Jedan pristup rasplinutom testiranju softvera

(II deo)

Sadržaj

- Uvod
- Principi crne i bele kutije
- Fazi testiranje softvera
- Primena genetskog algoritma u fazi testiranju
- Definisanje hromozoma pomoću kontekstno nezavisne gramatike
- Markovljevi lanci
- Definisanje funkcije prilagođenosti preko Markovljevih lanaca
- Završna razmatranja

Principi crne i bele kutije

Šta je testiranje softvera (formalno?)

- Testiranje softvera podrazumeva:
  - analizu softvera,
  - otkrivanje razlike između očekivanih i stvarnih uslova rada.
  - ocenjivanje karakteristika softvera.
- Testiranje softvera uključuje:
  - **verifikaciju**, tj. utvrđivanje da li se proces razvoja softvera odvija u skladu sa zadatim uslovima,
  - **validaciju**, tj. utvrđivanje da li softver odgovara korisničkim zahtevima.
- Drugim rečima,
  - **verifikacija** je utvrđivanje da li razvijamo softver **na odgovarajući način**.
  - **validacija** je utvrđivanje da li razvijamo **odgovarajući softver**.

Principi crne i bele kutije

Osnovne strategije testiranja softvera.

- Testiranje softvera zasnovano na principu **bele kutije**.
- Testiranje softvera zasnovano na principu **crne kutije**.

Principi crne i bele kutije

Princip bele kutije.
- Podrazumeva dostupnost izvornog koda programa.
- Zasniva se na manuelnoj, statičkoj analizi izvornog koda programa.
- Prednost:
  - Ljudskom analizom se može otkriti više (suptilnih) grešaka nego upotrebom alata za automatizovano testiranje.
- Nedostaci:
  - Vremenski zahtevno (tj. skupo).
  - Ponekad je teško utvrditi da li potencijalna slabost softvera stvarno zavisi od korisničkog unosa prilikom izvršavanja programa.

Princip crne kutije.

- Dostupna je specifikacija softvera, ali ne i izvorni kod.
- Analizira se odziv sistema na zadate ulazne podatke.
- Jedan od zadataka testiranja je da se utvrdi da li stvarni odziv sistema odgovara očekivanom (tj., specificiranom).
- Prednosti:
  - Lako se automatizuje (tj. jeftino).
  - Slabosti softvera koje se otkriju na ovaj način sigurno zavise od korisničkog unosa prilikom izvršavanja programa (poznato je i koji unosi ih uzrokuju).
- Nedostatak:
  - Slučajni izbor ulaznih podataka za testiranje može da dovede do toga da neke slabosti softvera ne budu otkrivene.

Principi crne i bele kutije

Konceptualne razlike.

- Testiranje zasnovano na principu **bele kutije** se uobičajeno koristi za **verifikaciju softvera**.
- Testiranje zasnovano na principu **crne kutije** se uobičajeno koristi za **validaciju softvera**.

Fazi testiranje softvera

- U nastavku predavanja razmatramo jedan pristup fazi testiranju softvera koji se bazira na:
  - genetskom algoritmu,
  - kontekstno nezavisnim gramatikama i
  - Markovljevim lancima.
- Fazi testiranje posmatramo u kontekstu **napada na softver**.

Fazi testiranje softvera

- **Šta je fazi testiranje softvera**?
  - Fazi testiranje softvera se zasniva na principu **crne kutije**.
  - Podrazumeva **zadavanje „nekorektnih“ ulaznih podataka** softveru sa namerom da se izazove njegovo nepravilno funkcionisanje.
  - Prednost:
    - automatizovani napad na softver – ulazni podaci se generišu na slučajni ili pseudoslučajni način.
  - Nedostaci:
    - prilikom generisanja novih ulaznih podataka, ne uzima se u obzir „kvalitet“ prethodno generisanih ulaznih podataka.
    - ne postoji mera napretka napada – ili smo uspeli da „oborimo“ softver, ili nismo,
    - nedeterminističko vreme izvršavanja – ne možemo da procenimo vreme potrebno za detekciju greške u softveru.

Cilj.

