

Univerzitet u Kragujevcu  
Fakultet inženjerskih nauka



## Seminarski rad iz predmeta Ekspertski sistemi

### Tema:

Ekspertski sistem koji otklanja probleme koji se javljaju kod bicikala

Student:  
Nikola Mitrevski 603/2017

Predmetni profesori:  
Vesna Ranković  
Predmetni saradnik:  
Tijana Šušteršić

Kragujevac 2020.

## Sadržaj:

1. Uvod .....	2
2. Opis ekspertskog sistema .....	4
3. Realizacija .....	5
3.1 Klasa „Bicikla“ .....	5
3.2 Klasa “Frame1” .....	6
3.3 Klasa “JFrame2” .....	6
3.4 Pravila .....	8
3.4.1 Pravila koja su pomogla u realizaciji pravila ES .....	8
3.5 Stabla odlučivanja .....	11
3.6 Ulančana aktivacija pravila .....	24
3.7 Izlaz .....	30
4. Zaključak .....	32
5. Literatura .....	33

## 1. Uvod

U ovom dokumentu biće objašnjen ekspertski sistem koji rešava osnovne probleme koji se javljaju kod bicikala, ali pre svega toga, potrebno je prvo objasniti pojam "ekspertski sistem".

Ekspertski sistem je računarski program kojim se rešava problem na način na koji to čini ekspert, odnosno čovek iz date oblasti.

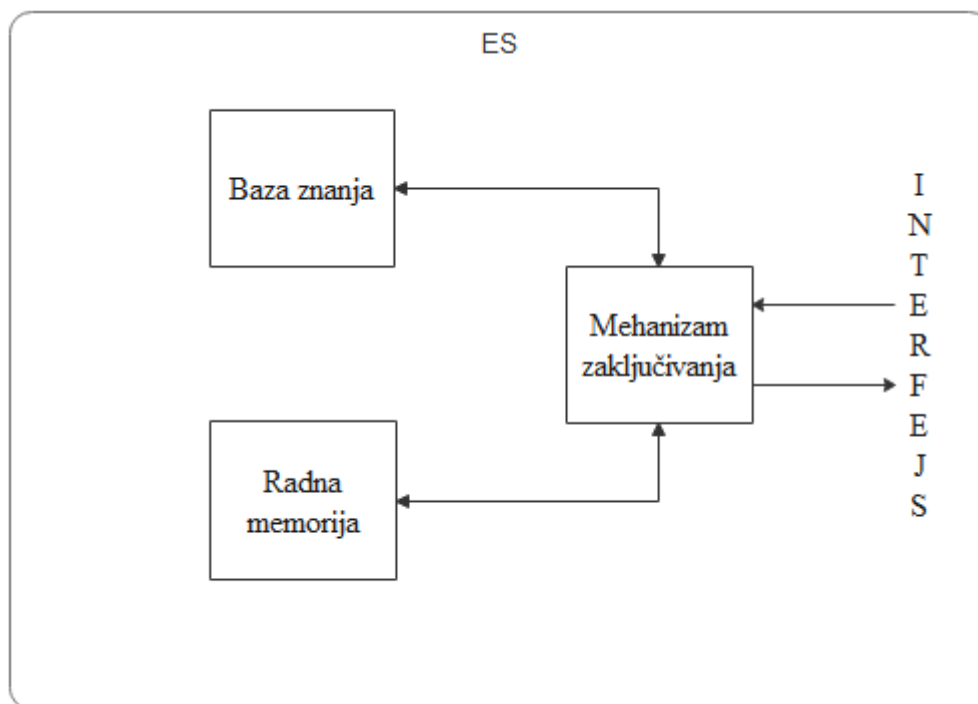
Ovakav sistem nastao je u okviru razvoja oblasti veštačke inteligencije, 70-tih i 80-tih godina dvadesetog veka.

Oblast „veštačka inteligencija“ se bavi proučavanjem kako da kompjuteri rade ono što ljudi trenutno rade bolje.

Neke od prednosti ekspertskog sistema u odnosu na eksperta su:

- Ljudsko znanje vremenom se gubi, ako se ne koristi često;
- Nakon nekog vremenskog perioda, aktivnog rada, čovek(expert) postaje umoran i ne razmišlja istim intezitetom kao na početku;
- Eksperski sistem je dosta brži od čoveka u rešavanju problema, koji su već definisani u okviru tog ekspertskog sistema.

Komponente ekspertskog sistema su: baza znanja, radna memorija i mehanizam zaključivanja.



Slika 1 Prikaz komponenata ES

Baza znanja je specijalizovana i jedinstvena za konkretni sistem koji sadrži znanje eksperata iz određene oblasti.

Informacije su unete putem sistema za prikupljanje znanja i ne menja se tokom vremena.

Trenutni podaci o problemu koji se rešava se nalaze u radnoj memoriji.

U poređenju sa bazom znanja, trenutni podaci su promenljivi i održavaju trenutno stanje u procesu rešavanja.

Mehanizam zaključivanja na osnovu trenutnih podataka i baze znanja rešava problem.

U razvoju ekspertskog sistema učestvuju ekspert, inženjer i korisnik.

Inženjer prikuplja heurističko znanje od jednog ili više eksperata i na osnovu toga dizajnira i implementira ES, koji kasnije i održava.

Ekspert poseduje znanje i iskustvo na osnovu kojih rešava probleme iz određenog domena.

Korisnik radi sa ekspertskim sistemom, tako što definiše zahteve vezane za korisnički interfejs sistema.

## 2. Opis ekspertskog sistema

Da bi algoritam(mehanizam zaključivanja) uspešno pronašao rešenje, potrebno je od korisnika prikupiti informacije o delu bicikle kod kojeg je nastao problem i takve informacije se smeštaju u radnu memoriju.

Informacije se prikupljaju preko unapred definisanog korisničkog interfejsa, gde korisnik selektuje određene stavke, koje su mu ponudjene.

Mehanizam zaključivanja na osnovu početnih činjenica i baze znanja koja je unapred definisana, pronalazi i prikazuje rešenje, samo ako je pre toga korisnik uspešno selektovao ponudjene stavke.

Ako je izabrao neku stavku koja je bila ponudjena, ali koja ne pomaže u rešavanju problema, korisnik će dobiti obaveštenje o tome i moraće ponovo da vrši selekciju kako bi dobio rešenje za svoj problem.

U okviru ovog sistema ponudjeni delovi su: guma, bandas, amortizer, lanac, srednji pogon, kurbla, pedala, kocnica, prednji menjac, zadnji menjac, kaseta, sticna, sediste, lula volana, tocak, viljuska kod kojih se mogu javiti problemi.

Ti problemi definisani su preko pravila koja se nalaze u okviru baze znanja, koja je smeštena u fajl „pravila.drl“.

Ukupno se u bazi znanja nalazi 60 pravila od kojih se može maksimalno aktivirati 42 pravila preko definisanog korisničkog interfejsa.

### 3. Realizacija

Programi pisani u Eclipse razvojnom okruženju su Java fajlovi koje kreiramo za potrebe našeg projekta. Uobičajena praksa je da za svaki Java fajl, zadužen za određeni posao, kreiramo pakete u kojima će se ovi fajlovi nalaziti.

Glavni Java fajl koji predstavlja main klasu importuje sve potrebne klase kako bi imao pristup vlasništvima tih klasa, metodama, koristeći komandu **import**.

Glavna klasa ovog projekta naziva se "Frame1" i ona je zadužena za iscrtavanje GUI prozora preko kog korisnik vrši selektovanje jednog od ponudjenih delova i ova klasa unutar sebe stvara objekat tipa "JFrame2", čijem konstruktoru kao argument prosledjuje prethodno selektovan "deo".

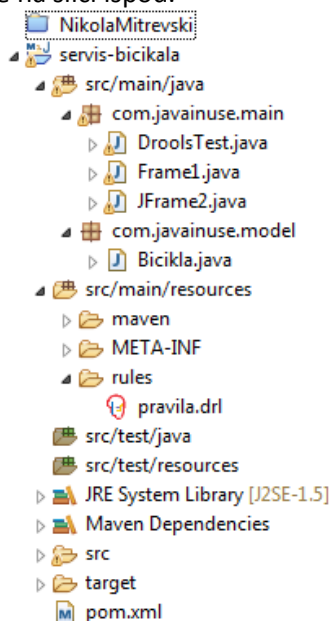
Klasa koja se zove "JFrame2" takodje je zadužena za iscrtavanje GUI prozora preko kog korisnik vrši selektovanje ponudjenih stavki, a te stavke predstavljaju početne činjenice ekspertskog sistema(ES), kao i primljen deo.

Pored toga klasa "JFrame2" zadužena je za pokretanje pravila na osnovu kojih dobijamo preporučena rešenja.

Baza pravila nalazi se u fajlu pod nazivom „pravila.drl“.

Klasa „Bicikla“ je zadužena za strukturu objekta čija je namena da definiše sve potrebne ulaze i izlaze u jedan ES.

Struktura projektnog fajla, data je na slici ispod.



Slika 2 Prikaz strukture projektnog fajla

#### 3.1 Klasa „Bicikla“

Klasa "Bicikla" je klasa koja sadrži potrebne ulaze u ES od kojih izlazi ES zavise.

Ulazi i izlazi ES su predstavljeni pomoću promenljivih unutar ove klase.

Promenljive koje predstavljaju izlaze iz ES u svom nazivu sadrže reč „akcija“, dok sve ostale promenljive predstavljaju ulaze u ES.

Klasa „Bicikla“ mora da dozvoli klasi „JFrame2“ da setuje osnovne ulaze u sistem i to pomoću javnih metoda set() koje smo definisali u klasi „Bicikla“.

Takodje mora da sadrži i definisane javne get() metode pomoću kojih klasa „JFrame2“ dobija izlaze iz ES.

Pošto se klasa „Bicikla“ i „JFrame2“ ne nalaze u istom paketu, potrebno je unutar klase „JFrame2“ importovati klasu „Bicikla“, linija ispod.

```
import com.javainuse.model.Bicikla;
```

### 3.2 Klasa "Frame1"

Klasa "Frame1" služi za kreiranje grafičkog korisničkog interfejsa(GUI) koji komunicira sa korisnikom. Ova klasa nasledjuje klasu "JFrame" za kreiranje GUI-a.

Kao i svaka glavna klasa, klasa "Frame1" sadrži main metodu koja vrši stvaranje instance "grafičke" klase koja tek onda postaje vidljiva korisniku, slika ispod.

```
public static void main(String[] args) {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                Frame1 window = new Frame1();
                window.frame.setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}
```

Slika 3 Prikaz dela koda koji se odnosi na main metodu

Pored main metode, ova klasa u sebi sadrži i svoj dizajn.

Dizajn čine veličina prozora, oblik prozora, dugme(JButton), JComboBox i font slova.

Prilikom instanciranja JComboBox-a se koristi niz „delovi“ u kome se nalaze elementi koji predstavljaju stavke JComboBox-a.

Dinamički deo ove klase čine objekat tipa JComboBox i objekat tipa JButton koji imaju svoje osluškivače. Namena osluškivača JComboBox-a je čuvanje selektovane stavke, koju je korisnik izabrao, slika ispod.

```
comboBox.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        deo = comboBox.getSelectedItem().toString();
    }
});
```

Slika 4 Prikaz dela koda koji se odnosi na osluškivač objekta JComboBox-a

Namena osluškivača JButton-a je da stvori objekat klase „JFrame2“, gde preko konstruktora kao argument prosledjuje selektovanu stavku u vidu Stringa, prikaže novi objekat i zatvori trenutni objekat nakon otvaranja novog, slika ispod.

```
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        // preko konstruktora("JFrame2") se prosledjuje podatak("deo")
        JFrame2 frame2 = new JFrame2(deo);

        frame2.show(); // prikazuje se nova forma(tipa JFrame2)

        frame.dispose();// zatvara se trenutna forma nakon otvaranja nove
    }
});
```

Slika 5 Prikaz dela koda koji se odnosi na osluškivač objekta JButton-a

### 3.3 Klasa "JFrame2"

Klasa "JFrame2" služi za kreiranje grafičkog korisničkog interfejsa(GUI) koji komunicira sa korisnikom. Ova klasa nasledjuje klasu "JFrame" za kreiranje GUI-a.

Klasa "JFrame2" sadrži parametrizovan konstruktor, koji se poziva prilikom stvaranja instance ove klase od strane objekta klase "Frame1" i koja tek onda postaje vidljiva korisniku.

Ovaj konstruktor ima parametar koji predstavlja „deo“, koji je prethodno korisnik izabrao i taj parametar se koristi za inicijalizaciju polja ove klase i to polje je jedna od početnih činjenica ES.

Dinamički deo ove klase čine metode „actionPerformed“ gde jednu od njih poziva osluškivač objekta JButton-a, slika ispod.

```
btnDalje.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
```

*Slika 6 Prikaz dela koda koji se odnosi na osluškivač objekta JButton-a*

Van ovih metoda se instancira objekat tipa JComboBox-a ali sa različitom vrednošću atributa, gde atribut mora biti tipa String.

