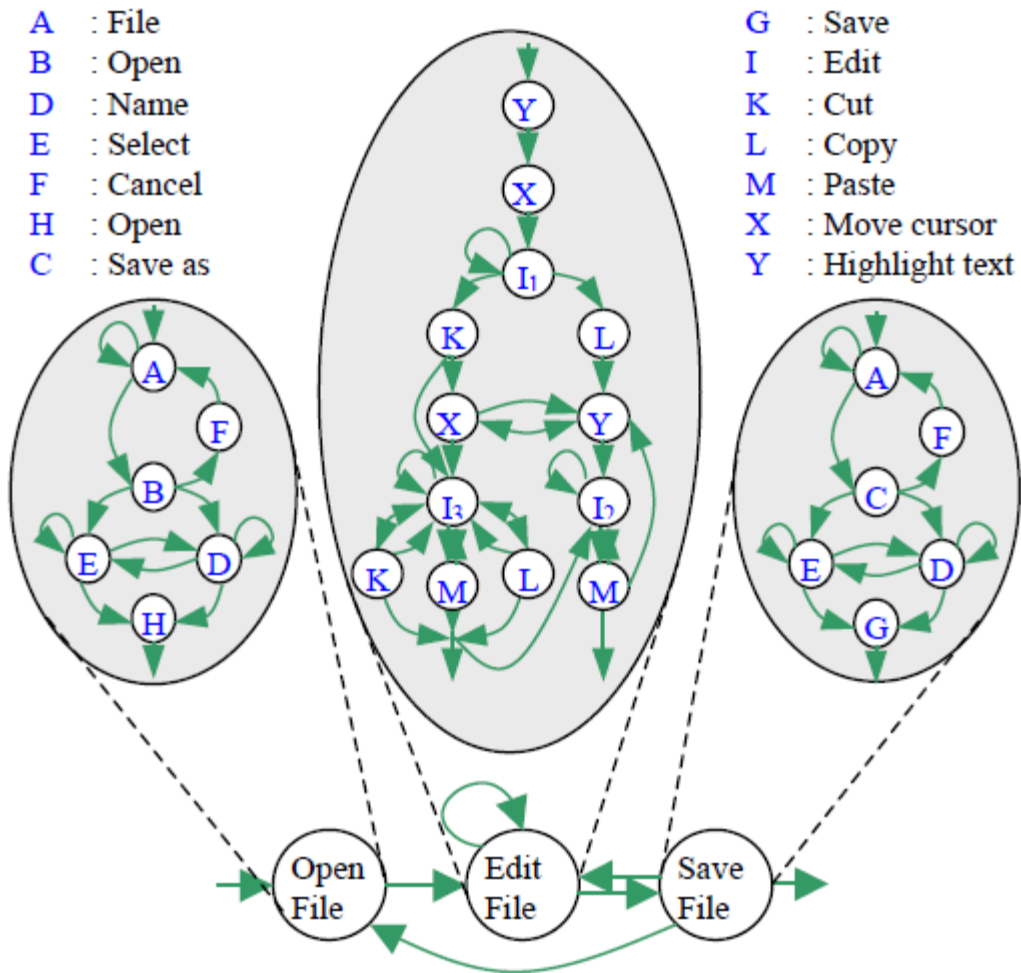
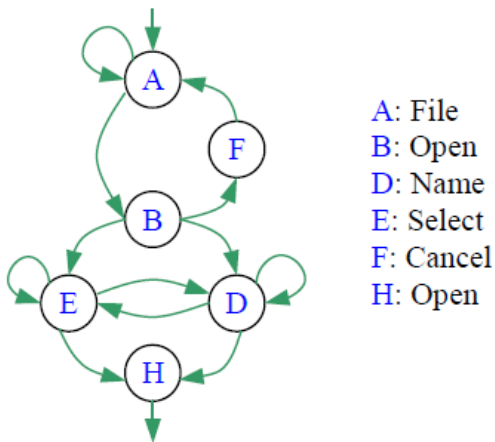


