

**UNIVERZITET CRNE GORE**  
**Elektrotehnički fakultet, Podgorica**

Materijal sa drugog termina predavanja iz

**EKSPERTNIH SISTEMA**

**EKSPERTNI SISTEMI – POJAM I ARHITEKTURA**  
**PREDSTAVLJANJE ZNANJA**

**Prof. dr Vesna Popović-Bugarin**

Podgorica, 2015.

# Ekspertni sistemi – pojam i arhitektura

## 2.1 Pojam ekspertnih sistema

Ekspertni sistemi su jedna od oblasti koje pokriva VI. Jedna od ranih definicija ekspertnih sistema je [1]: „Ekspertni sistem je inteligentan kompjuterski program koji koristi znanje i procedure zaključivanja kako bi riješio probleme dovoljno teške da zahtijevaju značajnu ljudsku ekspertizu za svoje rješavanje“. Dakle, ekspertni sistem je program koji emulira svojstvo ljudskog eksperta da donosi odluke. Termin emulira se koristi da bi označio namjeru da ekspertni sistem djeluje u svakom pogledu kao ljudski ekspert. Emulacija je mnogo zahtijevnija od simulacije, kod koje se zahtijeva djelovanje poput onoga šta se simulira, i to samo u nekim aspektima.

Ekspertni sistemi su grana VI u kojoj se vrši opsežna upotreba specijalizovanih znanja kako bi se problemi riješili na nivou na kojem bi ih riješio ekspert. Pod ekspertom se podrazumijeva čovjek koji posjeduje ekspertizu u određenoj oblasti, odnosno znanja ili specifične sposobnosti koje nisu poznate ili dostupne većini ljudi. Ekspert može riješiti probleme koje većina ljudi ne može da riješi ili ih ekspert rješava mnogo efikasnije.

Ukoliko bi se sistematizovali zahtjevi koji se postavljaju da bi se neko smatrao ekspertom, moglo bi se reći da su osnovna svojstva eksperta da [2]:

- Na optimalan način primijeni svoja znanja u rješavanju problema. Pri tome se podrazumijeva uzimanje u obzir činjenica i predviđanje relevantnih posljedica;
- Objasni i obrazloži svoje odluke i prijedloge;
- Komunicira sa drugim ekspertima i proširuje svoja znanja, prestrukturira i reorganizuje shvatanja i znanja;
- Formira i napušta određene zaključke, što dokazuje da je pronikao u suštinu određenih pojava i našao nove zakonitosti koje među njima vladaju;
- Određuje najbrži način dolaska do rješenja i njegove praktične primjene;
- U specifičnim situacijama intuitivno (heuristički), na osnovu svih dosadašnjih iskustava i događaja, ocijeni gdje se nalazi rješenje problema.

Ključni faktor za dobre performanse ekspertnog sistema je kvalitet znanja koje je u njega ugrađeno. Smatra se da se u slučaju da ekspertni sistem ne daje adekvatne odgovore, intervencije treba vršiti nad ugrađenim znanjem. Znanje u ekspertnom sistemu može biti znanje koje je generalno dostupno u knjigama, časopisima itd. i/ili ekspertiza. Dakle, može se reći da se razlikuju dva tipa znanja:

- Eksplicitno znanje – Znanje koje se zove činjenicama datog domena, odnosno znanje koje je široko poznato i nalazi se napisano u udžbenicima, časopisima i slično;
- Implicitno znanje – heurističko znanje – Ono znanje koje čovjek – ekspert gradi na osnovu iskustva i koje kombinovano sa prvim tipom znanja čini čovjeka ekspertom. Naime, heuristička znanja ne garantuju uspjeh prilikom rješavanja problema, kao što garantuje algoritamski način rješavanja problema. Naime, heuristička znanja, stečena na osnovu iskustva, mogu pomoći u konkretnom slučaju, ali nije garantovano da će uvijek funkcionisati. Ipak, u velikom broju oblasti kao što je medicina i inženjerstvo, heurizam igra ključnu

ulogu u rješavanju određene vrste problema. Čak i kada postoje tačna rješenja, i poznata su, njihovo korišćenje može, na osnovu iskustva, biti nepraktično po pitanju cijene i vremena koje zahtijeva njihova realizacija.

Često se zbog značaja znanja koje je ugrađeno u njih kao sinonimi za ekspertne sisteme koriste termini: sistemi zasnovani na znanju ili ekspertni sistemi zasnovani na znanju. Ipak, većina ljudi koristi termin ekspertni sistem, zato što je kraći, iako u njihovim ekspertnim sistemima često i nema ekspertize, već samo generalnog – eksplicitnog znanja. Kada su, 1970. godine, razvijeni prvi ekspertni sistemi, oni su sadržali isključivo ekspertno znanje. Ipak, termin ekspertni sistem se danas često koristi za bilo koji sistem koji je zasnovan na tehnologiji ekspertnih sistema. Tehnologija ekspertnih sistema uključuje specijalizovane jezike za razvoj ekspertnih sistema, programe i hardver koji je dizajniran da pomogne razvoju i izvršavanju ekspertnog sistema.

Ekspertni sistemi se razlikuju prema vrsti korisnika:

- Najčešće korisnik primijenjuje ekspertizu sistema na određeni zadatak;
- Neki ekspertni sistemi, kao što su sistemi medicinske dijagnostike, uključuju znanje grupe eksperata u cilju korišćenja od strane jednog eksperta iz iste grupe. Drugim riječima, ljekari kreiraju sistem za ljekare. Može se reći da ovakvi ekspertni sistemi djeluju kao inteligentni savjetnik/pomoćnik ekspertu.
- Neki ekspertni sistemi prenose znanje jedne grupe eksperata grupi ili pojedincu koji to nijesu. Ovakvi ekspertni sistemi se često nazivaju inteligentnim tutorima.

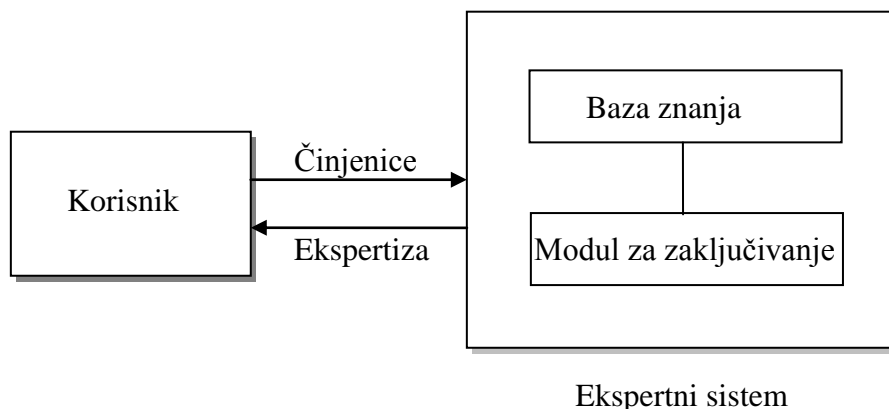
Bez obzira na vrstu krajnjeg korisnika, ekspertni sistemi posjeduju znanje o rješavanju specifičnih problema u određenoj oblasti – **domen znanja ekspertnog sistema**. U domenu znanja ekspertni sistem treba da donosi zaključke na isti način kao što bi činio ekspert prilikom rješavanja zadatog problema. Dakle, ekspertni sistem zahtijeva i postupak zaključivanja - metod rasuđivanja, korišćen da napravi spregu između znanja koje se čuva u ekspertnom sistemu i problema koji postavlja korisnik [2].

## **2.2 Opšt koncept ekspertnih sistema**

Osnovni koncept ekspertnog sistema zasnovanog na znanju je prikazan na slici 1. Korisnik obezbijeduje činjenice ili druge informacije ekspertnom sistemu, a kao odgovor prima savjet ekspertnog sistema. Jezgro ekspertnog sistema se sastoji od dvije glavne komponente: baze znanja i relacionog modula ili modula za zaključivanje. Pojednostavljeno se može reći da baza znanja sadrži znanja na osnovu kojih modul za zaključivanje donosi zaključke koji predstavljaju odgovor na korisnikovo pitanje.