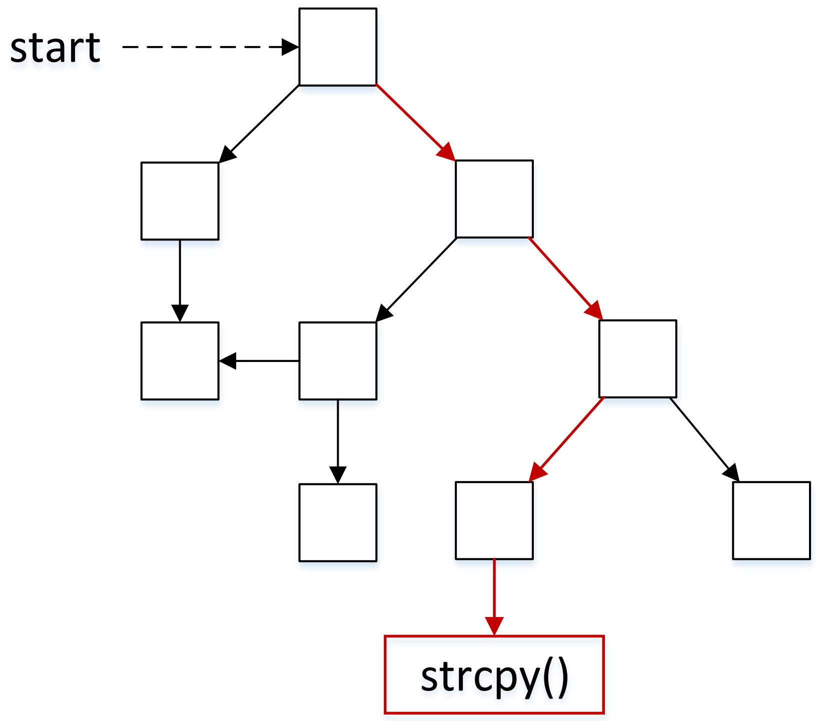
- Pristup fazi testiranju koji prevazilazi ove nedostatke bi trebalo da:
  - **„uči“** iz prethodno generisanih ulaznih podataka da bi unapredio generisanje budućih podataka,
  - **poveća verovatnoću** da se greška u softveru detektuje u razumnom vremenu.
  - **ostane automatizovan** u što većoj meri.

„Učenje“.

- Putanja izvršavanja programa zavisi od:
  - ulaznih podataka,
  - strukture programa.
- **Cilj** „učenja“ je da se uspostavi veza između ulaznih podataka i putanja izvršavanja programa.
- **Osnovna ideja**.
  - Tražimo ulazne podatke koji uzrokuju putanju izvršavanja programa koja će dovesti do bloka za koji je poznato da je podložan nekorektnom funkcionisanju za određene ulazne podatke.

Primer.

- Na slici je data hipotetička struktura izvršavanja izvornog koda programa (putanja koja dovodi do sporne funkcije je obeležena crvenom bojom).



Primena genetskog algoritma

Genetski algoritam.

- Posmatrani pristup fazi testiranju se zasniva na genetskom algoritmu.
- Potrebno je definisati:
  - **Hromozom** (kako se predstavljaju ulazni podaci?)
  - **Funkciju prilagođenosti** (kako se procenjuje kvalitet ulaznih podataka?)
- U ovom pristupu:
  - Hromozom se definiše pomoću kontekstno nezavisne gramatike.
  - Funkcija prilagođenosti se definiše pomoću Markovljevih lanaca.

Definisanje hromozoma pomoću kontekstno nezavisne gramatike

Osnovna idehja za definisanje hromozoma.

- U ovom primeru, hromozom predstavlja ulazne podatke.
- Jezik koji definiše ulazne podatke je definisan kontekstno nezavisnom gramatikom.
- Primer gramatike:

$$
\begin{array}{lcccccc}
S \rightarrow & sAs & | & xBx & | & m \\
A \rightarrow & bBb & | & B & & \\
B \rightarrow & aAa & | & C & | & AB \\
C \rightarrow & c & | & d & | & e
\end{array}
$$

- Objasnimo na primeru kako se koduje hromozom koji predstavlja ulazne podatke *xabdbax*.

Definisanje hromozoma pomoću kontekstno nezavisne gramatike

Označavanje produkcionih pravila.

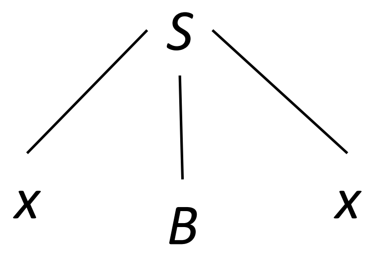
- Označimo delove produkcionih pravila brojevima 0, 1 i 2, kao što je prikazano.

$$
\begin{array}{lccccc}
 & 0 & & 1 & & 2 \\
S \rightarrow & sAs & | & xBx & | & m \\
A \rightarrow & bBb & | & B & & \\
B \rightarrow & aAa & | & C & | & AB \\
C \rightarrow & c & | & d & | & e
\end{array}
$$

Definisanje hromozoma pomoću kontekstno nezavisne gramatike

Sintaksno stablo za string *xabdbax*.