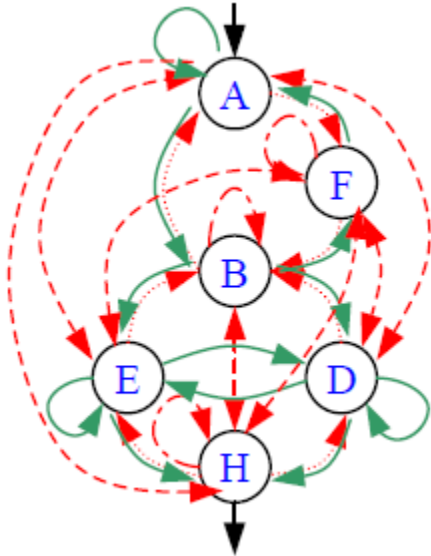
Fig. 2: Fig. 1 presented as a Finite-State Machine

Podgraf za operaciju OPEN fajla:



Sub-Graph	IPs
Open File	AA, AB, BD, BE, BF, EH, FA, ED, EE, DD, DE, DH

Completed FSA za operaciju Open file:



U ovom grafu, prema onome što ste naučili sa predavanja dodate su sledeće veze:

- veze suprotnog smera, gde već postoje veze (linije sa tačkicama)
- veze koje označavaju petlju na dati čvor (crtica-tačkica linija)
- veze između svih čvorova gde ne postoje veze (linije sa crticama)

Sub-Graph	FIPs
Open File	AD, AE, AF, AH, BA, BB, BH, DA, DB, EA, EB, FB, FF, HA, HB, HD, HE, HH

Fig. 4a: *The set of FIPs (Faulty Interaction Pairs)*

Sub-Graph	FCISs
Open File	AD, AE, AF, AH, ABA, ABB, ABH, ABDA, ABDB, ABEA, ABEB, ABFB, ABFF, AB(E+D)HA, AB(E+D)HB, AB(E+D)HD, AB(E+D)HE

Fig. 4b: *The set of FCISs (Faulty Complete Interaction Sequences)*

Zadatak 2 – Konstruisanje EFG za GUI

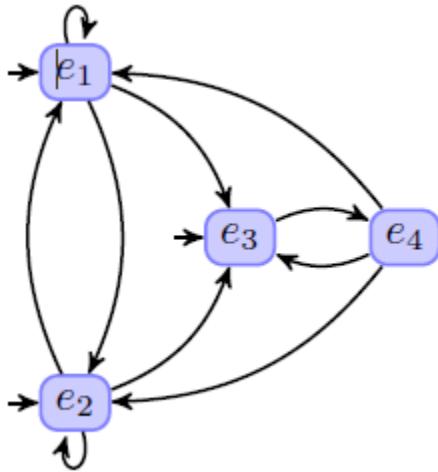
EFG (Event Flow Graph)

ESG (Event Sequence Graph)

Neka je dat primer sledeće aplikacije:



Konstruisaćemo EFG za GUI:



Sekvenca događaja dobijenih iz EFG:

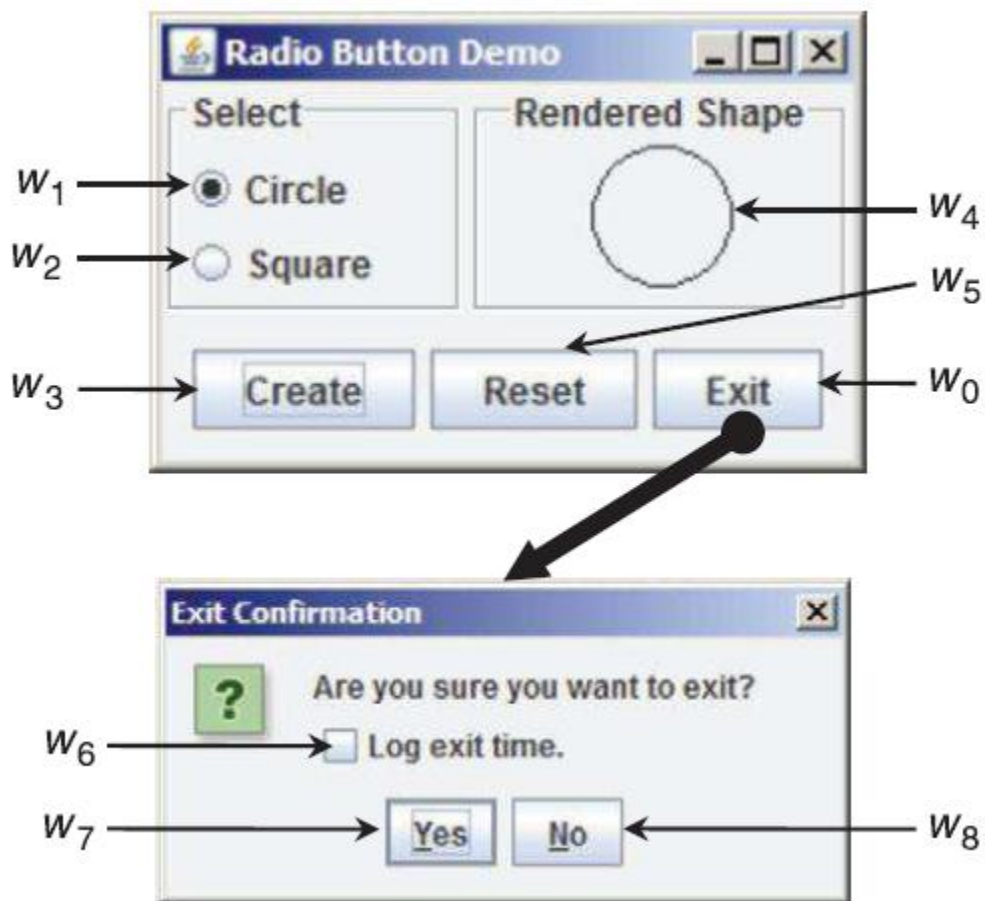
$$\begin{array}{ll}
 s_1 = \langle e_1, e_1 \rangle & s_6 = \langle e_2, e_3 \rangle \\
 s_2 = \langle e_1, e_2 \rangle & s_7 = \langle e_3, e_4 \rangle \\
 s_3 = \langle e_1, e_3 \rangle & s_8 = \langle e_4, e_1 \rangle \\
 s_4 = \langle e_2, e_1 \rangle & s_9 = \langle e_4, e_2 \rangle \\
 s_5 = \langle e_2, e_2 \rangle & s_{10} = \langle e_4, e_3 \rangle
 \end{array}$$

Daljim razvijanjem sekvenci dobijamo:

$$\begin{array}{ll}
 t_1 = \langle e_1, e_1 \rangle & t_6 = \langle e_2, e_3 \rangle \\
 t_2 = \langle e_1, e_2 \rangle & t_7 = \langle e_3, e_4 \rangle \\
 t_3 = \langle e_1, e_3 \rangle & t_8 = \langle e_3, e_4, e_1 \rangle \\
 t_4 = \langle e_2, e_1 \rangle & t_9 = \langle e_3, e_4, e_2 \rangle \\
 t_5 = \langle e_2, e_2 \rangle & t_{10} = \langle e_3, e_4, e_3 \rangle
 \end{array}$$