Kod razvoja ekspertnih sistema se javlja niz veoma krupnih pitanja, na koje treba dati odgovor [2]. Prvi problem koji se susrijeće kod ekspertnih sistema je način predstavljanja znanja. Kako predstaviti znanje iz datog domena u obliku pogodnih struktura podataka, tako da se efikasno može iskoristiti u rješavanju problema? Drugo, postavlja se pitanje kako koristiti znanje, kako projektovati mehanizam zaključivanja da bi se znanje efikasno koristilo u rješavanju problema? Treće, postavlja se pitanje akvizicije znanja, to jest, kako preuzeti znanje od eksperata i unijeti ga u računar? Da li je moguće automatizovati korak akvizicije znanja i obezbijediti neposrednu

komunikaciju eksperta i računara i nesmetan prenos znanja od eksperta ka računaru? U ovom trenutku, akvizicija znanja predstavlja jedno od ključnih pitanja u razvoju VI.



**Slika 1 Osnovni koncept ekspertnog sistema**

Znanje ugrađeno u ekspertni sistem može biti predstavljeno na više načina – ono može biti enkapsulirano u objektima (činjenicama) i pravilima. Čest metod za predstavljanje znanja je u obliku AKO ... TADA tipa pravila, kao na primjer:

*AKO svijetlo je crveno TADA stani*

Ukoliko postoji činjenica da je svijetlo crveno, ovo odgovara šablonu „svijetlo je crveno“. Pravilo je zadovoljeno i izvršava se akcija „stani“. Iako je ovaj primjer dosta jednostavan, veliki broj ekspertnih sistema je građen, i još uvijek se gradi, predstavljanjem znanja eksperata u obliku pravila.

Ovo je jedan od načina za predstavljanje znanja. Predstavljanje znanja će biti detaljnije rađeno u toku kursa.

Danas postoji veliki broj malih sistema koji su sagrađeni sa svega nekoliko stotina pravila. Ovi mali sistemi ne rade na nivou eksperta, ali su dizajnirani korišćenjem tehnologije ekspertnih sistema kako bi se njima izvršili zadaci koji zahtijevaju veliki obim znanja. Za ove male sisteme se najčešće koristi eksplicitno znanje.

Klasični ekspertni sistemi su zasnovani na implicitnom znanju, nepisanom znanju koje mora na neki način biti izvučeno-ekstrahovano od eksperta mnogobrojnim intervjuima u toku dužeg vremenskog perioda. Ove intervju sprovodi osoba koja i razvija – gradi ekspertni sistem i ona se naziva **inženjer znanja**. Proces izgradnje ekspertnog sistema se naziva **inženjerstvo znanja**. Inženjerstvo znanja obuhvata skup metoda i postupaka koje se odnose na prikupljanje, računarsko predstavljanje i memorisanje, kao i upotrebu ljudskog znanja u rješavanju složenih problemskih situacija.

Može se zaključiti da u razvoju ekspertnog sistema postoji nekoliko osnovnih faza. Na početku, inženjeri znanja stupaju u kontakt sa ekspertima kako bi intervjuima izvršili ekstrakciju njihovog znanja. Inženjeri znanja nakon toga kodiraju dobijeno znanje eksplicitno u bazu znanja. Ekspert procjenjuje rezultujući ekspertni sistem i upućuje sugestije i kritike inženjeru znanja, a u cilju poboljšanja ekspertnog sistema. Ovaj postupak se ponavlja sve dok eksperti ne ocijene da su performanse ekspertnog sistema na zadovoljavajućem nivou. Naravno, ocjenu ekspertnog sistema vrši i krajnji

korisnik. Osnovna šema ekspertnih sistema nezamisliva je bez korisnika, jer ekspertni sistem ima smisla samo ako se može praktično primijeniti. Testiranje ekspertnog sistema, u svim fazama razvoja, se provodi uz pomoć korisnika.

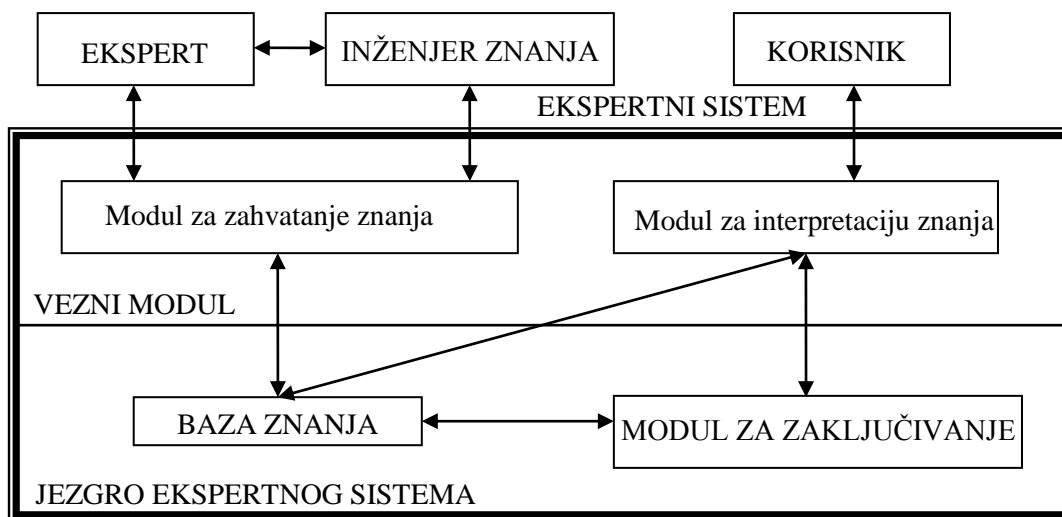
Ekspertni sistemu su generalno dizajnirani drugačije od konvencionalnih programa. Naime, problem koji se rješava ekspertnim sistemom najčešće nema rješenje koje se može predstaviti algoritmom. Dakle, rješavanje je zasnovano je na zaključivanju kako bi se dobila razumna rješenja. Ovdje se ne misli na zaključivanja koja vrši inženjer znanja u toku razvoja ekspertnog sistema, već na zaključivanja za koja se ekspertni sistem mora osposobiti. Razumna rješenja su najbolje što se može očekivati, ukoliko ne postoji algoritam koji bi obezbijedio optimalno rješenje. S obzirom da je ekspertni sistem zasnovan na zaključivanju, on mora biti sposoban da objasni način na koji je došao do zaključaka, tako da njegovo rezonovanje može biti provjereno. **Podsystem za objašnjavanje** je sastavni dio sofisticiranih ekspertnih sistema.

Neki ekspertni sistemi mogu dozvoljavati sistemu da uči pravila na osnovu primjera, kroz **indukciju pravila**, kojom sistem kreira pravila na osnovu tabele podataka.

## 2.3 Arhitektura ekspertnih sistema

Da bi se izvršiti postavljeni zadaci, ekspertni sistem je veoma složen programski paket. Može se reći da se sastoji od niza manjih programskih cjelina ili modula. Dva osnovna dijela su [2]:

- vezni modul ili korisnički interfejs,
- jezgro ekspertnog sistema.



Slika 2 Osnovna struktura ekspertnih sistema

**Jezgro ekspertnog sistema** se sastoji od dva glavna dijela i to:

- baze znanja,
- relacionog modula ili modula za zaključivanje; kao i
- radne memorije.

**Vezni modul** spaja eksperta i inženjera znanja, sa jedne strane, i korisnika, sa druge strane, sa bazom znanja i relacionim modulom. Iz tog razloga vezni modul se sastoji od dva manja modula (komunikaciona kanala) i to:

- modula za zahvatanje znanja,
- modula za interpretaciju znanja.

Nakon povezivanja ovih modula u jednu cjelinu, dobija se osnovna struktura ekspertnog sistema, koja se može prikazati slikom 2.

U mnogim ekspertnim sistemima postoji i **modul za akviziciju znanja**. Ovdje se pod akvizicijom znanja ne podrazumijeva prosto zahvatanje znanja od eksperta, već samostalno učenje ekspertnog sistema. Već je rečeno da neki ekspertni sistemi mogu učiti indukovanjem pravila kroz primjere, odnosno mogu automatski generisati pravila. Ipak, generalna znanja koja konstruiše inženjer znanja su mnogo kompleksnija od jednostavnih pravila dobijenih indukcijom.

### 2.3.1 Jezgro ekspertnog sistema<sup>1</sup>

Kvalitetno razumijevanje ekspertnog sistema podrazumijeva dobro upoznavanje sa njegovim jezgrom. Iz tog razloga, u najopštijim crtama, će biti prikazano jezgro ekspertnog sistema, odnosno princip rada baze znanja i relacionog modula ili modula za zaključivanje.