Sva polja klase „JFrame2“ koja se mogu proslediti kao niz String-ova konstruktoru JComboBox-a sadrže u svom nazivu reč „pitanje“, gde se na početku stvaranja prosledjuje odgovarajuće pitanje na osnovu polja „deo“, slika ispod.

```
if(deo.equals("Guma")) {
    //final JComboBox comboBox = new JComboBox(pitanje1);
    comboBox = new JComboBox(pitanje1);
    contentPane.add(comboBox);

    comboBox.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent arg0) {
```

*Slika 7 Prikaz dela koda koji se odnosi na stvaranje objekta JComboBox-a*

Kasnije se vrši setovanje JComboBox-a u zavisnosti od unetog odgovora koji se zadaje selekcijom od strane korisnika, slika ispod.

```
if(odgovor1.equals(pitanje1[1])) {
    if(i == 2) {
        comboBox.removeAllItems();
        for(int j = 0; j < pitanje2.length; j++) {
            comboBox.addItem(pitanje2[j]);
        }
    }
}
```

*Slika 8 Prikaz dela koda koji se odnosi na setovanje objekta JComboBox-a*

Svaki put kada treba da se izvrši zamena(setovanje) trenutnog sadržaja JComboBox-a, prvo se vrši brisanje prethodnog sadržaja, pa se tek onda vrši setovanje i to prolaženjem kroz odgovarajući niz String-ova i dodavanjem elemenata tog niza na mestima stavke, slika iznad.

Polja koja u svom nazivu imaju reč „odgovor“ su polja koja predstavljaju početne činjenice i ona se inicijalizuju na osnovu selektovane stavke objekta JComboBox-a, slika ispod.

```
odgovor1 = (String)comboBox.getSelectedItem();
```

*Slika 9 Prikaz dela koda koji se odnosi na inicijalizaciju polja klase „JFrame2“*

Kada korisnik unese početne činjenice, one se dodeljuju objektu klase „Bicikla“ preko set() metoda, slika ispod.

```
bicikl.setGuma_stanje(odgovor1);
bicikl.setVentil_stanje(odgovor2);
```

*Slika 10 Prikaz dela koda koji se odnosi na prosledjivanje ulaznih početnih činjenica ES*

Nakon postavljanja početnih vrednosti, potrebno je importovati sve setovane činjenice pomoću komande **kSession.insert(bicikl);** i zatim pokrenuti obradu pravila komandom **kSession.fireAllRules();**. Nakon izvršavanja svih pravila, pravila koja su pronadjena na osnovu početnih činjenica, ta pravila setuju vrednosti izlaznih činjenica koje ćemo u programu kasnije prikazati preko objekta klase JOptionPane, kao rešenje zadatog problema, slika ispod.

```
if(bicikl.getGuma_akcija1() != null) resenje += bicikl.getGuma_akcija1()+"\n";
if(bicikl.getGuma_akcija2() != null) resenje += bicikl.getGuma_akcija2();
if(resenje != "") JOptionPane.showMessageDialog(null, resenje);
```

*Slika 11 Prikaz dela koda koji se odnosi na formiranje i prikazivanje rešenja ES*



### 3.4 Pravila

Sva pravila koja postoje za problem određenog dela bicikle, definisali su eksperti iz te oblasti i ona su smeštena u okviru fajla „pravila.drl“, koji se nalazi u paketu src/main/resources/rules.

Za kreiranja pravila u .drl fajlu morala je biti ispoštovana određena sintaksa:

```
rule "naziv"
//atributi pravila
when
//uslov
then
//zaključak
end
```

Naziv pravila može biti bilo kakav niz slova, ali obavezno mora biti pod znacima navoda.

Atributi pravila su opcioni i utiču na ponašanje pravila.

U projektu, nije korišćen ni jedan atribut.

Uslov pravila sadrži jedan ili više logičkih iskaza koji liče na logičke izraze u Javi.

Ovde se koristi zajednici objekat o:Bicikla i postavljeni su različiti iskazi kako bi se odredila određena rešenja.

Zaključak pravila se daje u formi običnih Java naredbi.

U ovom delu setujemo promenljive koje predstavljaju rešenja ES.

#### 3.4.1 Pravila koja su pomogla u realizaciji pravila ES

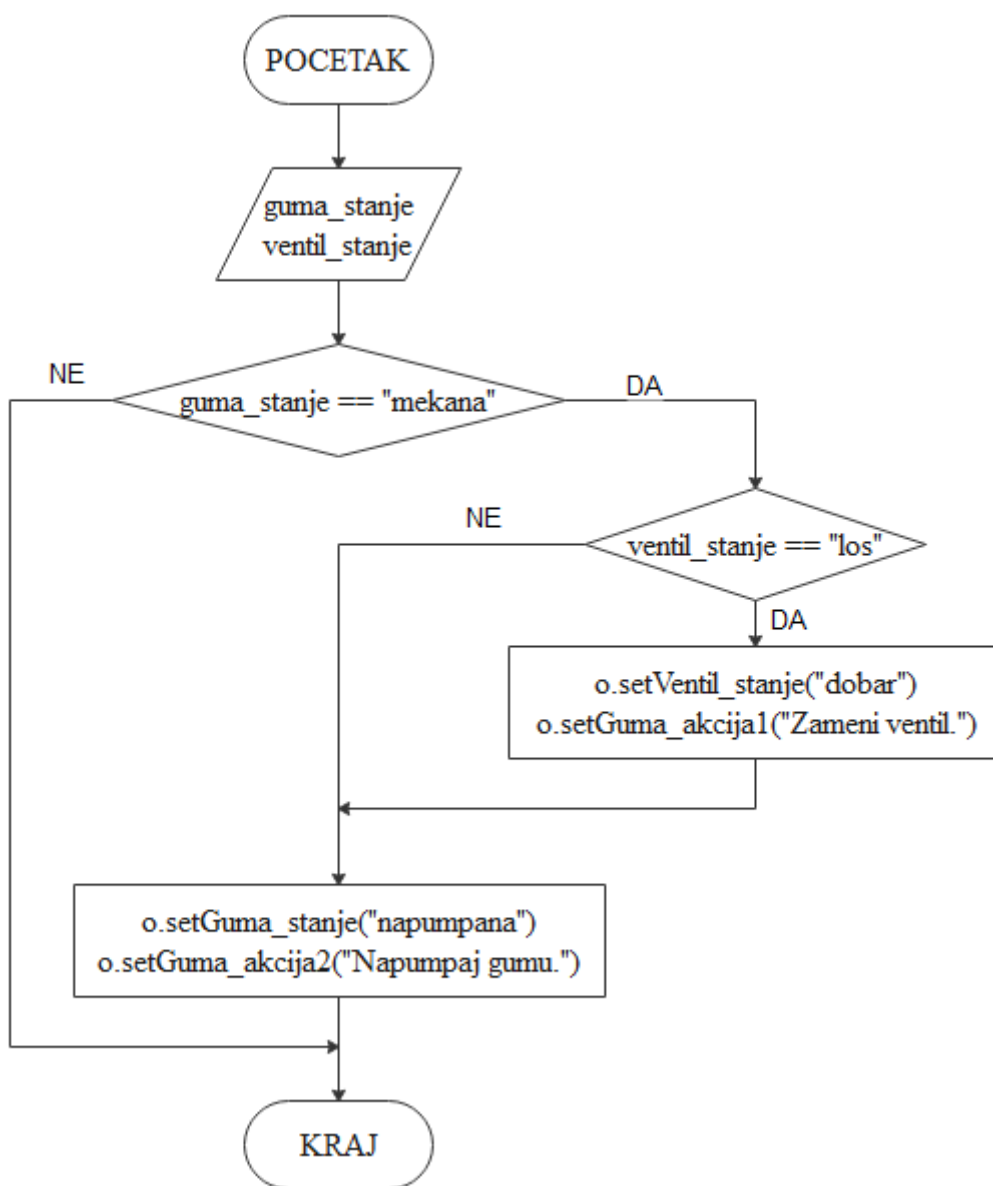
1.
  - Ako je guma mekana, onda je napumpaj;
  - Ako je guma i dalje mekana, tada proveriti ventil;
  - Ako je ventil loš, onda ga zameni novim i napumpaj gumu;
  - U suprotnom zameni gumu i napumpaj je.
2.
  - Ako bandas krivi, tada proveriti da li je pokidana neka zica;
  - Ako je zica pokidana, onda je zameni i centriraj bandas;
  - Ako zica nije pokidana i bandas ne može da se nacentrira, onda promeniti bandas i nacentriraj;
  - U suprotnom nacentriraj bandas.
3.
  - Ako je amortizer neispravan i ako je vazdusan, tada proveriti da li je prepumpan;
  - Ako je amortizer perumpnan, onda smanji pritisak vazduha u amortizeru;
  - U suprotnom povecaj pritisak vazduha u amortizeru(napumpaj ga).
4.
  - Ako je amortizer neispravan i ako ima opruge, onda ga podmazati;
  - Ako je amortizer i dalje neispravan, tada proveriti opruge;
  - Ako su opruge previse jake, onda ih zameni blazim;
  - U suprotnom zameni opruge jacim.
5.
  - Ako lanac preskace, tada proveriti njegovu duzinu;
  - Ako je lanac istegnut, onda ga zameni;
  - Ako lanac nije istegnut, tada proveriti da li su potroseni prednji ili zadnji zupcanici;
  - Ako su potroseni, onda ih zameni (prednji, zadnji ili oba);
  - U suprotnom zameni freewheel ili freehub.
6.
  - Ako srednji pogon ima lufta, tada proveriti da li su odvijene solje;
  - Ako su odvijene solje, onda ih pritegni;

- Ako solje nisu odvijene, tada proveri lezajeve;
  - Ako su lezajevi losi, onda ih zameni;
  - U suprotnom zameni osovinu srednjeg pogona.
- 7.
- Ako je kurbila labava, onda pritegni sraf koji ide u osovinu;
  - Ako je kurbila i dalje labava, tada proveri navoj u osovinu;
  - Ako je navoj osovine unisten, onda zameni osovinu;
  - U suprotnom zameni kurbilu.
- 8.
- Ako pedala ima lufta ili krcka, onda promeni lezajeve u pedali.
- 9.
- Ako kocnica ne koci i ako koci po površini obruca, tada proveri paknove;
  - Ako su paknovi potroseni, onda zameni paknove;
  - U suprotnom zameni sajlu i buzire.
- 10.
- Ako kocnica ne koci i ako koci po površini diska, tada proveri plocice;
  - Ako su plocice potrosene, onda zameni plocice;
  - Ako su plocice dobre, tada proveri disk;
  - Ako je disk los, onda zameni disk;
  - Ako je disk dobar i ako je u pitanju mehanicki disk, onda promeni sajlu i buzire;
  - U suprotnom promeni creva, klipove i sipaj ulje u hidraulicni disk.
- 11.
- Ako prednji menjac ne prebacuje, onda ga podesi;
  - Ako menjac i dalje ne prebacuje, tada proveri visinu menjaca;
  - Ako je podignut mnogo, onda ga spusti;
  - Ako je spusten mnogo, onda ga podigni;
  - Ako je visina menjaca dobra, tada proveri da li je u paraleli sa lancanicima;
  - Ako nije, onda ga ispravi;
  - Ako jeste, tada proveri sajlu;
  - Ako je uzrok sajla, onda promeni sajlu i buzire;
  - Ako je sa sajlom sve u redu, tada proveri index rucice menjaca;
  - Ako je propao index rucice, onda promeni rucicu;
  - U suprotnom zameni prednji menjac.
- 12.
- Ako zadnji menjac ne prebacuje, onda ga podesi;
  - Ako i dalje ne prebacuje, tada proveri da li je kuka menjaca kriva;
  - Ako je kuka menjaca kriva, onda zameni kuku;
  - Ako nije problem kuka, tada proveri index rucice menjaca;
  - Ako je propao index rucice, onda promeni rucicu menjeca;
  - Ako nije propao index rucice, tada proveri da li su potrosene rolnice;
  - Ako su potrosene rolnice, onda ih zameni;
  - Ako nisu problem rolnice, tada proveri sajlu;
  - Ako je sajla losa, onda je zameni;
  - U suprotnom zameni zadnji menjac.
- 13.
- Ako se kaseta vrti u prazno, onda skinuti kasetu i ocisti je;
  - U suprotnom promeni kasetu.
- 14.
- Ako sticna propada u ram, onda zategni selnu;
  - U suprotnom stavi deblju sticnu i zategnu selnu.