Zadatak 3

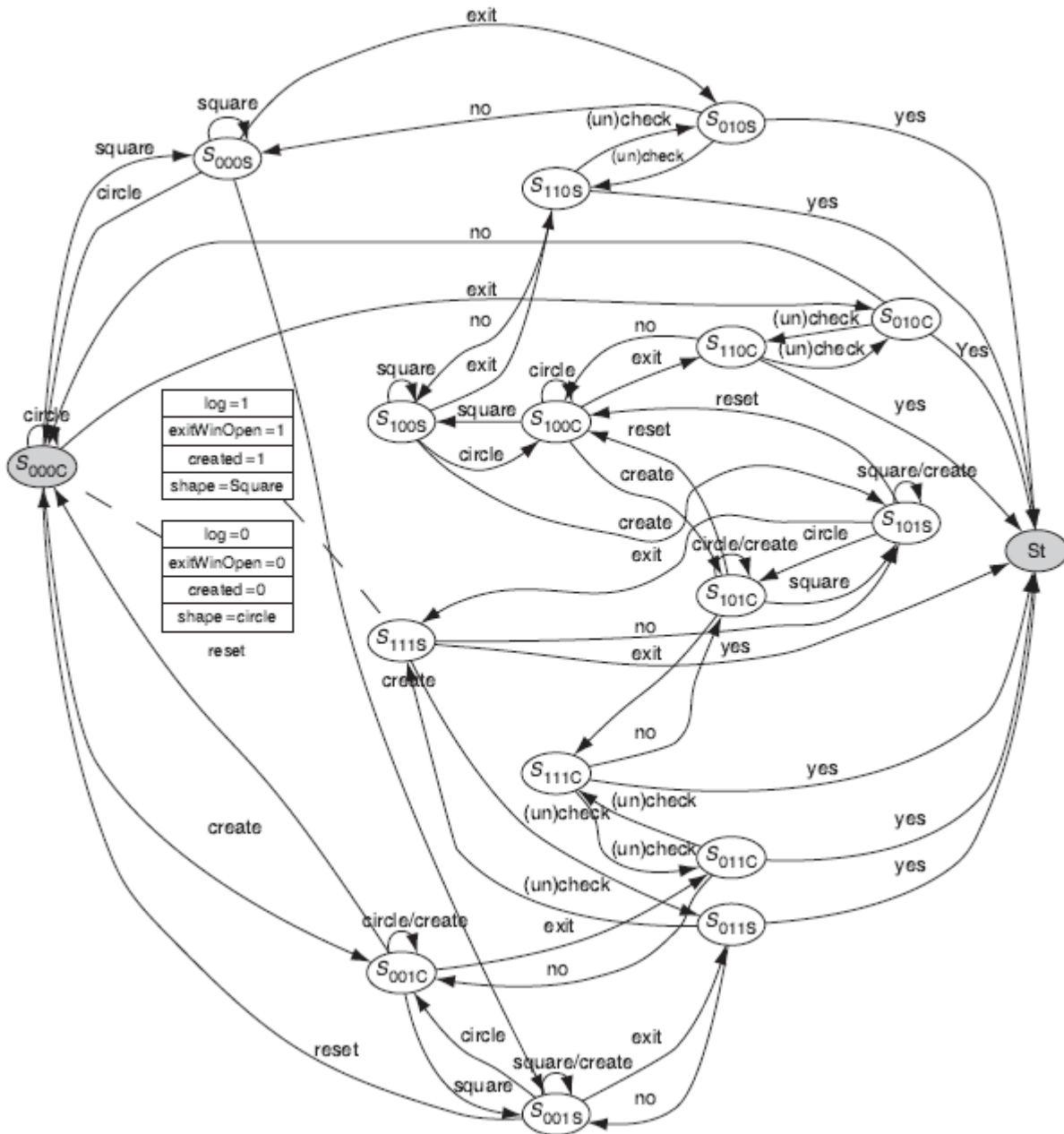
Neka je data sledeća aplikacija sa slike. Testirati opciju Radio Button ovog GUI koristeći FSM I VFSM.



Događaji koji se dobijaju klikom na određeni element:

Element	Naziv događaja
w1	circle
w2	square
w3	create
w4	reset
w5	exit
w6	(un)check
w7	yes
w8	no

FSM za Radio Button:



Test primer može biti (square, circle, create, exit, (un)check, yes) koji prolazi kroz stanja S000S, S000C, S001C, S011C, S111C, St.

VFSM - Variable Finite State Machine

Modifikacija FSM gde je svako stanje predstavljeno kao vektor sa 3 stanja. U vektoru se pamte da li je potrebno pamti log, da li je prozor za potvrdu izlaza (Exit) otvoren i koji je tip geometrijskog oblika koji je odabran.