	0		1		2
$S \rightarrow$	sAs	$\vert$	xBx	$\vert$	m
$A \rightarrow$	bBb	$\vert$	B		
$B \rightarrow$	aAa	$\vert$	C	$\vert$	AB
$C \rightarrow$	c	$\vert$	d	$\vert$	e

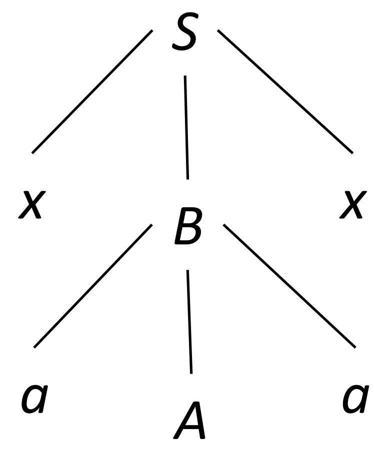


- Hromozom: 1 …

Definisanje hromozoma pomoću kontekstno nezavisne gramatike

Sintaksno stablo za string *xabdbax*.

	0		1		2
$S \rightarrow$	sAs	|	xBx	|	m
$A \rightarrow$	bBb	|	B		
$B \rightarrow$	aAa	|	C	|	AB
$C \rightarrow$	c	|	d	|	e



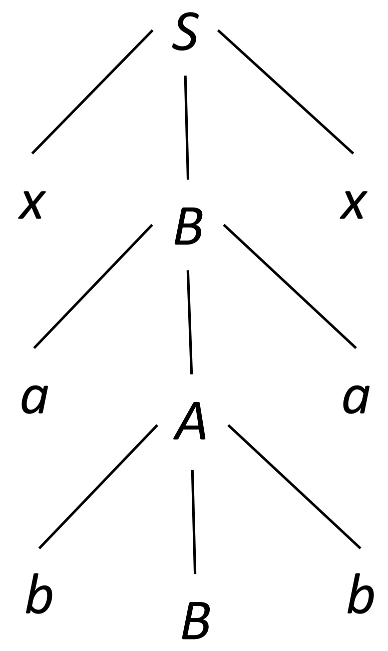
- Hromozom: 1 0 …

Definisanje hromozoma pomoću kontekstno nezavisne gramatike

Sintaksno stablo za string *xabdbax*.

	0		1		2
$S \rightarrow$	sAs	|	xBx	|	m
$A \rightarrow$	bBb	|	B		
$B \rightarrow$	aAa	|	C	|	AB
$C \rightarrow$	c	|	d	|	e

- Hromozom: 1 0 0 …

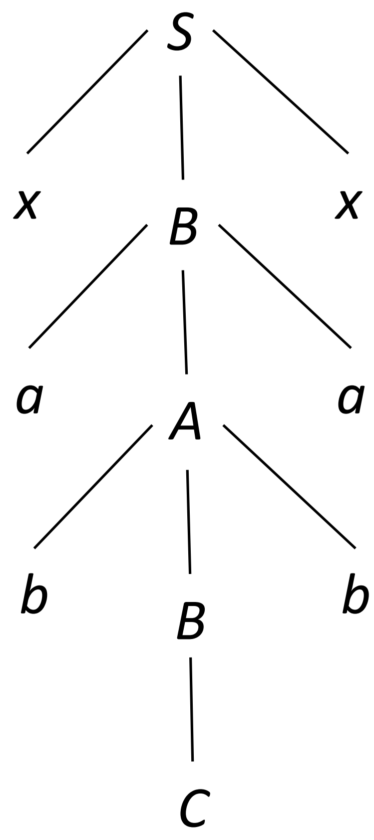


Definisanje hromozoma pomoću kontekstno nezavisne gramatike

Sintaksno stablo za string *xabdbax*.

$$\begin{array}{lccccc}
 & 0 & & \underline{1} & & 2 \\
S \rightarrow & sAs & | & xBx & | & m \\
A \rightarrow & bBb & | & B & & \\
B \rightarrow & aAa & | & C & | & AB \\
C \rightarrow & c & | & d & | & e
\end{array}$$

- Hromozom: 1 0 0 <u>1</u> …

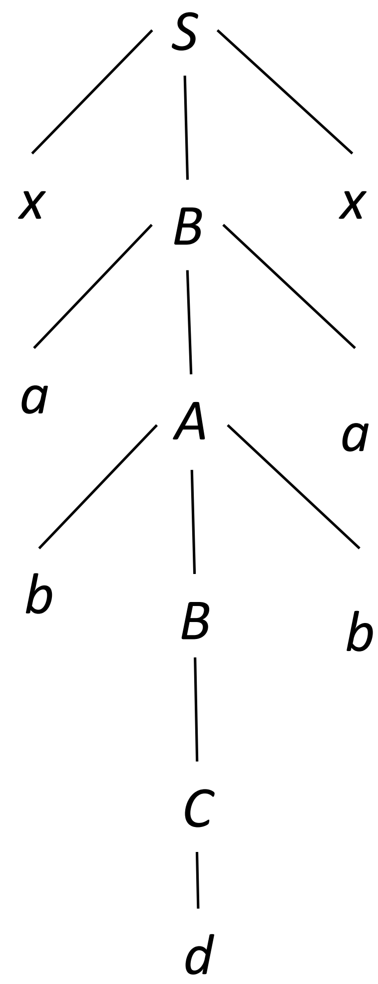


Definisanje hromozoma pomoću kontekstno nezavisne gramatike

Sintaksno stablo za string *xabdbax*.