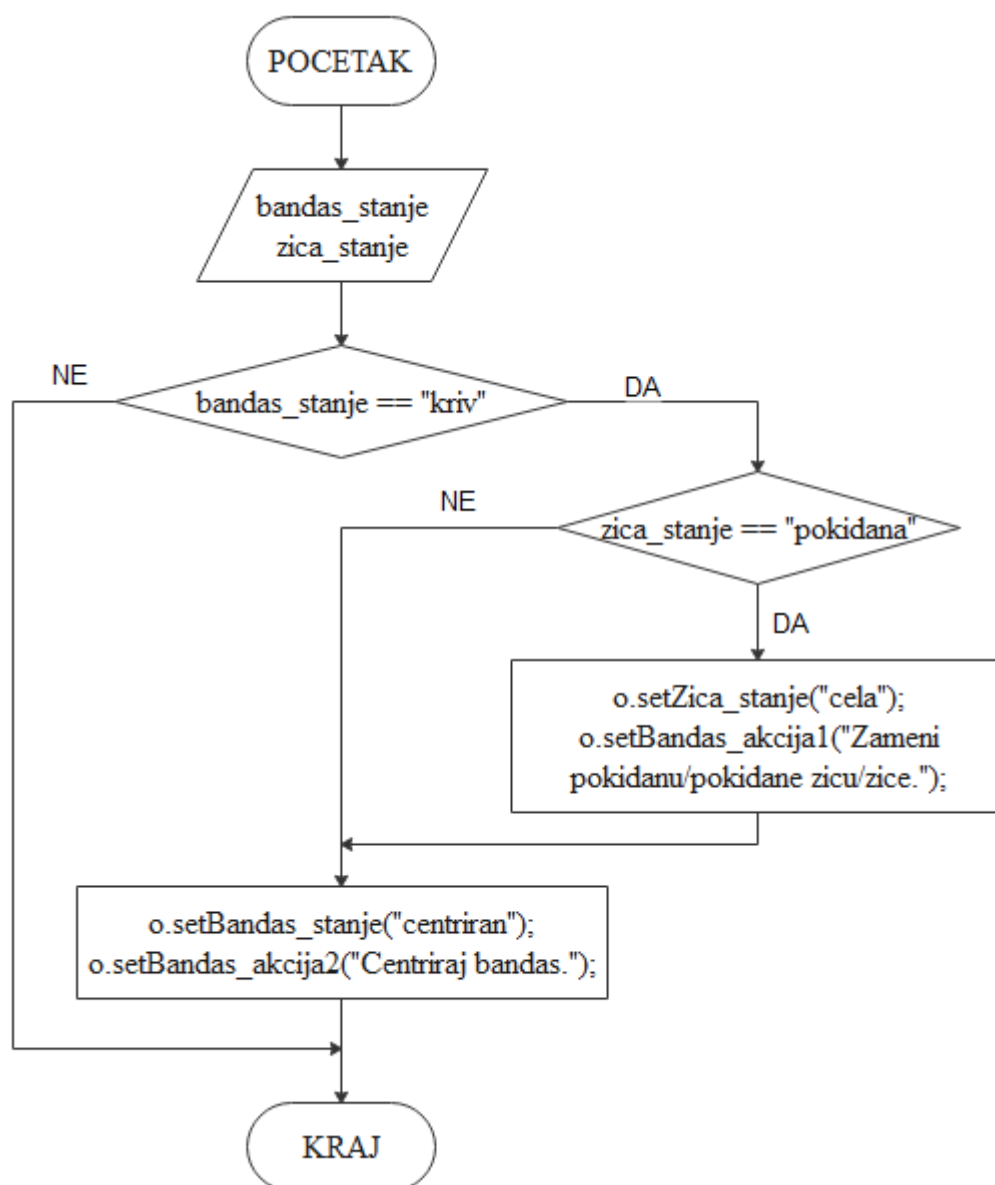
15.
  - Ako je sediste previse tvrdo, onda ga zameni meksim;
  - U suprotnom ga zameni tvrdjim.
16.
  - Ako je lula volana olabavljena, onda zategni sraf koji ulazi u orah.
17.
  - Ako je valan olabavljen, onda zategni srafove na luli.
18.
  - Ako tocak ima lufta, onda zameni lezajeve u glavi tocka;
  - U suprotnom zameni celu glavu;
19.
  - Ako viljuska ima lufta, onda promeni lezajeve u soljicama rama;
  - U suprotnom promeni viljusk.
20.
  - Ako ram ima rdju, onda osmirglaj zeljenu površinu i premazi je farbom.
21.
  - Ako lanac skripi, onda ga podmazi.
22.
  - Ako je u pitanju full suspension ram i ako ima u zglobovima lufta ili lose radi zglobov, onda zameni lezajeve u zglobovima;
  - U suprotnom zameni ram.
23.
  - Ako ram na sebi ima ulje, onda ga obrisi;
  - U suprotnom zameni hidraulicnu kockicu.
24.
  - Ako je ram udaren, tada proveriti da li je kriv;
  - Ako je ram kriv, onda zameni ram;
  - Ako ram nije kriv, tada proveriti da li je pukao;
  - Ako je pukao, onda ga zavari i ofarbaj;
  - U suprotnom zameni ram.
25.
  - Ako ti dužina bicikla ne odgovara, onda zameni dužinu lule;
  - U suprotnom zameni ram za ram odgovarajuće veličine.
26.
  - Ako ti je bicikl previse krut dok ga voziš po sumi, onda zameni ram za ram sa amortizerom;
  - U suprotnom zameni ram za ram sa većim amortizerom.
27.
  - Ako osećas vibracije pri vožnji, onda ispusti pritisak u gumama;
  - U suprotnom zameni gumu za siru i podesi pritisak.
28.
  - Ako je dropper post neispravan, tada proveriti sajlu;
  - Ako je sajla popustila, onda pritegni sajlu;
  - U suprotnom zameni dropper post.
29.
  - Ako je zaključavanje vile neispravno, tada proveriti sajlu;
  - Ako je sajla popustila, onda pritegni sajlu;
  - Ako je sajla dobra i stubovi vile imaju ogrebotine, onda zameni stubove vile i vrat vile;

### 3.5 Stabla odlučivanja

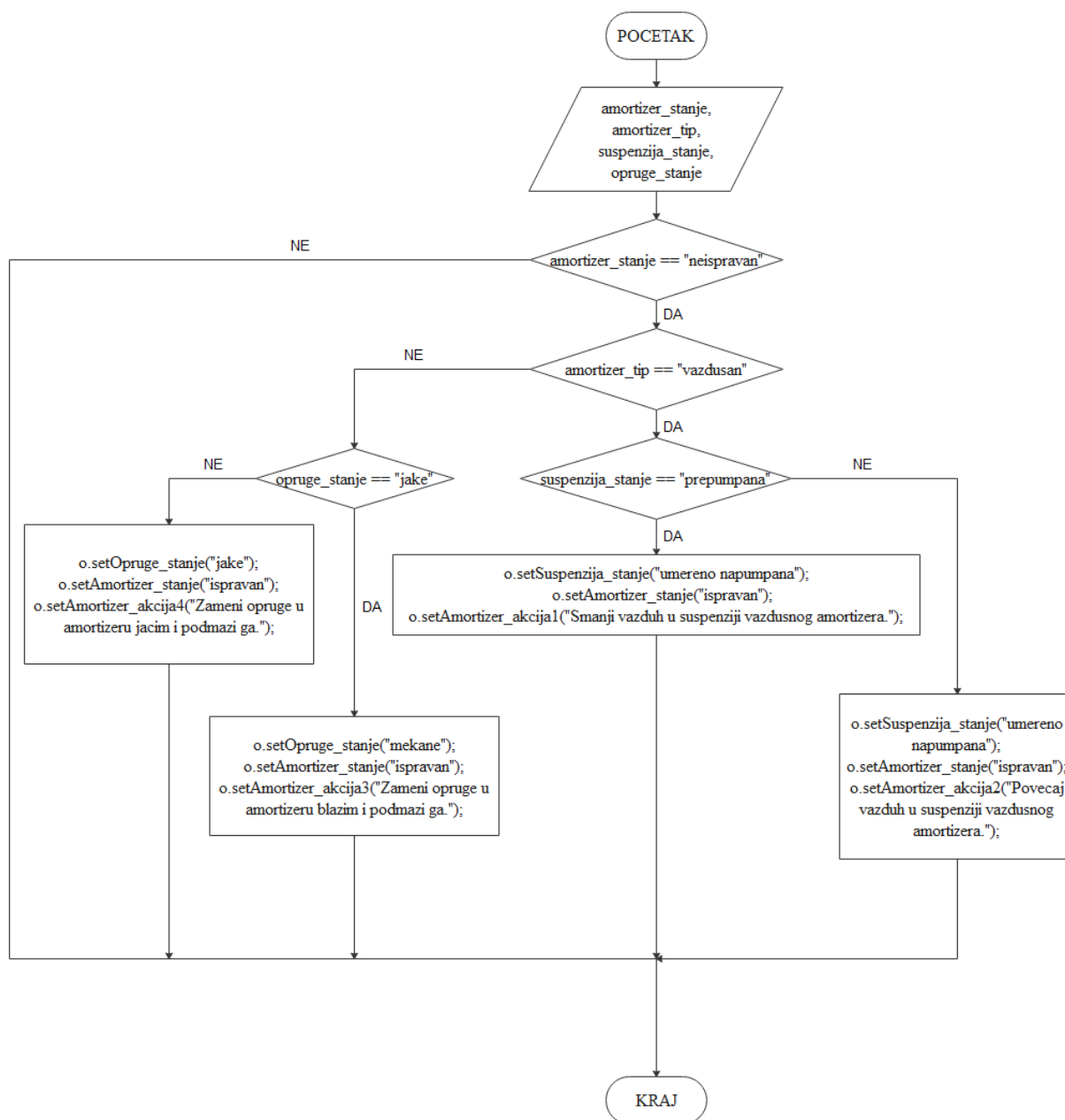
Na slikama ispod dati su prikazi stabala odlučivanja za ukupno 42 pravila, tj. za ona pravila koja je moguće aktivirati preko korisničkog interfejsa, od mogućih 60 pravila.



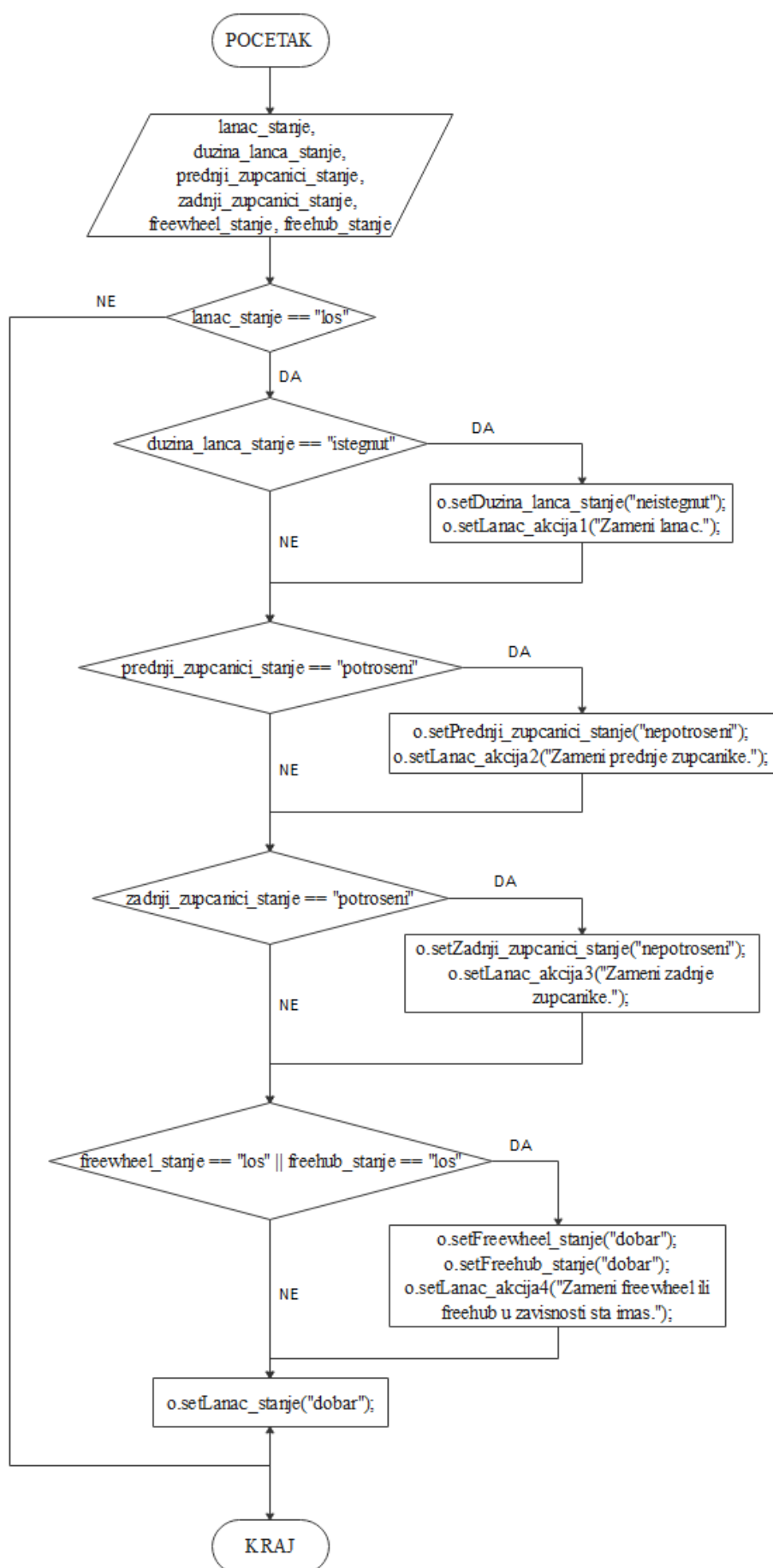
Slika 12 Stablo odlučivanja za problem "Guma"