#### Baza znanja

Pojavom vještačke inteligencije, a posebno razvojem ekspertnih sistema, javljaju se kao svojevrsan razvoj baza podataka i obrade podataka pojmovi baze znanja i procesiranje znanja. Baza znanja predstavlja skladište primitivnih znanja (osnovna fakta, proceduralna pravila, itd.) raspoloživih sistemu. U opštem slučaju znanje je smješteno u obliku fakata-činjenica i pravila, ali sama organizacija tog znanja značajno varira.

Projektovanje šeme predavljanja znanja utiče i na projektovanje softverske mašine za zaključivanje, procesa ažuriranja znanja i formiranja objašnjenja, a time ukupnu efikasnost sistema. **Izbor načina za predavljanje znanja je jedna od najkritičnijih odluka u projektovanju ekspertnih sistema.**

#### Radna memorija

**Radna memorija** je sastavni dio ekspertnih sistema koji sadrži sve informacije o problemu koji se trenutno rješava, bilo da su dobijene od korisnika, izvedene od strane sistema ili povučene iz baze znanja. Celokupne informacije dobijene za vreme konsultacije sa korisnikom se često nazivaju kontekst sesije. Za vrijeme konsultacije ekspertnog sistema, korisnik unosi informacije o tekućem problemu u radnu memoriju. Sistem uparuje ove informacije sa znanjem sadržanim u bazi znanja da bi došao do novih činjenica. Sistem zatim unosi ove nove činjenice u radnu memoriju i proces uparivanja se nastavlja. Eventualno, sistem dolazi do nekog zaključka u vidu pravila, koje se takođe unosi u radnu memoriju.

Radna memorija može sadržati i premise i zaključke pravila koja su zadovoljena i aktivirana za izvršavanje. Tipično, u radnoj memoriji se nalaze indeksi koji su povezani sa pravilom, slično indeksima korišćenim u relacionim bazama podataka, kako bi se ubrzalo pretraživanje po radnoj memoriji.

---

<sup>1</sup> Dio teksta ja zasnovan na materijalu dostupnom u [5]

## Procesiranje znanja – modul za zaključivanje

Prilikom rješavanja postavljenih zadataka, ekspertni sistem može naići na promjenljive situacije, odnosno na pitanje čiji odgovor nema dat eksplicitno u bazi znanja. U ovakvim slučajevima potrebno je da ekspertni sistem bude sposoban da izvede novo znanje iz postojećeg. U slučaju kada prilikom rješavanja postavljenog zadatka, ekspertni sistem nailazi na poznatu situaciju, odnosno pitanje čiji odgovor ima dat eksplicitno u bazi podataka, potrebno je samo da primijeni odgovarajuće znanje. Dakle, zadatak modula za zaključivanje je ili nalaženje zahtijevanog znanja kao postojećeg ili njegovo izvođenje iz postojećeg znanja. **Proces traženja odgovarajućeg znanja i izvođenje iz njega novog znanja je ključni element ekspertnog sistema.** Modul za zaključivanje je softverski sistem koji pronalazi znanje i izvodi novo znanje iz osnovnog znanja. On funkcioniše tako što činjenice iz baze znanja kombinuje sa informacijama dobijenim od korisnika u cilju izvođenja specifičnih zaključaka. Pri radu se koriste kontrolne strategije, koje odlučuju u kom trenutku treba primijeniti neko od pravila iz baze znanja na nove činjenice dobijene tokom konsultovanja sa korisnikom. Na ovaj način se simulira ljudsko razmišljanje. Primjer izvođenja novog znanja iz postojećeg bi bio, kada bi se iz:

*Petar je otac Zoranu*

*Zoran je otac Milanu*

izvelo znanje:

*Petar je djeda Milanu*

Bitan element modula za zaključivanje je strategija pretraživanja - razvijanje zahtijevanog znanja. Modul za zaključivanje odlučuje koja je od hipoteza (šablona – pravila), i da li je ijedna, zadovoljena. Dva su osnovna koncepta zaključivanja zastupljena u većini ekspertnih sistema: povratno ulančavanje, koje predstavlja proces rezonovanja s vrha ka dnu, i direktno ulančavanje zasnovano na procesu rezonovanja od dna ka vrhu. Povratno ulančavanje je rezonovanje od činjenica prema zaključku koji slijedi iz tih činjenica. Na primjer, ako vidimo da pada kiša prije nego što napustimo kuću (činjenica) trebamo ponijeti kišobran (zaključak). Direktno ulančavanje uključuje obratno rezonovanje, počinje se od hipoteza, da bi se potencijalni zaključak dokazao, i ide se ka činjenicama koje potvrđuju postavljene hipoteze. Na primjer, ukoliko ne pogledamo napolje prije izlaska iz kuće, a neko uđe u kuću sa mokrim cipelama i kišobranom, naša hipoteza je da vani pada kiša. Da bi se potvrdila ova hipoteza, može se pitati osoba koja je ušla u kuću da li napolju pada kiša. Ukoliko je odgovor potvrđan, naša hipoteza je dokazana i postaje činjenica, na osnovu koje se može izvesti zaključak da je potrebno ponijeti kišobran. Naime, hipoteze se mogu posmatrati kao činjenice čija je istinitost pod pitanjem i tek treba da se utvrdi, odnosno kao cilj koji je neophodno potvrditi. **Izbor paradigme zaključivanja, uz moguću kombinatorijalnu eksploziju, jako utiče na ukupne performanse ekspertnog sistema.**

Modul za zaključivanje se sastoji od:

- *uparivača šablona* – odlučuje koje pravilo da aktivira s obzirom na sadržaj radne memorije. Ovo je suštinski težak zadatak. Ukoliko svako pravilo ima par premisa i radna memorija sadrži stotine činjenica, uparivač šablona bi mogao da pretražuje milione kombinacija činjenica, kako bi pronašao one koje zadovoljavaju pravila. Na sreću, ogromna pažnja istraživača je posvećena ovom problemu u pronađen je veliki broj efikasnih algoritama za

pretraživanje. Ipak, za većinu ekspertnih sistema, uparivač šablona je još uvijek najskuplji dio.

- *agende* – kada uparivač šablona utvrdi koje pravilo treba biti aktivirano, preostaje da se odredi redosljed izvršavanja aktiviranih pravila. Lista pravila koja su aktivirana se nalazi u agendi. Zadatak agende je upotreba strategija za rješavanje konflikata, kako bi se odlučilo koje od aktiviranih pravila ima najveći prioritet, i treba biti izvršeno prvo. I ovo je potencijalno veliki problem, i skoro svaki ekspertni sistem ima svoj sopstveni pristup. Uobičajeno, strategija rješavanja konflikata mora uzeti u obzir specifičnost ili kompleksnost svakog pravila i relativnu starost činjenica koje zadovoljavaju premise u agendi. Pravila takođe mogu imati specificirani prioritet, povezan sa njima, tako da su neka pravila važnija i moraju se uvijek izvršiti prije nego neka druga. Kao primjer se mogu posmatrati dva pravila koja bi se koristila u programu za vožnju auta od strane robota. Prvo bi bilo: *AKO svijetlo je zeleno TADA kreni*, dok bi drugo bilo *AKO čovjek je ispred tebe TADA stani*. Ukoliko je robot stao na crveno svijetlo, i upali se zeleno kada se neko još uvijek nalazi na pješačkom prelazu, oba se pravila aktiviraju. Važno je da se prvo izvrši drugo pravilo. Dakle, drugom se pravilu mora dati veći prioritet.
- mašine za izvršavanje – kada se utvrdi koje pravilo je potrebno prvo izvršiti, mašina za izvršavanje to i čini.