	0	1	2
$S \rightarrow$	sAs	xBx	m
$A \rightarrow$	bBb	B	
$B \rightarrow$	aAa	C	AB
$C \rightarrow$	c	d	e

- Hromozom: 1 0 0 1 1.
- Ovaj hromozom predstavlja ulazne podatke *xabdbax*.



Markovljevo svojstvo.

- **Markova svojstvo** znači da pored datog trenutnog stanja, buduće stanje sistema ne zavisi od prošlih.
  - Drugim rečima to znači da opis sadašnjosti u potpunosti sadrži informaciju koja može uticati na buduće stanje procesa.
  - Dakle, pored date sadašnjosti, **budućnost ne zavisi od prošlosti**.
- U svakom trenutku sistem, na osnovu date raspodele slučajne promenljive, može promeniti stanje, ili ostati u istom.
  - Promene stanja nazivamo **prelazima**.
  - Verovatnoće koje se odnose na različite promene stanja nazivamo **verovatnoćama prelaza**.

Primer.

- Slučajna šetnja po brojnoj osi gde se, pri svakom koraku, pozicija menja za 1 (u jednom ili drugom smeru **jednako verovatno**).
- Sa svake pozicije postoje dva moguća prelaza: na sledeći ili na prethodni ceo broj.
- Verovatnoće prelaza zavise samo od trenutnog stanja, a ne od načina kako se do njega došlo.
- Na primer, ako je trenutna pozicija -3, prelaz u -2 ima verovatnoću 0,5, bez obzira na prethodne pozicije.

Formalna definicija.

- **Markovljev lanac** u matematici označava diskretan Markovljev slučajni proces.
- Neka su:
  - $x_1, x_2, \ldots, x_n$ – stanja procesa
  - $S = \{x_1, x_2, \ldots, x_n\}$ – prebrojivi skup stanja (prostor stanja lanaca).
  - $p(X_{n+1} = x_{n+1} | X_n = x_n)$ – verovatnoća prelaza iz stanja x_n u stanje x_{n+1}.
- Markovljev lanac je niz slučajnih promenljivih $X_1, X_2, \ldots X_n$ s Markovljevim svojstvom: $p(X_{n+1} = x | X_n = x_n, \ldots, X_1 = x_1) = p(X_{n+1} = x | X_n = x_n)$.

Predstavljanje pomoću grafova.

- Markovljeve lance možemo prikazati i **usmerenim grafovima**.
    - Čvorovi grafa su pojedina stanja.
    - Vrednosti napisane na granama predstavljaju verovatnoće prelaza (u odgovarajućem smeru).

Verovatnoća prelaza.

- U pojedinim situacijama je potrebno da se na osnovu sadašnjeg stanja sistema predvidi u kom će se narednom stanju nalaziti sistem.
- Na osnovu sadašnjeg stanja sistema:
  - ne možemo sigurno da predvidimo naredno stanje,
  - možemo da odredimo verovatnoću da će se sistem prilikom sledećeg posmatranja nalaziti u nekom određenom stanju.

Primer.

- Na železničkoj pruzi nalazi se semafor sa upaljenim ili crvenim (C) ili zelenim (Z) svetlom
- U ovom primeru železnička pruga i semafor predstavljaju jedan sistem.
- Pretpostavimo da smo u nekom vremenskom intervalu posmatrali u kom se stanju nalazi sistem i da smo zapazili sledeći niz:
- C, C, Z, Z, Z, C, Z, Z, C, C, C, Z, Z, Z, C, Z (*)

Primer.

- C, C, Z, Z, Z, C, Z, Z, C, C, C, Z, Z, Z, C, Z (*)
- Pitanja:

1. Ako je sistem sada u stanju C, kolika je verovatnoća da će pri sledećem posmatranju biti u stanju C?
2. Ako je sistem sada u stanju C, kolika je verovatnoća da će pri sledećem posmatranju biti u stanju Z?
3. Ako je sistem sada u stanju Z, kolika je verovatnoća da će pri sledećem posmatranju biti u stanju C?
4. Ako je sistem sada u stanju Z, kolika je verovatnoća da će pri sledećem posmatranju biti u stanju Z?

Primer.