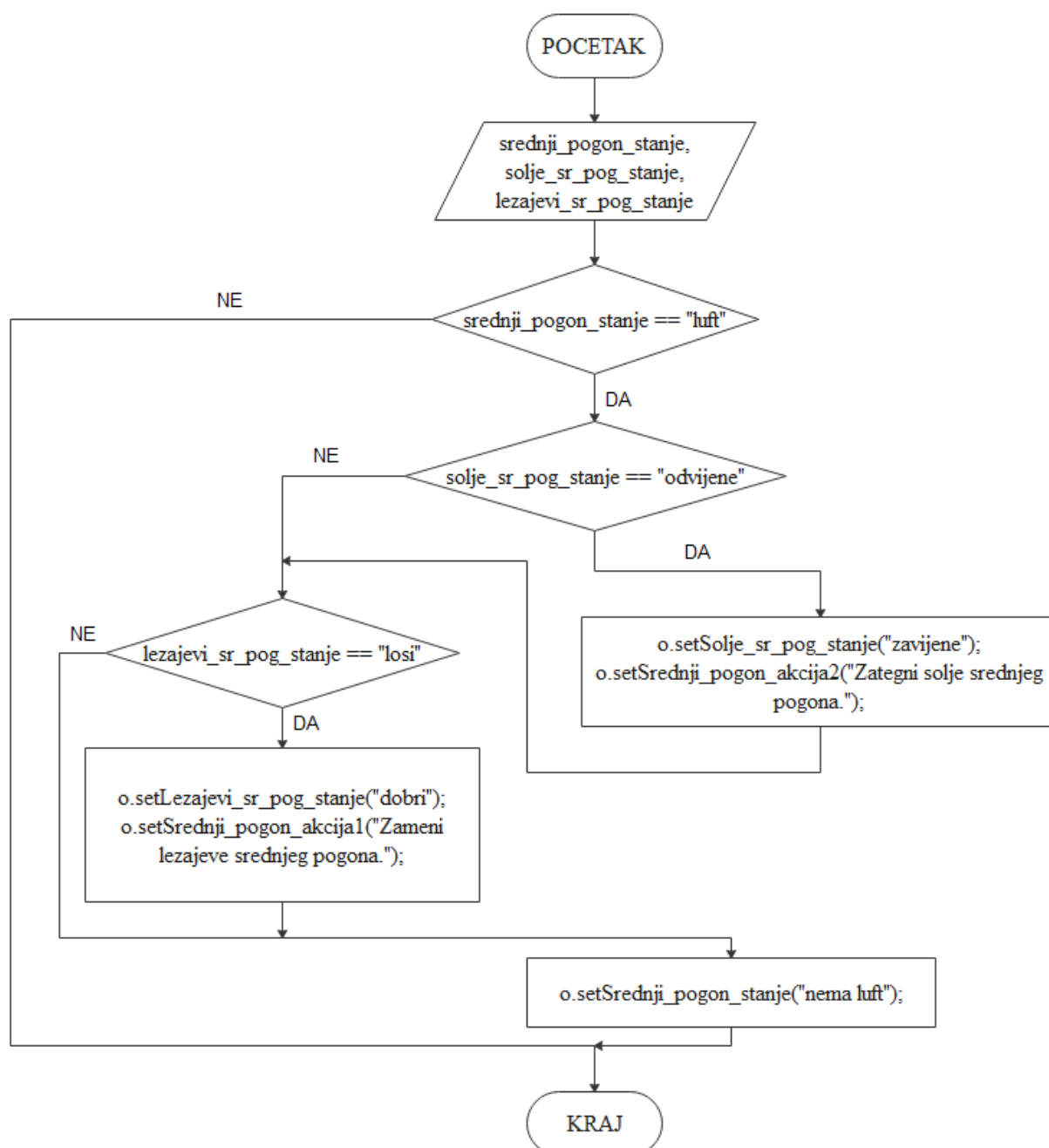
Slika 13 Stablo odlučivanja za problem "Bandaš"



Slika 14 Stablo odlučivanja za problem "Amortizer"

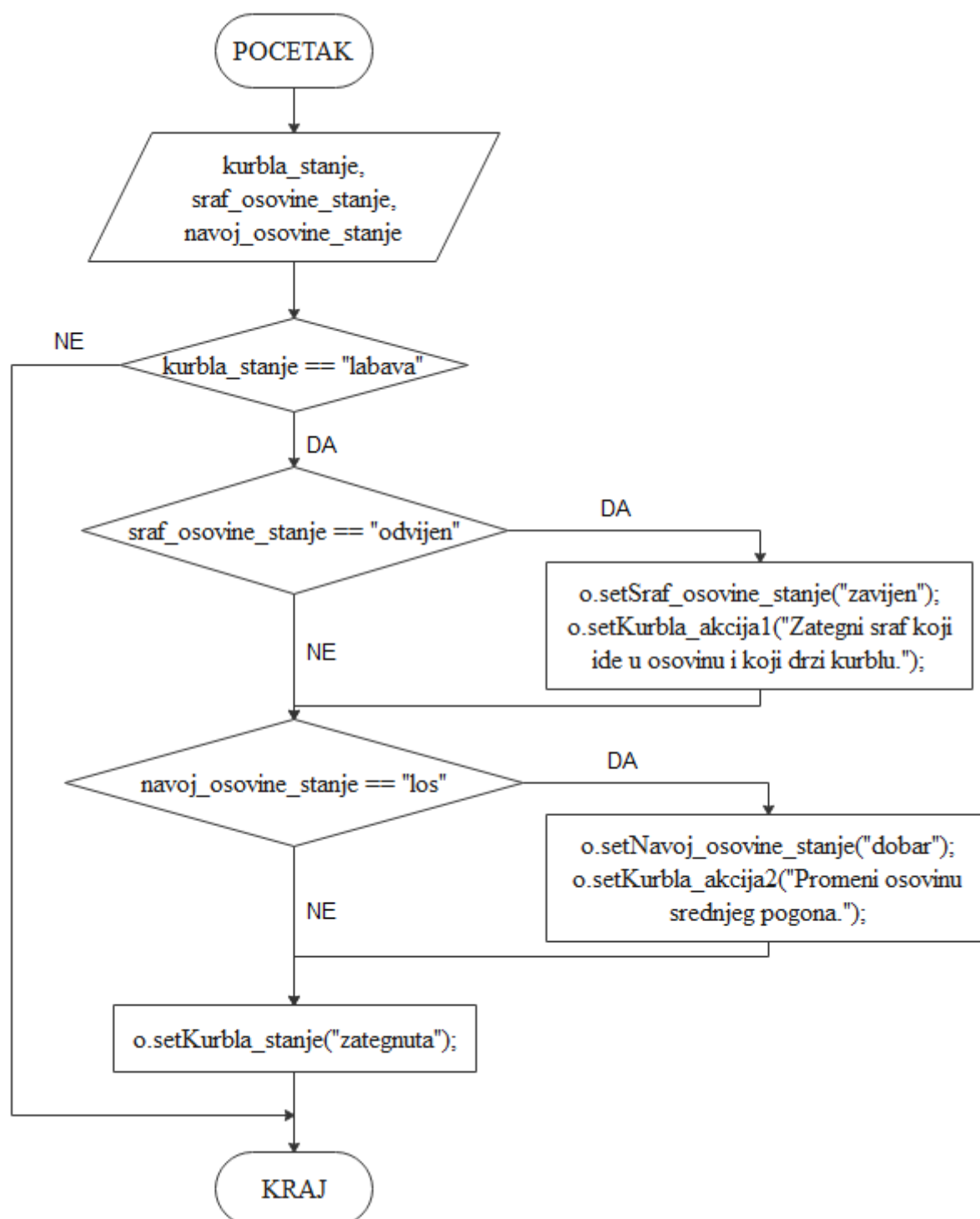


Slika 15 Stablo odlučivanja za problem "Lanac"

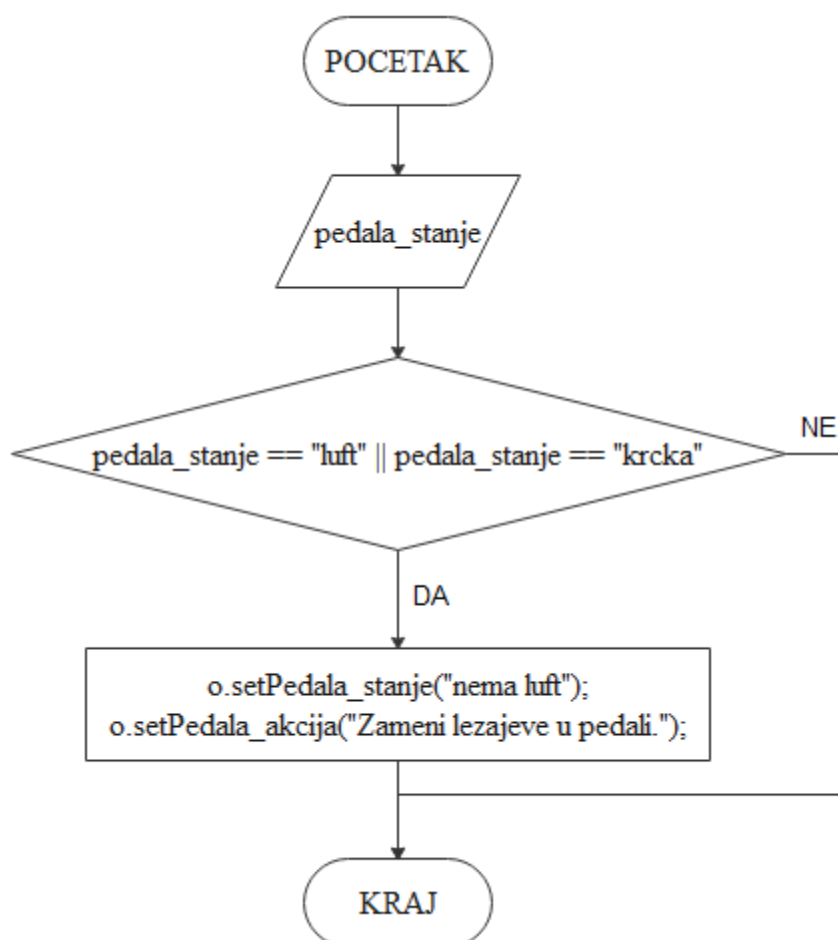


Slika 16 Stablo odlučivanja za problem "Srednji pogon"