Grubo se može reći da modul za zaključivanje funkcioniše u diskretnim ciklusima:

1. Sva pravila (šabloni) se upoređuju sa radnom memorijom (koristeći uparivač šablona) da bi se odlučilo koja se pravila trebaju aktivirati u toku tekućeg ciklusa. Ova lista sa neutvrđenim redoslijedom izvršavanja aktiviranih pravila, zajedno sa pravilima aktiviranim u prethodnoj iteraciji naziva se *lista konflikata*.
2. Kada se utvrdi redosljed izvršavanja pravila, lista konflikata postaje *agenda* – lista pravila čija će desna strana biti izvršena. Proces određivanja redoslijeda izvršavanja pravila se naziva rješavanje konflikata. Strategija za rješavanje konflikata će zavisiti od velikog broja faktora, pri čemu će mali broj njih biti pod kontrolom programera.
3. Ciklus se završava izvršavanjem pravila koje je na prvom mjestu u agendi (što će vrlo vjerovatno promijeniti sadržaj radne memorije), i cijeli proces se ponavlja.

Problem procesiranja znanja i njegovo prikazivanje su osnova na kojoj se zasniva kompletna teorija vještačke inteligencije, pa tako i teorije izgradnje ekspertnih sistema. Ova problematika je posebno istraživana sredinom 70-tih godina, tako da se do danas razvio čitav niz metoda i programskih alata, gdje je sve bazirano na matematičkim disciplinama, i to: statistici i teoriji vjerovatnoće, matricama i teoriji grafova, običnoj, višedimenzionalnoj i tzv. "neizrazitoj" (fuzzy) logici, predikatskom računu, itd.

Vremenom se izdvojio čitav niz metoda, koje danas dominiraju u izgradnji jezgra ekspertnih sistema:

- automatsko dokazivanje teorema,
- ekspertni sistemi zasnovani na matematičkoj logici,
- semantičke mreže,



- okviri znanja,
- produkcionni sistemi,
- metode fazi ekspertnih sistema,
- metode za izgradnju ekspertnih sistema zasnovane na neuronskim mrežama,
- genetički algoritmi i ekspertni sistemi,
- agenti, multi agenti i inteligentni agenti,
- inteligentne baze podataka i inteligentni informacijski sistemi.

Iako ove metode imaju niz dodirnih tačaka i određena preklapanja, one predstavljaju zaokružene cjeline.

### 2.3.2 Vezni modul<sup>2</sup>

Pristup do jezgra u izgrađenom ekspertnom sistemu je moguć isključivo preko veznog modula. Vezni modul je cjelina od dva svojevrsna komunikaciona kanala i to:

- modula za interpretaciju znanja, predviđenog za korisnika, i
- modula za zahvatanje znanja, koga koriste ekspert i inženjer znanja.

Modul za interpretaciju znanja je dio koji omogućava dijalog između donosioca odluke (korisnika) i sistema. Sa jedne strane služi da korisnik u toku rada sistema dostavi informacije koje sistem iz baze znanja nije uspio da dobije, a sa druge strane omogućava korisniku da za svaku odluku ekspertnog sistema traži dodatno objašnjenje o tome koji su ga zaključci vodili da donese takvu odluku. Objašnjenje se uglavnom sastoji u identifikaciji koraka u procesu rezonovanja i davanja obrazloženja za svaki korak. Saopštavanje ovih podataka je u suštini podskup problema obrade prirodnih jezika. Sistem mora da pristupi znanjima koja su korišćena u zaključivanju, i da ih prevede u oblik prilagođen i shvatljiv za korisnika.

**Modul za interpretaciju znanja** se sastoji od sljedećih dijelova:

- *Programa za analizu saopštenja od strane korisnika sistemu* – Ima zadatak da prima poruke od korisnika. Njime se vrši sintaksna, semantička i morfološka analiza (odnosno, analiza poštovanja pravila jezika, analiza logičkog oblika poruke i analiza oblika poruke), drugim riječima, ovdje se vrši filtriranje poruke.
- *Podsistema za dijalog* – U podsistemu za dijalog prvo se izdvajaju tzv. ključne riječi, uz pomoć riječnika objekata i riječnika relacija, koje se zatim pretvaraju u oblik pogodan za predmetnu oblast. Ovako obrađena informacija se predaje jezgru ekspertnog sistema, koje je sada obrađuje prema zadatim pravilima, tj. postavlja dodatna pitanja, traži optimalna rješenja ili daje objašnjenja.
- *Programa za sintezu saopštenja od strane sistema korisniku* – Rezultat procesiranja znanja se predaje programu za sintezu saopštenja od strane sistema korisniku, koji vrši obradu po sintaksnim, semantičkim i morfološkim pravilima. Pri ovome se podrazumijeva da je čitav ekspertni sistem spreman za rad, da su već uneseni baza znanja i svi mehanizmi zaključivanja.

**Modul za zahvatanje znanja** je zadužen za zahvatanje znanja od eksperta. Može se razmatrati na niz različitih načina, po područjima nauke i tehnike, sa stanovišta logike i heuristike, sa teorijskog i praktičnog stanovišta, itd. Međutim, ova

---

<sup>2</sup> Veći dio teksta je preuzet iz [2]

jednostavna pretpostavka je jedan od rubnih kamenova kompletne teorije i prakse ekspertnih sistema. Postavlja se pitanje, kako popuniti bazu znanja i kako pravilno definisati modul za zaključivanje, odnosno kako „naučiti“ konkretni ekspertni sistem?

Proces pribavljanja znanja je veoma složen i može se rasčlaniti u nekoliko jasno izdvojenih postupaka. Ti postupci predstavljaju neku vrstu algoritma za sve koji izgrađuju bilo koji ekspertni sistem, odnosno ponavljaju obavljanje određenih zadataka. Zadaci bitni za zahvatanje znanja su:

1. Definisanje neophodnosti proširenja i modifikacije znanja;
2. Dobijanje potpuno novih znanja o sistemu;
3. Formiranje novih znanja u obliku koji sistem poznaje;
4. Usklađivanje starih i novih znanja i prelazak na korak 1.

Navedeni postupci su pokazali da nova generacija ekspertnih sistema mora biti zasnovana na izgradnji modula za zahvatanje znanja, koji omogućava **automatizaciju zahvatanja znanja** u ekspertnim sistemima.

Pored izgradnje modula za zahvatanje znanja i automatizacije zahvatanja znanja, pojam automatsko učenje predstavlja treću stepenicu u razvoju i pojedini autori je smatraju jedinim pravim početkom teorije i prakse ekspertnih sistema. Ovo mišljenje je bazirano na mogućnosti ekspertnog sistema da samostalno i automatski indukcijom pravila dolazi do potpuno novih zaključaka. Jasno je da i modul za zahvatanje znanja mora imati programe za analizu i sintezu saopštenja od strane eksperta, koji se obično nazivaju mehanizmi uvođenja informacija, da bi se dobio oblik pogodan za logičku obradu i kasniju predaju novih znanja i pravila jezgru ekspertnog sistema.

Međutim, induktivno učenje je samo jedna od mogućnosti. Ovdje je od bitnog značenja program koji se može uslovno nazvati **intelektualni redaktor**. Dužnost intelektualnog redaktora je da prihvati strukturu svih oblika znanja i da stalno obavlja poređenje novih znanja, od strane eksperta, i starih znanja, koja se nalaze u jezgru ekspertnog sistema.

Način na koji intelektualni redaktor obavlja ovu funkciju određuje i sam tip automatskog zahvatanja znanja, odnosno učenja:

- induktivno, ili posebna vrsta analognog,
- jezičko konceptualno,
- eliminaciono, uz pomoć stabla grananja,
- zvjezdasto,
- uz pomoć uzoraka, itd.

Svaka od ovih metoda ima svoje prednosti i nedostatke, a mogu se koristiti i njihove kombinacije.

Zajedničko za sve oblike i tipove zahvatanja znanja je tzv. opšti mehanizam učenja i samostalnog zaključivanja.

Da li se radi o induktivnom, jezičko konceptualnom i/ili nekom drugom obliku zahvatanja znanja, odlučuju modeli pojmova i/ili pravila. Očigledno, radi se o sistemu sa povratnom vezom, gdje se novostvoreni pojam i/ili pravilo ukomponuju samo ako su prošli dodatni test mehanizma zaključivanja i eksperta.

Jedna od važnih karakteristika ovog načina učenja je i stalni dijalog sa ekspertom. Sistem donosi svoje zaključke i postavlja pitanja svome "učitelju", te pokazuje svoja rješenja. Za ovakav tip konverzije zadužen je podsistem za dijalog.

Podsistem za dijalog, u širem smislu, u sebi sadrži i analizu i sintezu saopštenja, isto kao i modul za interpretaciju znanja. U zajednici sa intelektualnim redaktorom predstavlja modul za zahvatanje znanja.