- C, C, Z, Z, Z, C, Z, Z, C, C, C, Z, Z, Z, C, Z (*)
- Iz niza (*) se vidi da je sistem u stanju C bio 7 puta.
    - U 3 slučaja je ostao u stanju C.
    - U 4 slučaja je prešao u stanje Z.
- Iz niza (*) se vidi da je sistem u stanju Z bio 8 puta.
    - U 5 slučajeva je ostao u stanju Z.
    - U 3 slučaja je prešao u stanje C.
- Zaključujemo da je:
    1. Verovatnoća $p(X_{n+1} = \mathrm{C} | X_n = \mathrm{C}) = {}^3/_7$
    2. Verovatnoća $p(X_{n+1} = \mathrm{Z} | X_n = \mathrm{C}) = {}^4/_7$
    3. Verovatnoća $p(X_{n+1} = \mathrm{C} | X_n = \mathrm{Z}) = {}^3/_8$
    4. Verovatnoća $p(X_{n+1} = \mathrm{Z} | X_n = \mathrm{Z}) = {}^5/_8$

Markovljevi lanci reda *m*.

- Markovljevi lanci reda *m* (sa **memorijom** *m*) su oni za koje važi (za konačno *m*):
- $p(X_n = x_n | X_{n-1} = x_{n-1}, X_{n-2} = x_{n-2}, \dots, X_1 = x_1) = p(X_n = x_n | X_{n-1} = x_{n-1}, X_{n-2} = x_{n-2}, \dots, X_{n-m} = x_{n-m})$.
- Drugim rečima, buduće stanje zavisi od *m* prеđašnjih.
- U slučaju *m*=1 radi se o **prostom** Markovljevom lancu.

Homogeni Markovljevi lanci.

- Markovljev lanac je **homogen** (sa stacionarnim verovatnoćama stanja) ako verovatnoće prelaza ne zavise od vremena, tj.: $p(X_{n+1} = x_{n+1} | X_n = x_n) = p_{ij}$, nezavisno od *n*.

Matrica prelaza (na primeru vremenske prognoze).

- Matrica prelaza *P* je matrica čiji je (*i*, *j*)-ti element jednak $p_{ij} = p(X_{n+1} = j | X_n = i)$.
- Pretpostavimo pojednostavljeni model predviđanja vremena u kom vremenska prognoza u nekom danu zavisi samo od vremense prognoze prethodnog dana.
  - Neka je *S* = {1, 2}, gde stanja označavaju: 1 – kiša, 2 – nije kiša.
  - Ako kiša pada danas, onda je verovatnoća da će padati i sutra jednaka α.
  - Ako kiša nije padala danas, onda je verovatnoća da neće padati ni sutra jednaka β.
  - Matrica prelaza je data sa: $P = \begin{bmatrix} \alpha & 1-\alpha \\ \beta & 1-\beta \end{bmatrix}$.

Primer 1.

- Čovek putuje na posao vozom ili kolima, pri čemu:
  - nikad ne putuje dva dana uzastopno vozom,
  - sa podjednakom veorvatnoćom putuje vozom ili kolima ako je prethodnog dana putovao kolima.
- Pretpostavimo da je prvog dana otišao na posao kolima ako je prilikom bacanja kocke pala šestica.
- Na osnovu datih podataka formirati matricu prelaza i odrediti verovatnoće putovanja vozom i kolima petog dana.

Rešenje primera 1.

- Postoje dva stanja: putovanje vozom (1) i putovanje kolima (2).
- Elementi matrice *P*, p_{ij} su verovatnoće prelaza iz stanja *i* u stanje *j*, tj. verovatnoće da ako se jednog dana na posao ide prevozom *i*, narednog će se koristiti prevoz *j*.
- Matrica prelaza u jednom koraku data je na sledeći način: $P = \begin{bmatrix} 0 & 1 \\ {}^1/_2 & {}^1/_2 \end{bmatrix}$.
- Zašto?
  - Čovek se ne vozi uzastopno dva dana vozom.
  - Narednog dana će sigurno ići autom, pa je p_{12}=1.
  - U drugoj vrsti matrice *P*, ako je danas za prevoz korišćen auto, sutra je jednako verovatno da će se ići autom ili vozom, pa su elementi p_{21}=p_{22}=${}^1/_2$.

Rešenje primera 1 (nastavak).

- Početni raspored verovatnoća dat je vektorom: $p_0 = \begin{bmatrix} \frac{5}{6} & \frac{1}{6} \end{bmatrix}$.
    - Pri tome je ${}^{1}/{}_{6}$ verovatnoća da pri jednom bacanju kocke padne šestica, tj. da čovek danas na posao ide kolima.
    - Ako šestica ne padne, što se dešava sa verovatnoćom $1 - {}^{1}/{}_{6} = {}^{5}/{}_{6}$, na posao će ići vozom.

Rešenje primera 1 (nastavak).