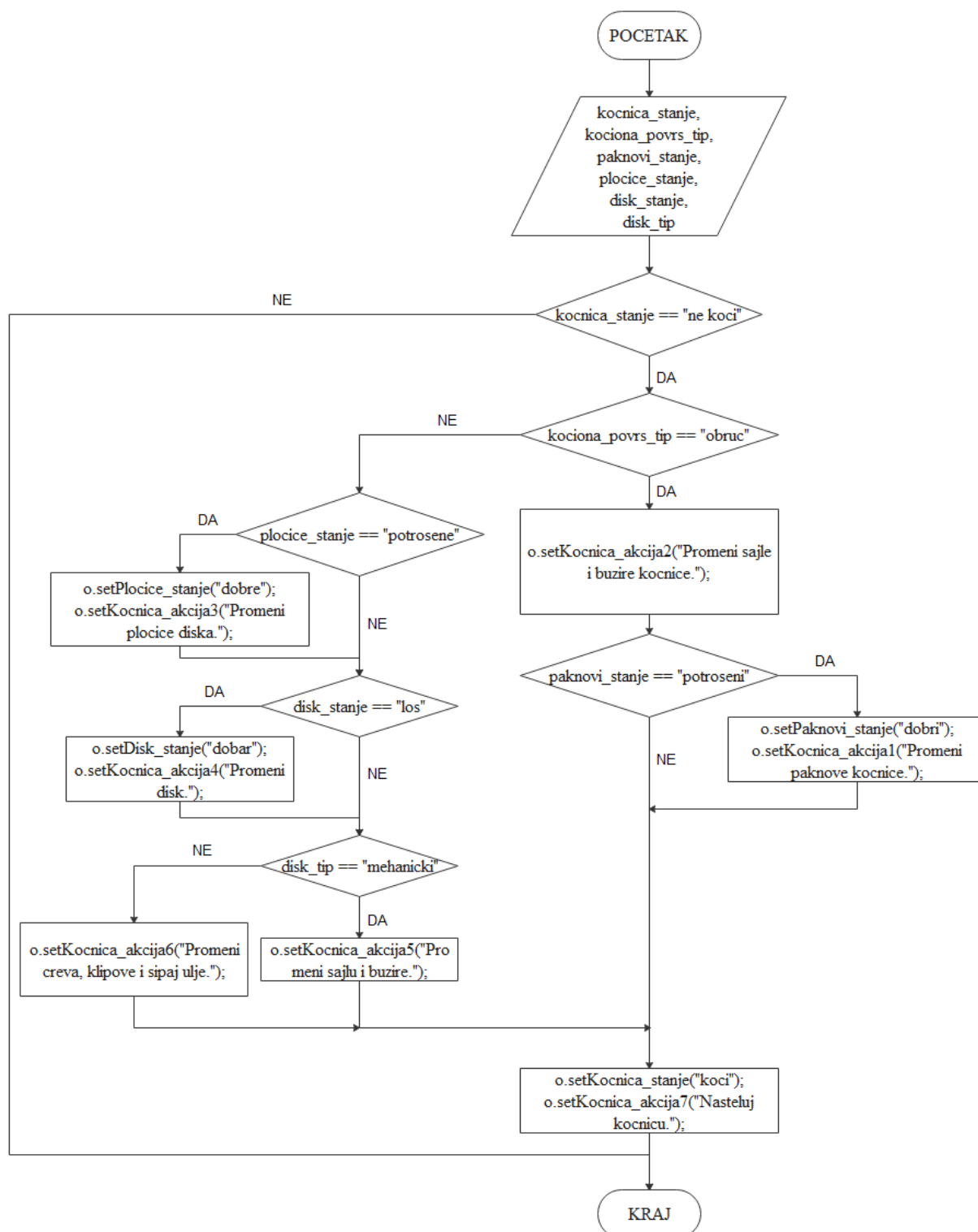




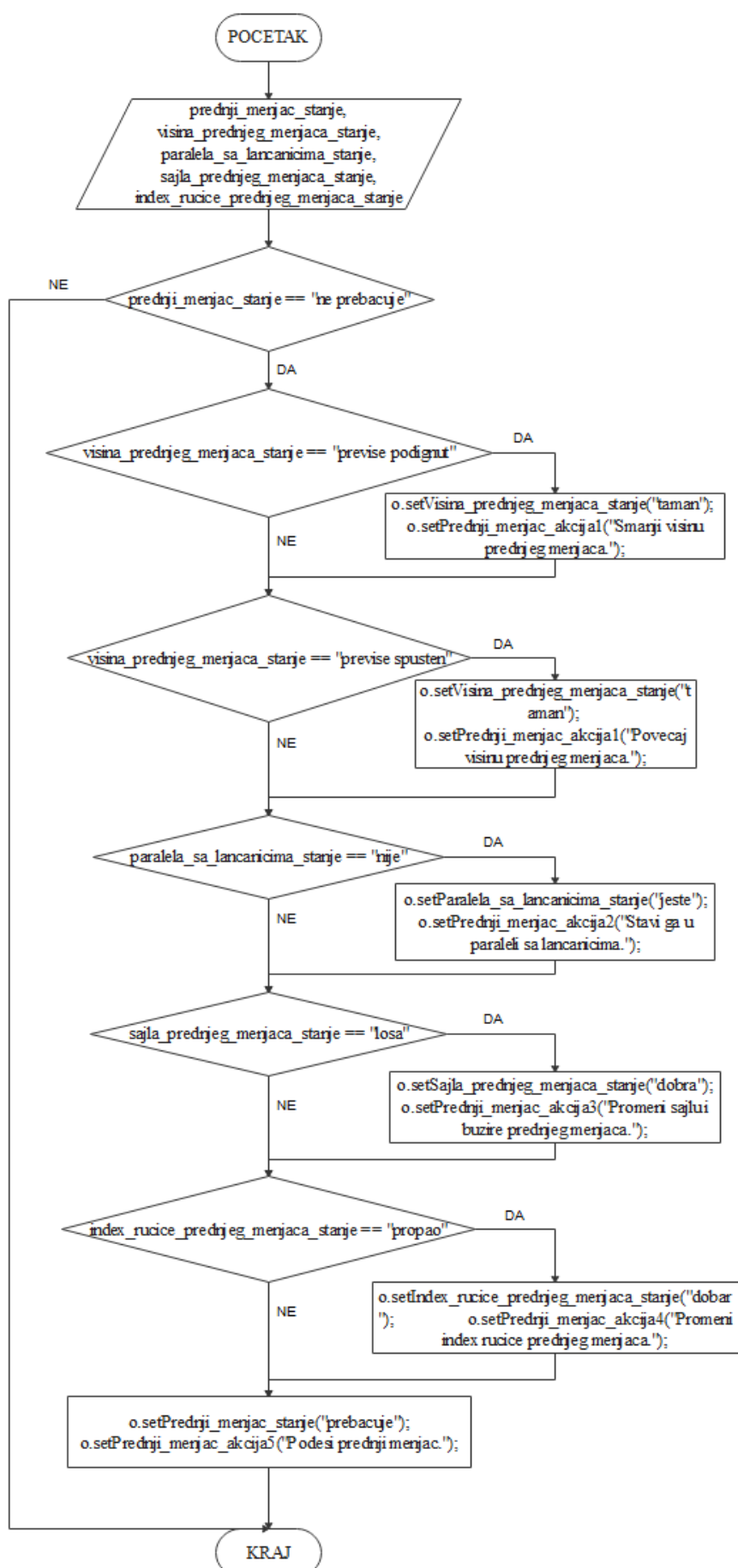
Slika 17 Stablo odlučivanja za problem "Kurbła"



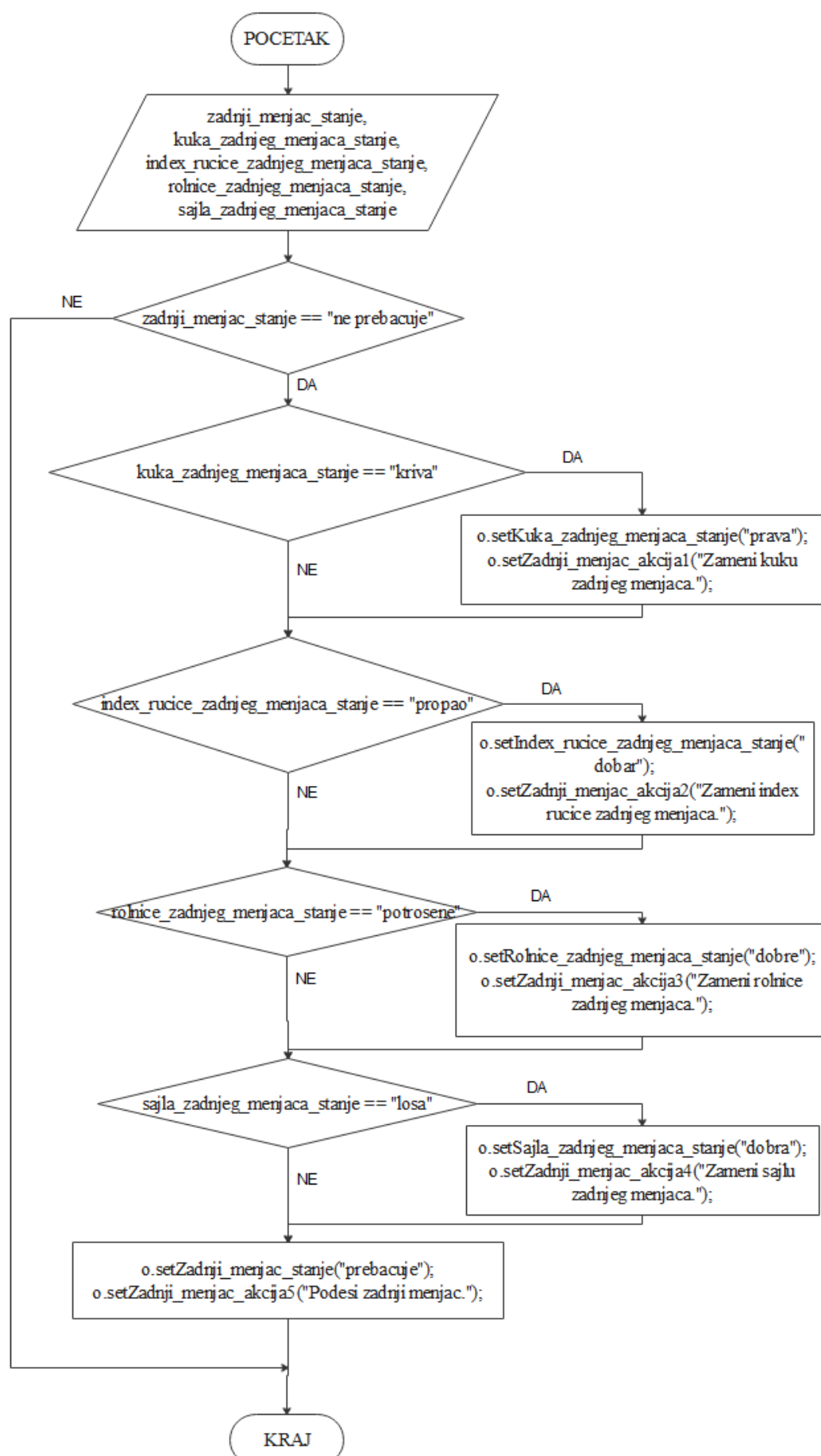
Slika 18 Stablo odlučivanja za problem "Pedala"



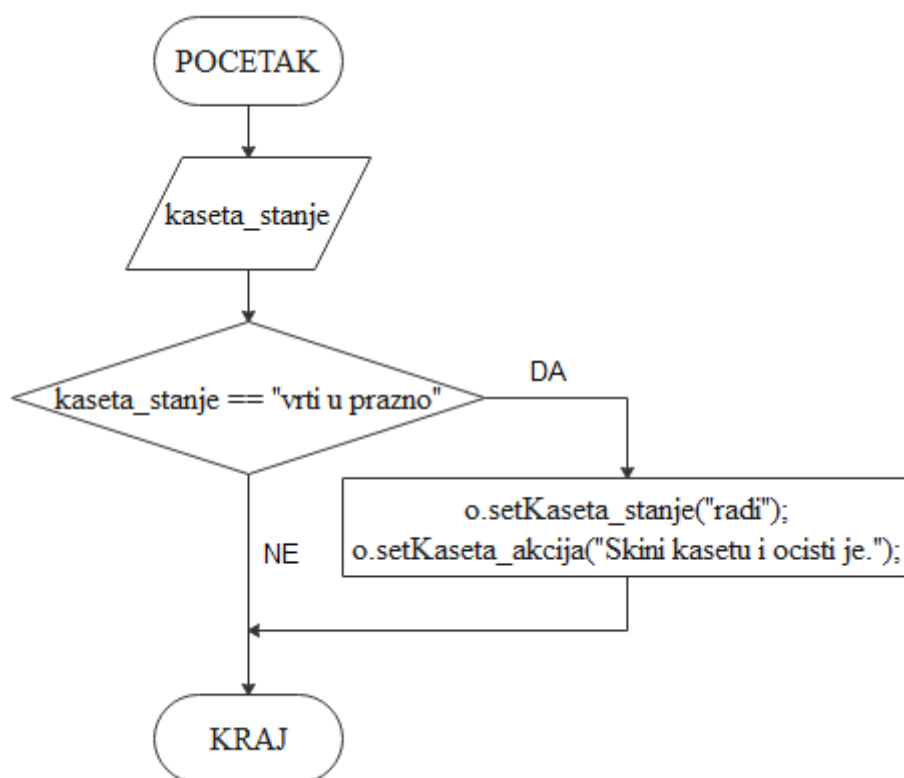
Slika 19 Stablo odlučivanja za problem "Kočnica"



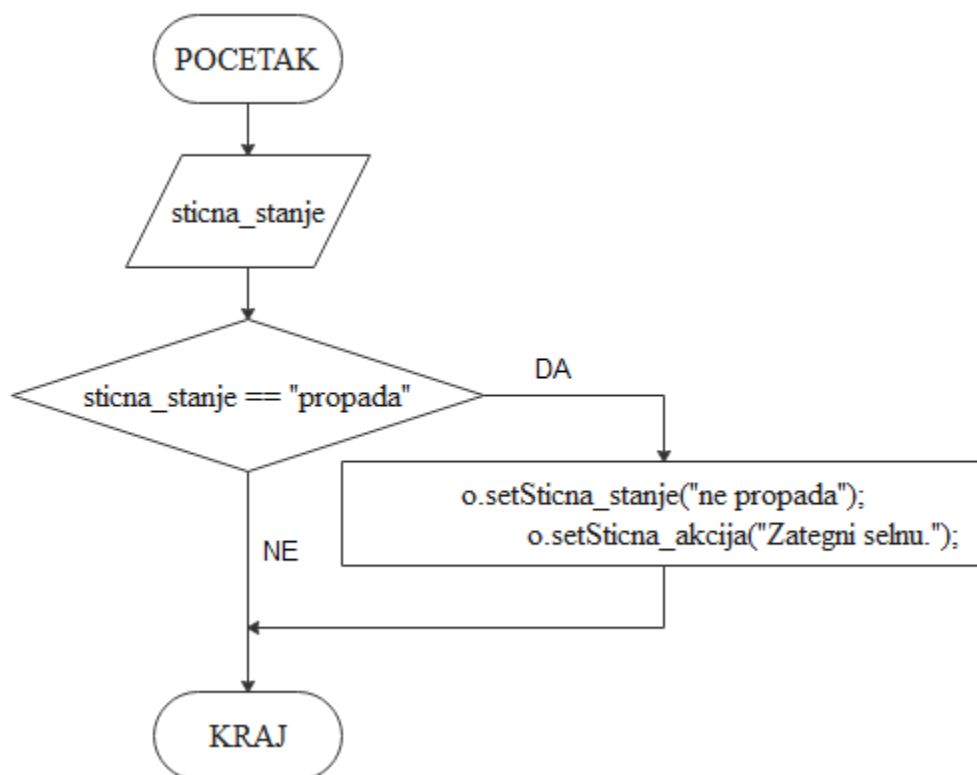
Slika 20 Stablo odlučivanja za problem "Prednji menjač"