## **2.4 Prednosti ekspertnih sistema**

Može se zaključiti, imajući na umu arhitekturu ekspertnih sistema, da je njihova realizacija zahtijevna i skupa. Moguće je postaviti pitanje: zašto razvijati ekspertne sisteme kad postoje ljudi eksperti? Odgovor na to pitanje se može dobiti kroz definisanje prednosti u primjeni ekspertnih sistema [1]. Prednosti u primjeni ekspertnih sistema su:

- *Povećana dostupnost.* Ekspertiza je dostupna na bilo kojem pogodnom računaru/hardveru. Može se reći da ekspertni sistemi predstavljaju mas produkciju ekspertize.
- *Smanjena cijena.* Ekonomski gledano cijena ekspertnog sistema je niska u poređenju sa izdacima za rad eksperta. Tamo gdje je potrebno mnogo stručnjaka jeftinije je primijeniti ekspertni sistem. Ujedno je lakše obezbijediti više računara nego obučiti stručnjake za neku oblast. Dakle, cijena obezbijedivanja ekspertize po korisniku je u velikoj mjeri smanjena.
- *Smanjena opasnost.* Ekspertni sistemi se mogu koristiti u okruženjima koja mogu biti opasna po čovjeka (nuklearne elektrane i sl.).
- *Postojanost.* Za razliku od ljudi eksperata, koji se mogu penzionisati, dati otkaz ili umrijeti, znanje ekspertnog sistema, teorijski gledano, može trajati beskonačno dugo. Osim toga, sposobnosti eksperta vremenom opadaju jer čovek stari i nije uvek i permanentno u situaciji da drži korak sa tehnološkim promjenama, dok su mogućnosti ekspertnog sistema nezavisne od vremena, zdravlja ili emocija.
- *Višestruka ekspertiza.* Korišćenjem ekspertnih sistema za rješavanje problema, znanje više eksperata (ugrađeno u bazu znanja) može biti korišćeno istovremeno i neprekidno, i u bilo koje doba dana ili noći. Nivo ekspertize dobijene kombinovanjem znanja više eksperata može prevazići nivo ekspertize jednog eksperta. Osim toga, ekspertni sistem uvijek postavlja predviđena pitanja, koja bi možda ekspert zaboravio da pita.
- *Povećana pouzdanost.* Pri donošenju odluke, ekspert može da bude pod uticajem okoline i da bude subjektivan. Menadžer, na primjer, mora kvalitetno i efikasno da donese odluku, u ograničenom vremenu koje mu je dato za to ili u stresnim situacijama kojima je izložen. Ekspertni sistem za iste ulazne parametre uvek donosi iste odluke. Na njega okolina ne utiče, pa zato kažemo da je on konzistentan.
- *Obezbijedivanje objašnjenja.* Ekspertni sistem može eksplicitno i do detalja objasniti rezonovanje koje ga je dovelo do određenog zaključka. Ovo povećava pouzdanost da je donijeta odluka korektna. Čovjek ekspert može biti previše umoran, nedostupan ili nevoljan da ovo čini sve vrijeme. Drugi razlog za postojanje mogućnosti obezbijedivanja objašnjenja rezonovanja se javlja u fazi razvoja ekspertnog sistema. Naime, sistem za objašnjenje potvrđuje da je

akvizicija znanja učinjena na ispravan način i da je znanje ispravno unijeto u sistem. Znanje može biti nepravilno unijeto u sistem, bilo greškama prilikom ukucavanja teksta, ili usljed nesporazuma između inženjera znanja i eksperta. Dodatan izvor grešaka mogu biti nepredviđene interakcije unutar ekspertnog sistema. Sve se ovo može otkriti kroz probne primjere kada se pokušava riješiti problem, čije je rješenje i način rezonovanja koji je do njega doveo, unaprijed poznat.

- *Brz odgovor.* U nekim primjenama može biti neophodan veoma brz odgovor, ili odgovor dobijen u realnom vremenu. Zavisno od hardvera i softvera koji koristi, ekspertni sistem može odgovoriti brže i pouzdanije od ljudskog eksperta. Ovo je naročito bitno u nekim vanrednim situacijama.
- *Prenosivost znanja – inteligentni tutor.* Ekspertni sistem se projektuje tako da može da prenosi znanja, dok to za jednog eksperta može da bude težak, ponekad nerješiv problem. Ekspert je stručnjak za rješavanje problema, a ne za objašnjavanje svojih odluka i prenošenje iskustva. On za to najčešće nema vremena, sve i da želi to učiniti.

Projektovanje ekspertnih sistema ima i jednu indirektnu prednost. Naime, da bi se unijelo u računar i koristilo u ekspertnom sistemu, znanje eksperata mora biti predstavljeno u eksplicitnoj formi. S obzirom da je na ovaj način znanje eksplicitno poznato, umjesto da bude podrazumijevano u umu eksperta, može biti provjerena njegova tačnost, konzistentnost i kompletnost. Nakon ovih provjera može se pojaviti potreba za izmjenama znanja ili ponovnim ispitivanjima, što unaprijeđuje kvalitet znanja.

### Prednosti čovjeka eksperta<sup>3</sup>

Ekspertni sistem ipak ne može u potpunosti zamijeniti čovjeka eksperta. Postoje situacije u kojima, bez obzira na nivo razvoja ekspertnih sistema, ekspert daje bolje rezultate.

### Kreativnost čoveka i sposobnost adaptiranja znanja

Za nove probleme ekspert nalazi kreativna rješenja i u stanju je da se adaptira na promjene i novonastale situacije, dok ekspertni sistem može da rješava samo probleme iz uskog područja za koje posjeduje ugrađeno znanje. Tipičan ekspertni sistem ne može koristiti analogiju da zaključuje o novonastalim situacijama na način kako to čovjek čini. Čovjek raspolaže tehnikama pamćenja koje mu omogućavaju da pojedine informacije svrstava u grupe, pa čitavu grupu pamti kao jedan element. Sposobnost asocijacije omogućava mu povezivanje odvojenih predstava tako da jedna izaziva drugu. Čovjekov nervni sistem je adaptivan i samoučeći, što mu omogućava uspješno snalaženje u novim i nepoznatim situacijama. Istovremeno, za novu vrstu problema ekspertni sistem je neefikasan, a često i neupotrijebljiv. Da bi mogao da rješava nove situacije prvo mora da mu se proširi baza znanja odgovarajućim pravilima.

### Senzorske sposobnosti čoveka

---

<sup>3</sup> Veći dio teksta je preuzet iz [4]

Velika prednost čovjeka je što informacije prima svim svojim čulima: vidom, sluhom, dodirrom i mirisom, a ekspertni sistem manipuliše isključivo simbolima u vidu slova i brojeva. Percepcija je cjelovit čulni doživljaj objektivne stvarnosti. Na osnovu iskustva, čovjek je u stanju da iz ukupno percipiranih informacija izdvaja bitne od nebitnih i iz haotične mase utisaka organizuje cjelinu opažaja. Čovjek na čulni doživljaj stvarnosti odmah reaguje, pošto su centri opažanja povezani sa centrima za motoriku. Percepciju je nemoguće prenijeti na računar (računari imaju samo ograničenu mogućnost unosa slike i zvuka). Sve one informacije koje čovjek percipira čulima moraju biti transformisane u simbole. Ovom transformacijom gubi se znatna količina informacija.

Ekspertni sistem ne poseduje širinu sagledavanja problema. Fokusira se samo na glavni problem. Uzimanje u obzir djelova koji se odnose na glavni problem, ali su odvojeni od njega, zahtjeva obradu velikog broja drugih, sporednih informacija.

### Zdrav razum

Racionalan čovjek se koristi zdravim razumom pri odlučivanju i u tome je racionalno ekspeditivan. Na primer, ako se pojavi nekakav nelogičan podatak, on će ga odmah uočiti i bez oklijevanja eliminisati iz daljeg razmatranja, dok ekspertni sistem mora da pretraži sve svoje podatke, da izgubi dosta vremena, da bi na kraju utvrdio nekonzistentnost podataka. Ako nekog čovjeka pitate za adresu ili telefon Petra II Petrovića Njogoša, on će vam se odmah nasmejati u lice. Ako isto pitanje postavite računaru, on prvo mora da pretraži svoju cjelokupnu bazu podataka o licima i adresama da bi odgovorio da tako nešto ne postoji.