- Verovatnoće prelaza za 4 dana date su matricom: $P^4 = P^2P^2$, gde je $P^2 = PP$.
  - $P^2 = PP = \begin{bmatrix} 0 & 1 \\ 1/2 & 1/2 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1/2 & 1/2 \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 \\ 1/4 & 3/4 \end{bmatrix}.$
  - $P^4 = P^2P^2 = \begin{bmatrix} 1/2 & 1/2 \\ 1/4 & 3/4 \end{bmatrix} \begin{bmatrix} 1/2 & 1/2 \\ 1/4 & 3/4 \end{bmatrix} = \begin{bmatrix} 3/8 & 5/8 \\ 5/16 & 11/16 \end{bmatrix}.$
- Konačno:
  - $p = p_0P^4 = \begin{bmatrix} \frac{5}{6} & \frac{1}{6} \end{bmatrix} \begin{bmatrix} 3/8 & 5/8 \\ 5/16 & 11/16 \end{bmatrix} = \begin{bmatrix} \frac{35}{96} & \frac{61}{96} \end{bmatrix}.$
- Verovatnoća da čovek petog dana putuje vozom je $\frac{61}{96}$, bez obzira kako je danas otišao na posao.

Primer 2.

- Tri dečaka bacaju loptu jedan drugom.
  - Dečak A će uvek baciti loptu dečaku B.
  - Dečak B će uvek baciti loptu dečaku C.
  - Dečak C će sa podejdnakom verovatnoćom bacati loptu dečacima B i A.
- Pretpostavimo da je u startu lopta bila u rukama dečaka C.
- Izračunati verovatnoće posedovanja lopte posle trećeg bacanja.

Rešenje primera 2.

- Postoje tri stanja: lopta je kod dečaka A (1), kod dečaka B (2) i kod dečaka C (3).
- Matrica prelaza u jednom koraku (dobacivanju lopte) je: $P = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0{,}5 & 0{,}5 & 0 \end{bmatrix}$, gde su po vrstama date verovatnoće bacanja lopte za dečaka A, B i C, respektivno.
- Incijalni vektor verovatnoća: $p_0 = [0 \quad 0 \quad 1]$.
- Situacija:
  - Nakon prvog bacanja: $p_1 = p_0 P = [0{,}5 \quad 0{,}5 \quad 0]$.
  - Nakon drugog bacanja: $p_2 = p_1 P = [0 \quad 0{,}5 \quad 0{,}5]$.
  - Nakon trećeg bacanja: $p_3 = p_2 P = [0{,}25 \quad 0{,}25 \quad 0{,}5]$.
- Napomena. Do istog rezultata smo mogli da dođemo koristeći zapis prethodnog zadatka: $p_3 = p_0 P^4$.

Primer 3.

- Pretpostavimo da vremenski uslovi na današnji dan zavise od vremenskih uslova prethodna dva dana.
  - Ako je poslednja dva dana padala kiša, i narednog dana će padati sa verovatnoćom 0,7.
  - Ako je kiša padala danas, ali ne i juče, onda će sutra padati sa verovatnoćom 0,5.
  - Ako je kiša padala juče, a danas ne, onda će sutra padati sa verovatnoćom 0,4.
  - Ako poslednja dva dana nije bilo kiše, naredni dan će biti kišni sa verovatnoćom 0,2.
- Ako je kiša padala u ponedeljak i utorak, kolika je verovatnoća da će padati i u četvrtak?

Rešenje primera 3.

- Postoje četiri različita stanja vremenskog sistema:
  - kiša je padala poslednja dva dana (1),
  - kiša je padala juče, ali ne i dan pre (2),
  - kiša nije padala juče, ali je padala dan pre (3),
  - poslednja dva dana kiša nije padala (4).
- U ovom slučaju, p_{11}=0,7, jer ako je kiša padala poslednja dva dana (stanje 1), padaće i sutra (opet stanje 1, danas-sutra) sa verovatnoćom 0.7.
- Prelaz iz stanja 1 u stanje 2 nije moguć, jer ako je kiša padala poslednja dva dana, a sutra ne pada, to znači da je narednog dana vreme u stanju 3, pa je p_{12}=0, a p_{13}=0,3. Iz istog razloga je p_{14}=0,3.
- Sličnim razmatranjima dobijaju se i preostale verovatnoće prelaza.

Rešenje primera 3 (nastavak).

- Matrica prelaza data je sa: $P = \begin{bmatrix} 0,7 & 0 & 0,3 & 0 \\ 0,5 & 0 & 0,5 & 0 \\ 0 & 0,4 & 0 & 0,6 \\ 0 & 0,2 & 0 & 0,8 \end{bmatrix}$.
- Dvodnevni prelaz dobija se računanjem matrice: $P^2 = PP = \begin{bmatrix} 0,49 & 0,12 & 0,21 & 0,18 \\ 0,35 & 0,20 & 0,15 & 0,30 \\ 0,20 & 0,12 & 0,20 & 0,48 \\ 0,10 & 0,16 & 0,10 & 0,64 \end{bmatrix}$.
- Verovatnoće koje nas interesuju su p_{11}^2 i p_{12}^2, jer ako je dva dana uzastopno padala kiša (stanje 1), dva dana kasnije će padati, ako je sistem u stanju 1 ili 2.
- Dakle, četvrtak je kišni dan sa verovatnoćom 0,49+0,12=0,61.