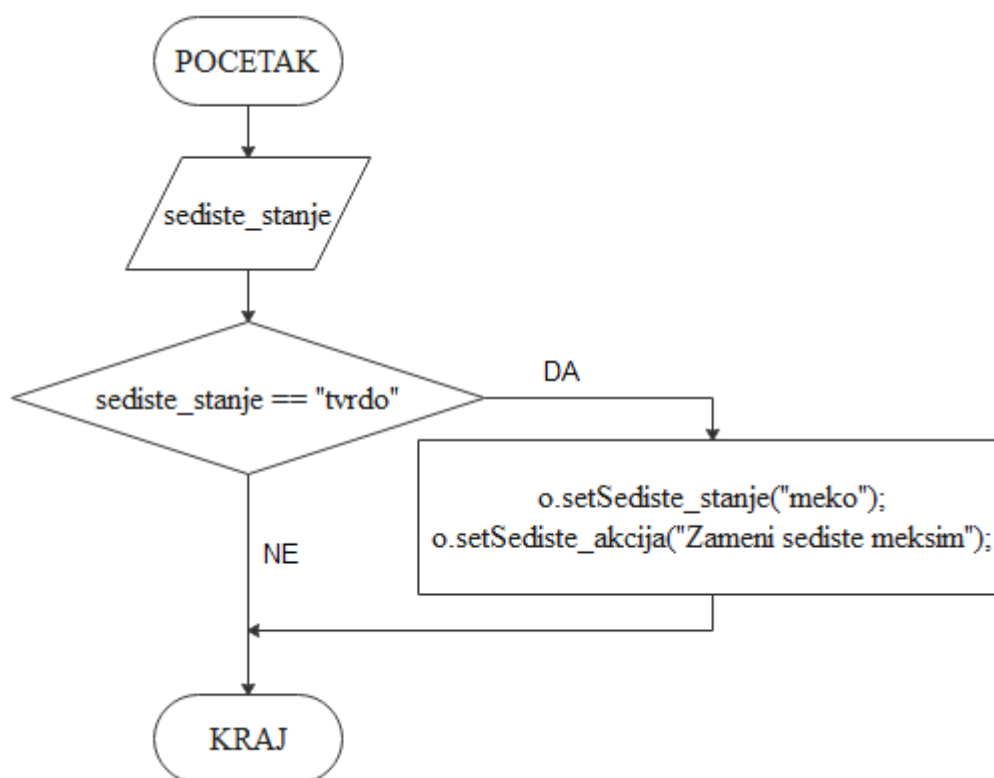
Slika 21 Stablo odlučivanja za problem "Zadnji menjač"



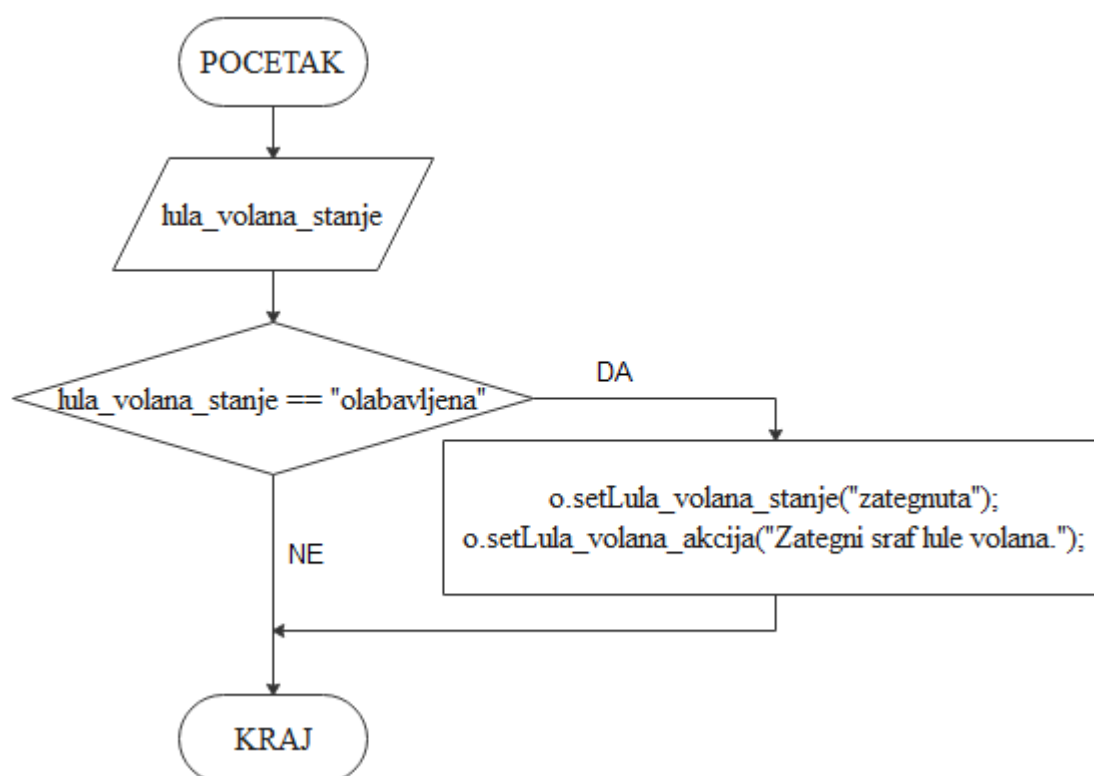
Slika 22 Stablo odlučivanja za problem "Kaseta"



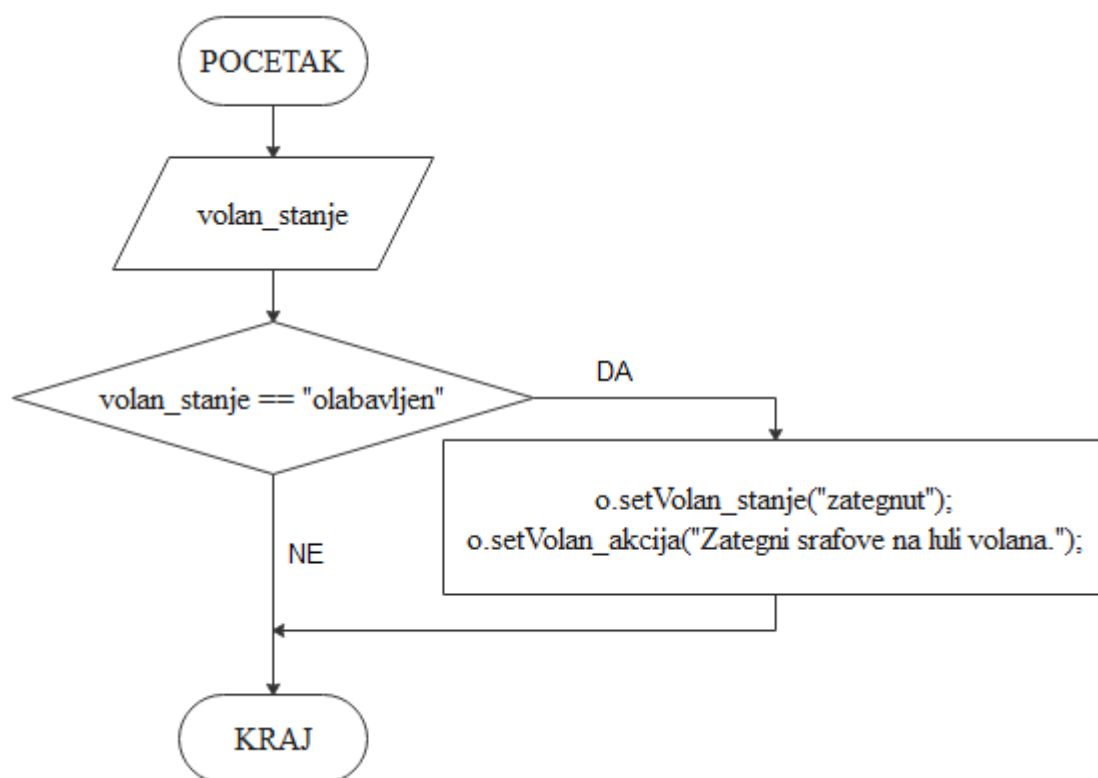
Slika 23 Stablo odlučivanja za problem "Šticna"



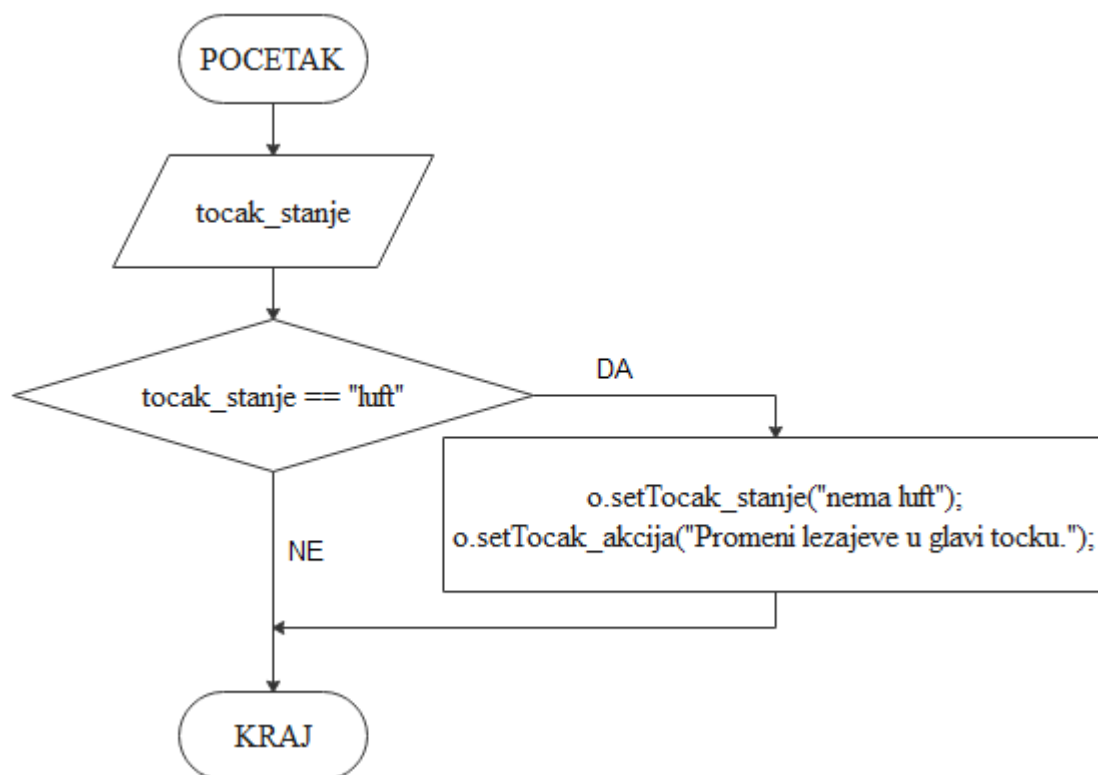
Slika 24 Stablo odlučivanja za problem "Sedište"



Slika 25 Stablo odlučivanja za problem "Lula volana"



Slika 26 Stablo odlučivanja za problem "Volan"



Slika 27 Stablo odlučivanja za problem "Točak"

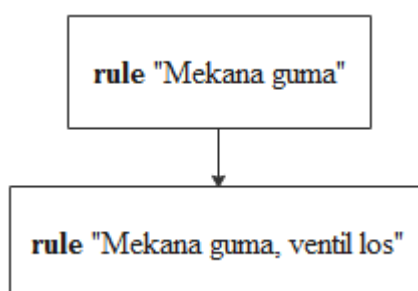


### 3.6 Ulančana aktivacija pravila

Pojam "ulančana aktivacija pravila" predstavlja aktivaciju više pravila ES, u zavisnosti od prosledjenih ulaza, gde se obradom tih pravila dobijaju odgovarajući izlazi.

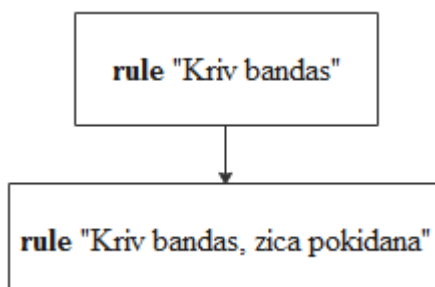
Ispod su data objašnjenja i prikazana ona pravila koja učestvuju u ulančavanoj aktivaciji pravila i to najviše 42 pravila, tj. ona pravila koja je moguće aktivirati preko korisničkog interfejsa, od mogućih 60 pravila.

U zavisnosti od odgovarajućih ulaza u ES, može se aktivirati pravilo „Mekana guma“ i pravilo „Mekana guma, ventil los“ i u tom slučaju, znači da je došlo do aktivacije više pravila, što predstavlja ulančanu aktivaciju pravila, slika ispod.