Postojeći ekspertni sistemi imaju još uvijek značajna ograničenja i zahtijevaju intenzivan istraživački rad kako bi se izvršilo njihovo usavršavanje.

Odlike većine postojećih ekspertnih sistema mogu se svesti na sledeće:

- Ograničeni su na usko područje ekspertize. Kako se područje primjene širi, tako broj mogućnosti koje treba obrađivati eksponencijalno raste, a efikasnost sistema eksponencijalno opada.
- Jezici i sredstva za predstavljanje znanja imaju ograničene izražajne mogućnosti.
- Interfejs ka korisniku (ulaz/izlaz) većine ekspertnih sistema je krut i nije na prirodnom jeziku.
- Proces izgradnje ekspertnog sistema nije u potpunosti dostigao naučnu egzaktnost, već još uvek, u velikoj mjeri, zavisi od vještine pojedinca.
- Neophodno je postojanje stručnjaka – autoriteta u oblasti, radi održavanja konzistentnosti elemenata u bazi znanja koji se preklapaju.
- Opasnost od jednostranog viđenja predmetne problematike, s obzirom na uobičajeno oslanjanje na samo jednog stručnjaka iz oblasti.
- Nepostojanost ponašanja, koja se ogleda u tome da i najbolji ekspertni sistemi mogu dati pogrešne odgovore.
- Otežano rukovanje većinom ekspertnih sistema.
- Zbog svojih ograničenja, ekspertni sistemi se prvenstveno koriste kao savjetnici i asistenti u rješavanju problema. U budućnosti se očekuje veća postojanost ponašanja i veća autonomnost sistema, tako da se prevaziđe uloga inteligentne pomoći.

## **2.5. Opravdanost razvoja ekspertnih sistema<sup>4</sup> - primjena i domeni**

Osnovna pitanja koja se postavljaju pred svakog budućeg korisnika ekspertnih sistema su:

- Da li je ekspertni sistem pogodan za rješavanje postavljenih zadataka?
- Koji su elementi odgovarajućih djelatnosti najpristupačniji za izgradnju ekspertnih sistema?

Jedan od najvažnijih zahtjeva je da u izgradnji ekspertnih sistema učestvuju vrhunski specijalisti iz svoje oblasti. To su ljudi koji imaju veliko iskustvo profesionalnog rada u analiziranoj predmetnoj oblasti. Bez takvih saradnika, rad na izgradnji ekspertnih sistema može biti uzaludan. Dakle, neophodno je imati makar jednog eksperta koji je voljan da učestvuje u izgradnji ekspertnog sistema. Naime, izrada ekspertnog sistema podrazumijeva ispitivanje znanje eksperta koji učestvuje i potragu za mogućim greškama, stoga nisu svi eksperti voljni da učestvuju u ovom procesu. Čak i u situacijama kada postoji veći broj eksperata koji su voljni da učestvuju u izgradnji ekspertnog sistema, mudro je ograničiti njihov broj. Različiti eksperti mogu imati različite pristupe rješavanju jednog istog problema, a nekada čak mogu doći i do različitih zaključaka. Pokušaj da se iskodira više načina rješavanja jednog problema može dovesti do unutrašnjih konflikata i nekompatibilnosti u ekspertnom sistemu. Dakle, sledeći bitan element je da se ocjene pojedinih eksperata u osnovi podudaraju. Eksperti moraju znati da jasno objasne metode koje koriste pri rješavanju zadataka definisane predmetne oblasti. Ako su odgovori nejasni, specijalisti za izgradnju ekspertnih sistema neće uspeti da "preuzmu" znanja i ugrade ih u odgovarajuće programe.

Sledeća karakteristika je da postavljeni zadatak ne smije da bude isuviše težak. Ako ekspert ne može izvršiti obuku početnika odgovarajuće kvalifikacije (ako mu je zadatak nerazumljiv), ili ako su ekspertu potrebni dani i nedelje, a ne sati za rješavanje problema, to je verovatno isuviše složeno i za inženjera znanja koji treba da izvrši projektovanje ekspertnog sistema. Međutim, ako je moguće izvršiti dekomponovanje zadatka na podzadatke, onda za svaki podzadatak može biti izgrađen odgovarajući ekspertni sistem. Izgradnja ekspertnog sistema je opravdana i u onim slučajevima ako nema na tržištu dovoljno eksperata a njihove usluge su skupe.

Ekspertni sistemi veliku opravdanost imaju i u onim zadacima gde je potrebno na više mesta imati eksperte, kao na primer na platformama za istraživanje nafte, gdje na svakoj platformi mora biti makar jedan specijalista. Na kraju, razrada ekspertnih sistema je opravdana i u slučajevima nepogodnim za čoveka, kao što su atomski reaktori, kosmičke stanice ili posjeta drugim planetama.

Ključni faktori u opredjeljenju smisla izgradnje ekspertnih sistema su karakter, složenost i širina postavke zadatka koji je potrebno riješiti. Karakter ekspertnih sistema mora biti takav da se rješavanje zadatka izvodi manipulacijom simbolima i simboličkim strukturama, što je i osnovna razlika od klasičnog programiranja.

Takođe je većina zadataka za koje je opravdana izgradnja ekspertnih sistema heuristička po svojoj prirodi. Konvencionalni kompjuterski problemi se mogu koristiti za rješavanje problema koji imaju algoritamsko rješenje. Zadaci koji se mogu riješiti korišćenjem algoritama, tj. formalnih procedura, garantujući dobijanje tačnog rješenja

---

<sup>4</sup> Veći dio teksta je preuzet iz [4]

svaki put kada se primijenjuju, nisu pogodni za izgradnju ekspertnih sistema. Izgradnja ekspertnih sistema ima smisla samo onda ako su se sva ostala sredstva pokazala neprimjenljivim, odnosno, ekspertni sistemi su najpogodniji za situacije za koje nema efikasnog algoritamskog rješenja. Za ovakve situacije kažemo da su **loše strukturirani problemi**.

S druge strane, ekspertni sistemi ne treba ni da budu suviše laki. To mora biti ozbiljan zadatak, gde je potrebno da čovek potroši godine učenja i praktičnog rada da bi postao ekspert u konkretnoj predmetnoj oblasti. Dakle, za izgradnju ekspertnih sistema mora se postaviti zadatak dovoljne širine. On mora biti toliko uzak da bi se napravio, a i dovoljno širok da bi predstavljao praktični interes. Nažalost, širina je ograničena odgovarajućom predmetnom oblašću.

Najveće kočnice u razvoju ekspertnih sistema su prvenstveno ljudske prirode, jer ljudski eksperti imaju utisak da im se želi oduzeti njihovo znanje. Jedan svjetski priznati stručnjak za oboljenja visokog arterijskog pritiska jedino može povećati znanje ekspertnih sistema jer raspolaže brojnim dosijeima rijetkih slučajeva.

U poslednje vrijeme čine se pokušaji da se ekspertni sistemi međusobno nadgrađuju ako imaju isti jezik definisanja baze znanja. Poznat je jedan ekspertni sistem sposoban da istovremeno obrađuje probleme iz područja geologije, kombinatorne analize i bridža.

Perspektiva razvoja ekspertnog sistema je da bude veći "ekspert" u nekom određenom trenutku od bilo kog ljudskog eksperta. Ovo se pokušava postići tako što baza znanja treba da posjeduje znanja većeg broja stručnjaka, čija se iskustva i znanja sistematizuju u zajedničku osnovu. No, kao što je već naglašeno, sa druge strane prijete opasnost i od nekih kontradikcija i nekoherentnosti u bazi znanja.

Još uvek ne postoji sistem za sticanje znanja razvijen za kompjuter ni približno sličan ljudskom načinu razmišljanja. Još uvijek su ekspertni sistemi loši u npr. čitanju knjiga, prisustvovanju sastancima, diskusiji sa svojim "kolegama" itd. No, sa druge strane, ekspertni sistem često može pružiti bolju informaciju od čovjeka eksperta jer se ne zamara, ne stari, i nepogrešiv je na svom maksimalnom nivou kompetencije.

U medicini i geologiji ekspertni sistemi već sad imaju u većini slučajeva performanse koje se mogu ravnopravno upoređivati sa najboljim svjetskim stručnjacima. Inače, u bolnici u Stanford-u ljekari direktno potpisuju ljekarske nalaze koje izrađuje ekspertni sistem.