Definisanje funkcije prilagođenosti preko Markovljevih lanaca

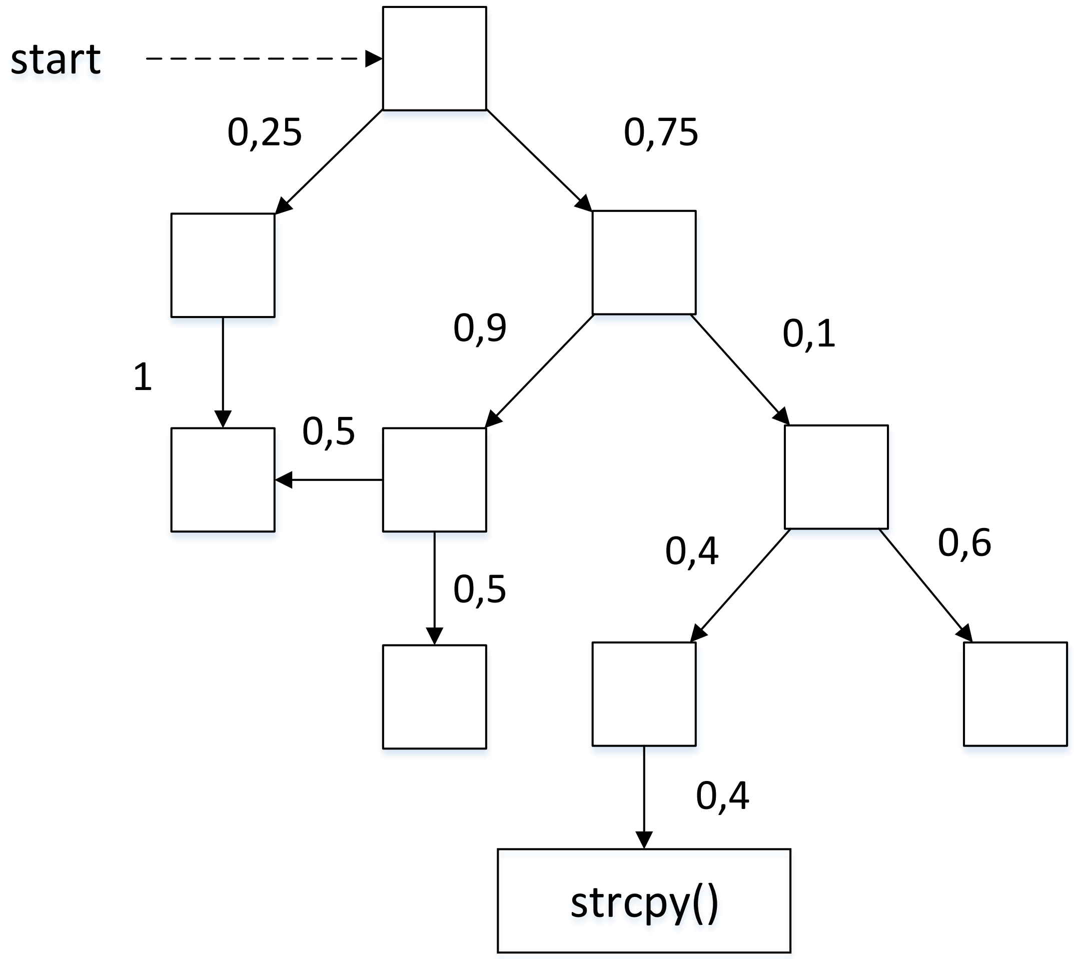
Osnovna ideja za definisanje funkcije prilagođenosti.

- Posmatramo odziv programa, i orijentišemo se na strukturi izvršavanja programa.
- Cilj je da ulazni podaci maksimizuju deo izvornog koda koji je podvrgnut napadu.
- Ovi podaci se mogu opisati na dva načina:
  - Uzrokuju ponašanje softvera koje nije prethodno zabeleženo.
  - Uzrokuju putanje izvršavanja programa koje dovode do blokova koji ranije nisu posećeni.
- Kako se kvantifikuje posećenost blokova?

Definisanje funkcije prilagođenosti preko Markovljevih lanaca

Izvršavanje programa kao Markovljev proces.