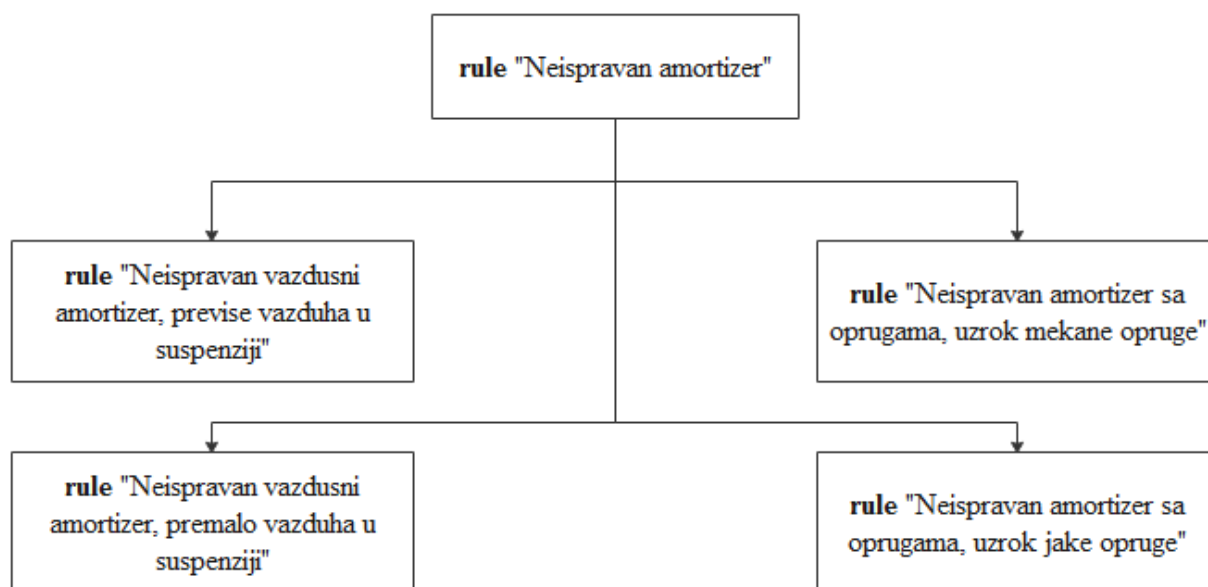
Slika 28 Ulančana aktivacija pravila za problem "Mekana guma"

U zavisnosti od odgovarajućih ulaza u ES, može se aktivirati pravilo „Kriv bandas“ i pravilo „Kriv bandas, zica pokidana“ i u tom slučaju, znači da je došlo do aktivacije više pravila, što predstavlja ulančanu aktivaciju pravila, slika ispod.



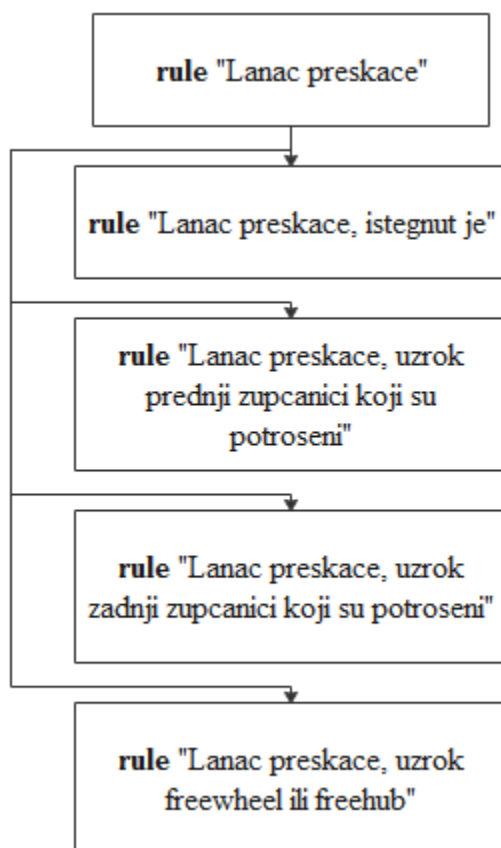
Slika 29 Ulančana aktivacija pravila za problem "Kriv bandas"

U zavisnosti od odgovarajućih ulaza u ES, može se aktivirati pravilo „Neispravan amortizer“ i samo jedno od pravila koja su prikazana na slici ispod i u tom slučaju, znači da je došlo do aktivacije više pravila, što predstavlja ulančanu aktivaciju pravila.



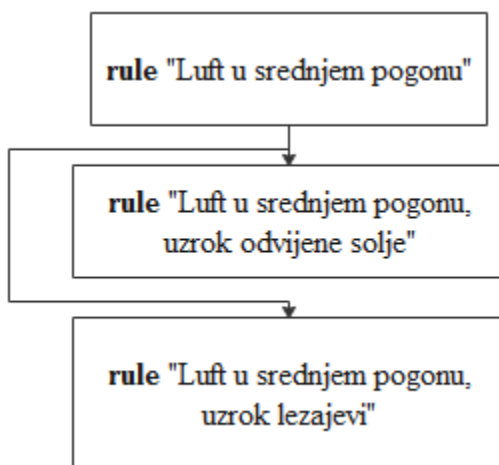
Slika 30 Ulančana aktivacija pravila za problem "Neispravan amortizer"

U zavisnosti od odgovarajućih ulaza u ES, mogu se aktivirati pravila koja su prikazana na slici ispod i u tom slučaju, znači da je došlo do aktivacije više pravila, što predstavlja ulančanu aktivaciju pravila.



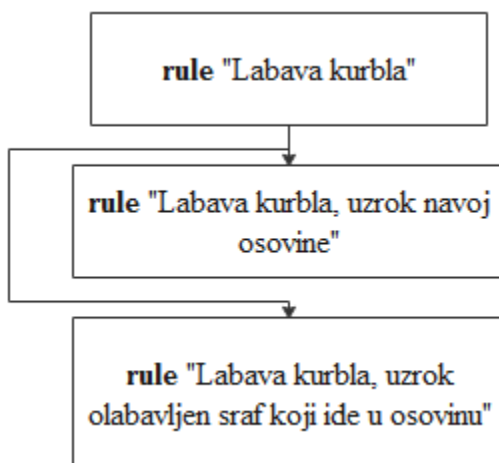
Slika 31 Ulančana aktivacija pravila za problem "Lanac preskace"

U zavisnosti od odgovarajućih ulaza u ES, mogu se aktivirati pravila koja su prikazana na slici ispod i u tom slučaju, znači da je došlo do aktivacije više pravila, što predstavlja ulančanu aktivaciju pravila.



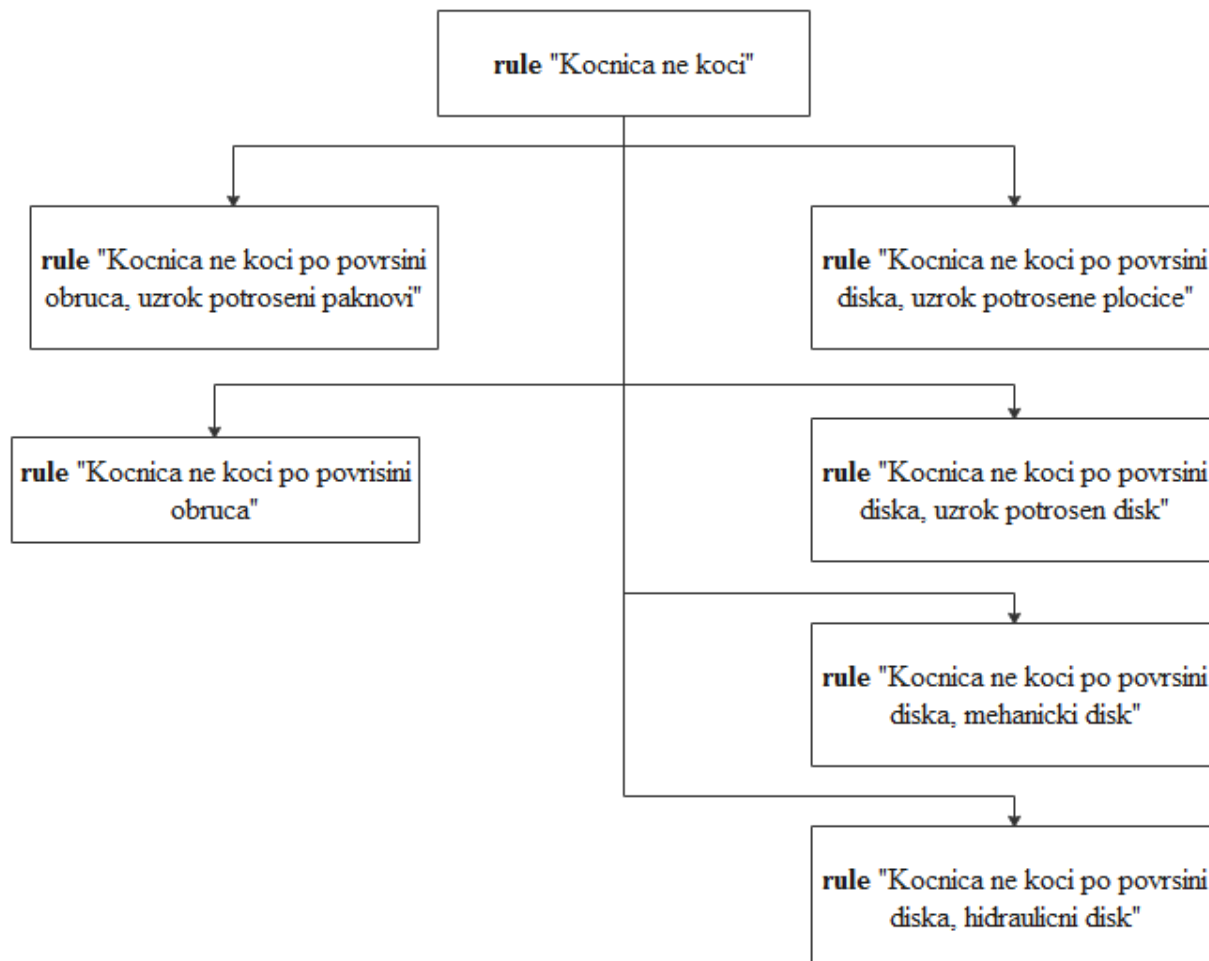
Slika 32 Ulančana aktivacija pravila za problem "Luft u srednjem pogonu"

U zavisnosti od odgovarajućih ulaza u ES, mogu se aktivirati pravila koja su prikazana na slici ispod i u tom slučaju, znači da je došlo do aktivacije više pravila, što predstavlja ulančanu aktivaciju pravila.



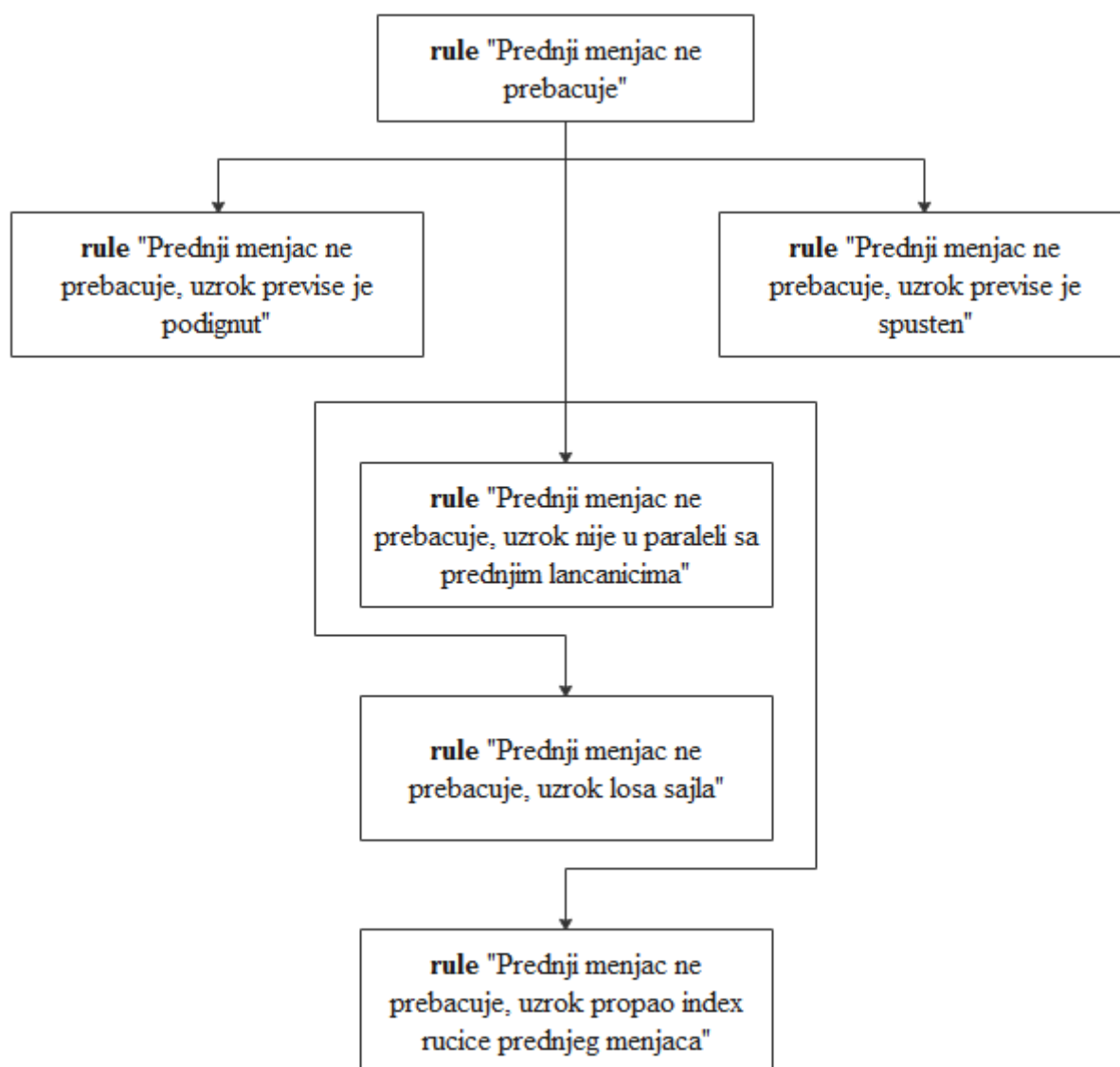
Slika 33 Ulančana aktivacija pravila za problem "Labava kurbla"