Prilikom izrade ekspertnih sistema potrebo je čuvati se „kombinacionih eksplozija“, jer se može dogoditi da za bazu znanja na nivou od 500 pravila i 500 činjenica vrijeme izračunavanja bude neprihvatljivo. S druge strane, ponekad je složeno upravljati i mehanizmom za zaključivanje, jer on mora da omogući ekspertu da svoja znanja definiše u obliku pravila na „vlastitom“ jeziku i na taj način učini ga nezavisnim od informatičara.

Kada je razvoj ekspertnih sistema moguć i opravdan [4]

Evidentno je da su ekspertni sistemi još uvek nova oblast u primjeni računara, pa malo zbog neznanja, a malo zbog pretjerivanja, postoji tendencija da se ekspertnim sistemom naziva i ono što nije ekspertni sistem. Isto tako, ekspertnim sistemima se pokušavaju riješiti problemi koji nisu podesni za rješavanje pomoću njih. Međutim, ekspertni sistemi nisu svemoguć i postoje već rečena jasna ograničenja njihove

primjene. Ne može se ekspertni sistem projektovati za svaku primjenu, ni za svaki obim posla. Kada se onda to čini? To se čini ako je razvoj moguć, opravdan i odgovarajući.

Da bi razvoj ekspertnog sistema bio moguć, potrebni su sledeći uslovi:

- rješavanje problema ne zahtijeva rasuđivanje zdravog razuma,
- zahtijeva misaone vještine,
- eksperti mogu da definišu metode rješavanja problema i oni su jedinstveni,
- problem nije suviše složen,
- problem je razumljiv.

Razvoj ekspertnog sistema je opravdan ako:

- postoji ekonomska isplativost, ekspertni sistem se razvija za praktičnu namjenu,
- eksperti često napuštaju radno mesto,
- eksperti su rijetki,
- potrebna je ekspertiza na mnogo mjesta,
- radno mesto je nepodesno ili čak štetno za čoveka, zbog kontaminiranosti ili drugog razloga koji ugrožava život, pa čovjeka zamjenjuje računar.

Razvoj ekspertnog sistema je odgovarajući u slučaju da:

- rješavanje problema zahtijeva rukovanje simbolima,
- koriste se heuristička rješenja,
- problem nije suviše jednostavan,
- rješenje ima praktičnu primjenu,
- rješavanje problema nije suviše obimno.

## **2.6 Odnos sistema za podršku odlučivanju i ekspertnih sistema**

U savremenom okruženju podrazumijeva se da je svaka ozbiljna, profesionalna djelatnost podržana osmišljenim, najčešće računarom podržanim informacionim sistemom. Dakle, informacioni sistem koji vrši akviziciju, bilježi transakcije i pruža izvještaje o stanju realnog sistema smatra se realnošću koja pruža potrebne, ali ne i dovoljne uslove za odlučivanje. Iz tih razloga je zapravo i došlo do razvoja disciplina o kojima je reč – sistema za podršku odlučivanju i ekspertnih sistema.

Sistem za podršku odlučivanju pretpostavlja postojanje neke vrste baznog informacionog sistema nad koji se nadgrađuje. S obzirom na kompleksnost problema koje obrađuje, to obično znači računarski i interaktivni sistem rada, mada u principu (ali samo teoretski) računar nije neophodan uslov funkcionisanju sistema za podršku odlučivanju.

Osnovni cilj sistema za podršku odlučivanju i ekspertnog sistema je u osnovi isti, utoliko što je namjena oba sistema povećanje kvaliteta odlučivanja. Ipak, filozofija koja leži u osnovi njihove izgradnje je dosta različita. Cilj sistema za podršku odlučivanju je da podrži korisnika kod donošenja (slabo struktuiranih) odluka, obezbjeđujući mu brz i jednostavan pristup do podataka, modela i znanja. Na



drugoj strani, cilj ekspertnih sistema je da obezbijedi korisniku zaključak ili odluku koja je tačna u svako doba (ili bar tačnija od bilo koje koju bi korisnik mogao da donese bez ekspertnog sistema). Dakle, sistem za podršku odlučivanju pomaže pri odlučivanju, dok ekspertni sistem "odlučuje". Osim toga, ekspertni sistem se ne može koristiti kod slabo strukturiranih procesa odlučivanja.

Poređenjem operativnih razlika, zaključujemo da sistem za podršku odlučivanju dozvoljava korisniku suočavanje sa problemom na ličan, fleksibilan način, obezbijedujući mu mogućnost manipulacije podacima i kontrolu njihove upotrebe u toku procesa odlučivanja. Ekspertni sistem korisniku ostavlja malo, ili nimalo, fleksibilnosti pri analizi problema. Umjesto toga, izvodi se odgovarajući segment znanja na način koji je određen mehanizmom za zaključivanje.

S obzirom da ekspertni sistem djeluje kao nezavisan konsultantski sistem, a sistem za podršku odlučivanju kao mehanizam za podršku odlučivanju, njihovi koncepti ni u kom slučaju nisu oprečni; sve više se radi na njihovom približavanju – do integrisanja, po nekoliko osnova: ekspertni sistem može biti generator alternativa – samostalno ili kao dio sistema za podršku odlučivanju, ali i obratno, memorisano znanje i odgovarajuća pravila sistema za podršku odlučivanju mogu postati dio ekspertnog sistema. Hoće li ekspertni sistem dominantno postati dio sistema za podršku odlučivanju, ili obrnuto, nije u krajnjoj liniji od presudnog značaja.

## Rješavanje problema pretraživanjem

Klasično pretraživanje koje smo koristili u kursevima programiranja sastoji se u nalaženju nekog podatka u skupu podataka organizovanih na određeni (poznati) način. Pretraživanje se vršilo na osnovu određene karakteristike podatka koji se traži. U ekspertnim sistemima pretraživanje ima drugačiji smisao. Podsjetimo se, zadatak ekspertnog sistema je rješavanje problema koji se inače teško može riješiti metodama klasičnog programiranja. Kada su u pitanju problemi koje je potrebno rješavati inteligentnim agentima, najčešće je prvo potrebno odgovoriti na pitanja: šta tražiti i gdje tražiti. Dok se za prvo pitanje može reći da je *ključ pretraživanja*, drugo se odnosi na tzv. *prostor pretraživanja*. Kada je u pitanju vještačka inteligencija, prostor pretraživanja se odnosi na *skup mogućih stanja*, pa se češće naziva *prostorom stanja*. Stanje se može posmatrati kao snimak okruženja u jednom trenutku vremena. Za razliku od uobičajenog prostora pretraživanja, prostor stanja u problemima VI nije uvijek poznat u cjelosti prije rješavanja problema.

Proces rješavanja problema pretraživanjem se može organizovati u dvije faze. Prva faza je predstavljanje problema korišćenjem odgovarajućeg dobro organizovanog prostora stanja, dok je druga faza testiranje postojanja dobro definisanog cilja u tom prostoru stanja. Rješavanje problema pretraživanjem u ekspertnim sistemima se odnosi na identifikaciju rješenja nekog problema (cilja) i određivanje optimalne putanje koja vodi do tog rješenja kroz jedan ili više mogućih prelaza od datog startnog stanja. Može se reći da agent (softver) za rješavanje problema odlučuje šta da radi pronalaženjem sekvence akcija koje će ga odvesti do željenog stanja (rješenja). Od inteligentnih agenata se očekuje da maksimizuju mjeru svog učinka. Ponekada je rješavanje problema pojednostavljeno ukoliko agent može da prilagodi svoj cilj kako bi ga lakše ostvario.

Procedura rješavanja problema pretraživanjem se najbolje može shvatiti kroz analizu konkretnog problema. Zamislimo da se agent nalazi na ljetovanju u Budvi.