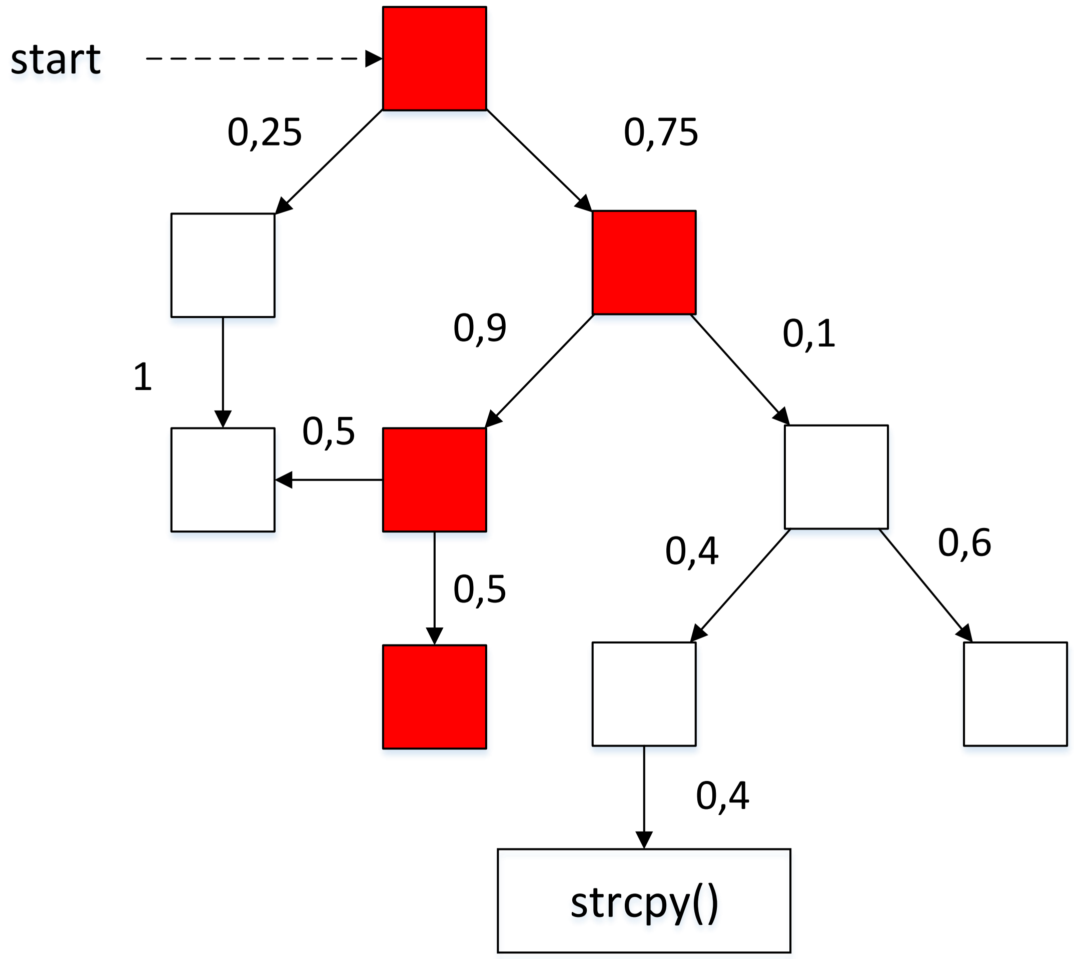
- Struktura izvršavanja programa se posmatra kao Markovljev lanac, pri čemu se verovatnoće prelaza između blokova izračunate na osnovu prethodno generisanih ulaznih podataka.



Definisanje funkcije prilagođenosti preko Markovljevih lanaca

Mera prilagođenosti.

- Verovatnoća putanje označene na slici je $0{,}75 \times 0{,}9 \times 0{,}5$.
- Što je verovatnoća manja (tj., što je posmatrani blok manje posećivan), “kvalitet” ulaznih podataka koji su uzrokovali posmatranu putanju izvršavanja je veći.



Završne napomene

- Kod testiranja softvera zasnovanog na principu crne kutije, izvorni kod nije dostupan.
  - Odakle znamo strukturu izvršavanja programa?
  - Struktura se **heuristički aproksimira**.
- Kod testiranja softvera zasnovanog na principu crne kutije, dostupna je specifikacija softvera.
  - Međutim, pojam specifikacije softvera nije detaljnije razmatran – o tome će biti reči na nekom od narednih predavanja.

Literatura

1. Embleton, S., Sparks, S., Cunningham, R. (2006) Sidewinder: An Evolutionary Guidance System For Malicious. Input Crafting, Black Hat USA 2006, http://www.blackhat.com/presentations/bh-usa-06/BH-US-06-Embleton.pdf.
2. M. Gnjatović, D. Stefanović (2018): Izabrane teme iz bezbednosti i sigurnosti informacionih sistema. Fakultet tehničkih nauka, Univerzitet u Novom Sadu.
3. Paul A. Gagniuc (2017): Markov Chains – From Theory to Implementation and Experimentation. John Wiley & Sons, USA, NJ.
4. Radni materijal iz predmeta Menadžment rizikom, Univerzitet u Novom Sadu, http://www.ef.uns.ac.rs/Download/menadzment_rizikom_master/2010-01-11_prilog_za_markovljeve_lance.pdf

Hvala na pažnji

Pitanja su dobrodošla.