U zavisnosti od odgovarajućih ulaza u ES, može se aktivirati pravilo „Kocnica ne koci“ i leva ili desna polovina pravila, koja su prikazana na slici ispod i to sa izuzetkom da može da aktivira u desnoj polovini samo jedno od dva pravila ("Kocnica ne koci po površini diska, mehanicki disk" ili "Kocnica ne koci po površini diska, hidraulicki disk") i u tom slučaju, znači da je došlo do aktivacije više pravila, što predstavlja ulančanu aktivaciju pravila.



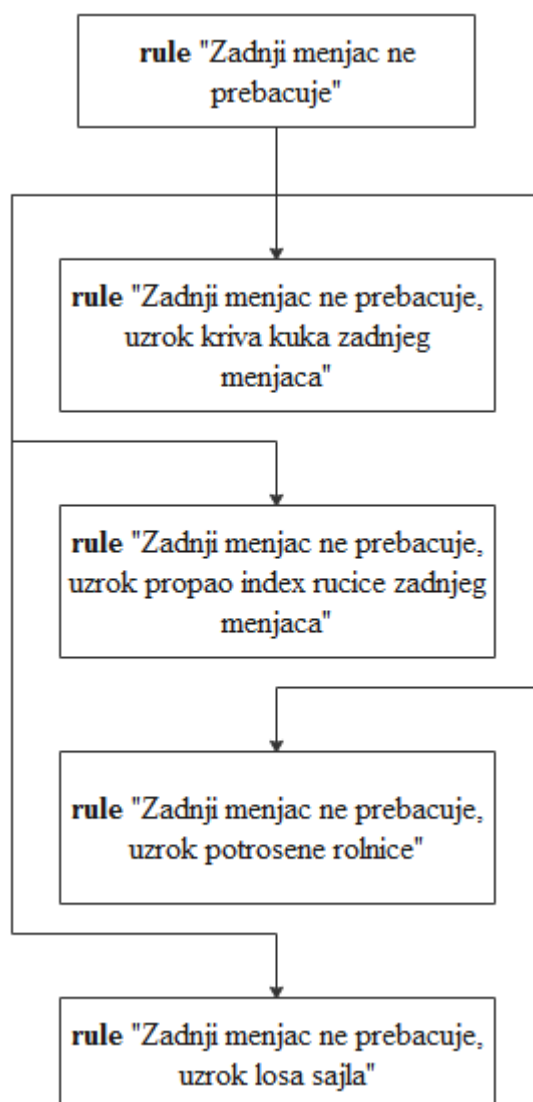
Slika 34 Ulančana aktivacija pravila za problem "Kocnica ne koci"

U zavisnosti od odgovarajućih ulaza u ES, mogu se aktivirati pravila, koja su prikazana na slici ispod i to sa izuzetkom da može da se aktivira samo jedno od dva pravila ("Prednji menjac ne prebacuje, uzrok previse je podignut" ili "Prednji menjac ne prebacuje, uzrok previse je spusten") i u tom slučaju, znači da je došlo do aktivacije više pravila, što predstavlja ulančanu aktivaciju pravila.



Slika 35 Ulančana aktivacija pravila za problem "Prednji menjac ne prebacuje"

U zavisnosti od odgovarajućih ulaza u ES, mogu se aktivirati pravila koja su prikazana na slici ispod i u tom slučaju, znači da je došlo do aktivacije više pravila, što predstavlja ulančanu aktivaciju pravila.



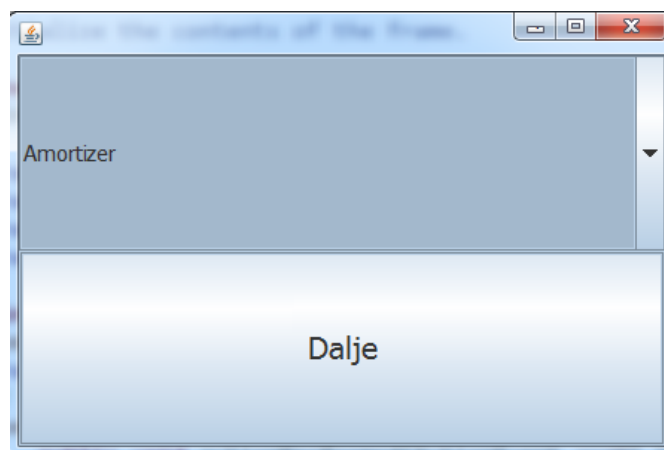
Slika 36 Ulančana aktivacija pravila za problem "Zadnji menjac ne prebacuje"

### 3.7 Izlaz

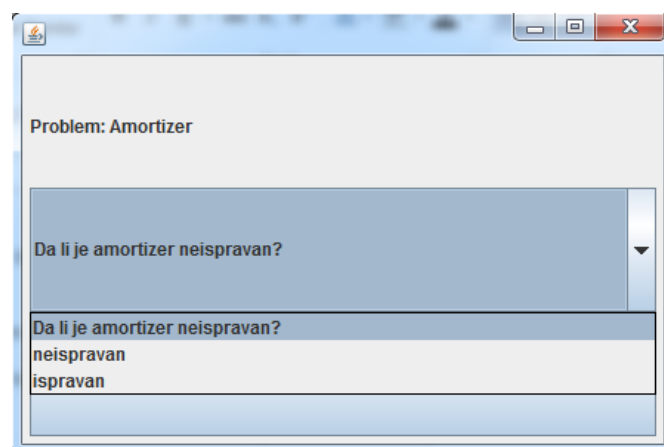
Kao izlaz, ES sistem nam daje na znanje šta trebamo da uradimo kako bi smo rešili odredjeni problem koji se javio kod odredjenog dela bicikla.

Da bi testirali ES, potrebno je pokrenuti klasu „Frame1“ komandom *“Run as Java application”*.

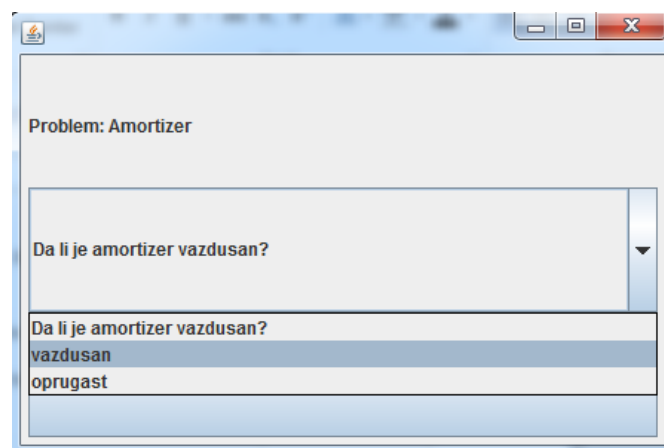
Rezultat korišćenja sistema prikazan je na slikama ispod.



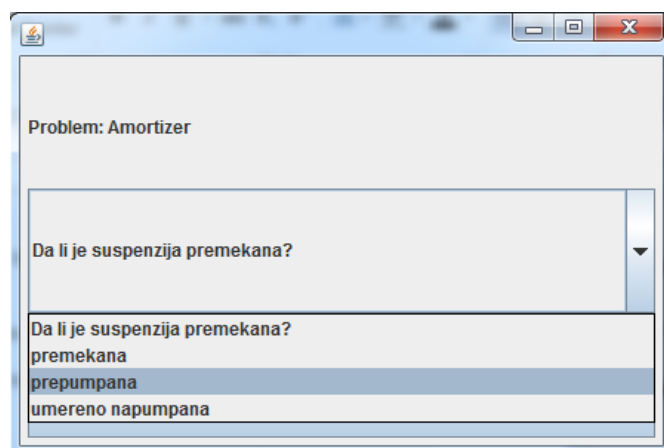
Slika 37 Prikaz rada GUI aplikacije, I deo



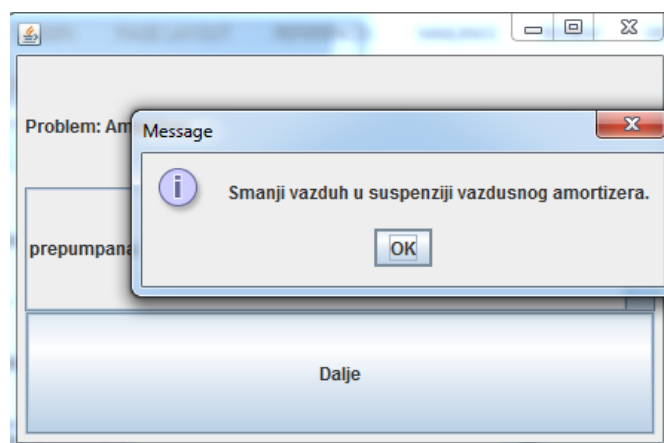
Slika 38 Prikaz rada GUI aplikacije, II deo



Slika 39 Prikaz rada GUI aplikacije, III deo



Slika 40 Prikaz rada GUI aplikacije, IV deo



Slika 41 Prikaz rada GUI aplikacije, V deo, rešenje



## 4. Zaključak

Ekspertski sistem je računarski program kojim se rešava problem na način na koji to čini ekspert, odnosno čovek iz date oblasti.

Ekspertski sistem se sastoji od: baze znanja, radne memorije i mehanizma zaključivanja.

Baza znanja je specijalizovana i jedinstvena za konkretni sistem koji sadrži znanje eksperata iz određene oblasti.

Radna memorija predstavlja memoriju u kojoj se nalaze trenutni podaci o problemu koji se rešava

Mehanizam zaključivanja na osnovu trenutnih podataka i baze znanja rešava problem.

U razvoju ekspertskog sistema učestvuju ekspert, inženjer i korisnik.

Inženjer prikuplja heurističko znanje od jednog ili više eksperata i na osnovu toga dizajnira i implementira ES, koji kasnije i održava.

Ekspert poseduje znanje i iskustvo na osnovu kojih rešava probleme iz određenog domena.

Korisnik radi sa ekspertskim sistemom, tako što definiše zahteve vezane za korisnički interfejs sistema.

Programi pisani u Eclipse razvojnom okruženju su Java fajlovi koje kreiramo za potrebe našeg projekta. Uobičajena praksa je da za svaki Java fajl, zadužen za određeni posao, kreiramo pakete u kojima će se ovi fajlovi nalaziti.

Glavni Java fajl koji predstavlja main klasu importuje sve potrebne klase kako bi imao pristup vlasništvima tih klasa, metodama, koristeći komandu **import**.

Za kreiranje pravila u .drl fajlu mora biti ispoštovana određena sintaksa:

**rule** "naziv"

//atributi pravila

**when**

//uslov

**then**

//zaključak

**end**

Naziv pravila može biti bilo kakav niz slova, ali obavezno mora biti pod znacima navoda.

Atributi pravila su opcioni i utiču na ponašanje pravila.

Uslov pravila sadrži jedan ili više logičkih iskaza koji liče na logičke izraze u Javi.

Zaključak pravila se daje u formi običnih Java naredbi i u tom delu se setuju promenljive koje predstavljaju rešenja ES.

Ulančana aktivacija pravila predstavlja aktivaciju više pravila ES, u zavisnosti od prosledjenih ulaza, gde se obradom tih pravila dobijaju odgovarajući izlazi.

## 5. Literatura

1. ekspertski sistemi, četvrta godina, prvi semestar, računarska tehnika i softversko inženjerstvo, moodle portal: <http://moodle.fink.rs>, 04.01.2021 (14:00)
2. <https://www.youtube.com/watch?v=oeswfZz4IW0> , 04.01.2021 (14:01)
3. <https://www.youtube.com/watch?v=r8Qiz9Bn1Ag> , 04.01.2021 (14:02)