Mjera učinka agenta bi mogla sadržati više faktora: obići Kotor, izbjeći sunčanicu, uživati u noćnom životu i slično. Način na koji bi agent mogao maksimizovati mjeru svog učinka u ovom slučaju uključuje kompleksne procese odlučivanja, sa mnogo kompromisa i detaljnom analizom turističkih vodiča i informatora. Međutim, ukoliko je poslednji dan odmora i agent treba da uhvati let iz Podgorice, ili mu propada karta i mora dodatno platiti hotel u Budvi, njegov cilj se svodi na dolazak u Podgoricu. Sada ima smisla odbaciti sve one akcije koje ne bi za rezultat davale dolazak u Podgoricu na vrijeme da se uhvati željeni let. Na ovaj način je i sam proces odlučivanja agenta u velikoj mjeri pojednostavljen. Vidimo da jasno definisan cilj utiče na bolju organizaciju ponašanja agenta. Ukoliko je cilj jasno definisan smanjuje se broj zadataka koje agent pokušava da odradi kako bi postigao svoj cilj. *Definisanje cilja* na osnovu trenutnog stanja i mjere agentovog učinka je prvi korak u rješavanju problema.

Kada je u pitanju rješavanje problema pretraživanjem, cilj se može definisati kao skup stanja u kojima je cilj zadovoljen. Zadatak agenta se sada svodi na pronalaženje sekvence akcija koja će ga dovesti do ciljnog stanja. Prije nego što može identifikovati sekvencu akcija koje će ga dovesti do ciljnog stanja, agent mora definisati koju vrstu akcija i stanja da posmatra. Kada je u pitanju ostvarivanje cilja – dolazak u Podgoricu na vrijeme da se uhvati željeni let, prilikom rješavanja problema, nećemo se baviti detaljima kao što je način vožnje, gužva na putu, broj putnika u automobilu, već ćemo analizirati samo akcije koje čini dolazak iz jednog u drugi grad. Ovo je zapravo *formulacija problema* – definisanje akcija i stanja koja će se uzimati u obzir prilikom rješavanja problema pretraživanjem. U konkretnom slučaju akcije su vožnja iz jednog u drugi grad, dok su stanja – biti u određenom gradu.

Od jednog grada do drugog može postojati više puteva različitih dužina. Postavlja se pitanje kako da agent odluči kojim putem da ide. Ukoliko krene iz Budve ima mogućnost da ide u najmanje tri grada, na tri različite strane: Kotor, Bar, Cetinje. Nijedan od ovih gradova nije ciljna destinacija agenta. Ukoliko agent nema nikakvu informaciju o stanju svijeta, odnosno nije upoznat sa putnom infrastrukturom Crne Gore, neće znati kuda da ide iz Budve. Naime, agent ne zna koju od mogućih akcija da odabere jer ne zna mnogo o stanjima (gradovima) u koja će ga odabrana akcija prevesti. Najbolje što može da pokuša je da slučajno odabere jedan od tri moguća pravca.

Ukoliko bi se agentu dostavio opis ove putne mreže (ili, ukoliko je agent zna), on može utvrditi dodatne informacije o stanjima (gradovima) u kojima bi se mogao naći i o akcijama koje bi mogao sprovesti iz pojedinih stanja. U ovom slučaju, zadatak agenta je da iskoristi ove informacije kako bi utvrdio sekvencu stanja (gradova) i akcija koje bi uslijedile nakon odabira svakog od narednih stanja (tri susjedna grada), kako bi se na kraju utvrdilo koja od tih sekvenci vodi do željenog stanja (Podgorice). Jednom kada je utvrdio koja sekvenca ga vodi do cilja, agent sprovodi akcije kako bi prešao iz jednog u drugo stanje utvrđene sekvence (vozio se iz jednog u drugi grad). Generalno govoreći, agent pred kojim se nalazi više istovremenih opcija sa nepoznatim vrijednostima može odlučiti šta da radi tako što će prvo ispitati različite moguće sekvence akcija koje ga vode do stanja sa poznatom vrijednošću i nakon toga odabrati najbolju sekvencu. Proces ispitivanja različitih mogućih sekvenci akcija i stanja u koje one vode je zapravo pretraživanje. Nakon što se pretraživanjem utvrdi sekvence akcija pristupa se realizaciji pojedinih akcija – faza izvršavanja.

Sada se rješenje postavljenog zadatka svodi na formulisanje cilja i problema koji se treba riješiti, pozivanje procedure pretraživanja koja će utvrditi sekvencu akcija koja prevodi iz početnog stanja, preko niza mogućih stanja, do ciljnog stanja, i izvršavanje tih akcija. Izvršavanje akcija se najčešće svodi na izvršavanje prve akcije u sekvenci utvrđenih akcija, uklanjanje izvršene akcije i ponavljanje ove procedure sve dok se ne dođe do cilja.

Pseudokod za ovakav način rješavanja problema bi bio:

```
function akcija=jednostavno_rjesenje_problema(opazanje)
    seq, sekveca akcija, inicijalno prazna
    stanje, neki opis trenutnog stanja svijeta
    cilj, cilj koji u početku nije definisan
    problem, formulacija problema
    stanje = obnoviStanje(stanje, opazanje)
    if seq==prazna
        cilj = ustanoviCilj(stanje)
        problem = formulišiProblem(stanje, cilj)
        seq=Pretražuj(problem)
    akcija=Prva(seq)
    seq=preostale_akcije(seq)
return akcija
```

Prikazan je jednostavan način za rješavanje problema koji se svodi na definisanje cilja i problema, pretraživanje za sekvencom akcija koje bi riješile postavljeni problem, i sekvencijalno izvršavanje akcija.

Prilikom rješavanja problema gore opisanim postupkom, nakon utvrđivanja koja se akcija treba izvršiti, ta akcija se sprovodi ne obazirući se na stanje u svijetu (put zatvoren zbog radova), računajući da će utvrđena akcija uvijek dovesti do dobrog rješenja. Ovakav pristup se može koristiti u slučajevima kada je *stanje svijeta statičko* – ne mijenja se ukoliko agent ne djeluje na njega. Takođe se implicitno smatra da je *stanje svijeta diskretno*, odnosno da postoji ograničeni broj potpuno definisanih stanja i akcija. Smatra se da je početno stanje poznato i da je *stanje svijeta determinističko* (nasuprot stohastičkog) – sledeće stanje svijeta je kompletno definisano pomoću trenutnog stanja i akcije koju preduzima agent (ako je okruženje determinističko osim u slučaju akcije drugih agenata, tada je okruženje *strategijsko*).

### **3.1. Definisanje problema i rješenja**

Dobro definisan problem i opis stanja svijeta će u velikoj mjeri olakšati nalaženje rješenja (postizanje cilja). Rekli smo da kada se zna šta je cilj, može se i prostor mogućih stanja suziti i olakšati pretraga, odnosno smanjiti zahtjevi koji se postavljaju pred agenta. Prilikom definisanja prostora stanja treba odbaciti sve nevažne detalje iz stvarnog svijeta, koji se želi modelirati, vodeći računa da su prostorom stanja opisane sve pojedinosti bitne za rješavanje analiziranog problema. Prostor stanja mora obuhvatiti sva moguća stanja prilikom rješavanja problema. Najbolje bi bilo, ukoliko je moguće, prostor stanja definisati bez direktnog nabiranja pojedinih stanja (kojih često ima beskonačno mnogo). U primjeru u kojem nam je cilj stizanje na let, nijesmo uzeli u obzir stanje na putu, gužvu, količinu goriva u automobilu, vremenske uslove i slično, jer bi to dodatno proširilo prostor stanja i zahtijevalo dodatne odluke od agenta. Proces uklanjanja detalja iz reprezentacije svijeta naziva se *apstrakcija*.

Prostor stanja se sastoji od: skupa početnih stanja, skupa dozvoljenih stanja, skupa prelaza (operatora) i skupa završnih stanja. Struktura prostora stanja odgovara strukturi problema koji se rješava i definiše se imajući na umu način rješavanja problema, prevođenje iz početnog u željeno stanje upotrebom dozvoljenih prelaza.

*Predstavljanje problema u prostoru stanja zasniva se na definisanju:*

- **Početnog stanja** (stanje od kojeg se započinje pretraživanje). Početnih stanja može biti više. U primjeru u kojem je potrebno stići na avion koji polijeće iz Podgorice, a agent se nalazi u Budvi, imamo samo jedno početno stanje  $U(Budvi)$ .
- **Skupa dozvoljenih akcija, tj. operatora.** Akcije-operatori su prelazi iz određenog stanja u drugo stanje iz skupa mogućih stanja. Ukoliko se agent nalazi u nekom stanju  $x_i$  primjenom operatora  $opi$  se može prevesti u stanje  $x_j$ , što se može predstaviti kao

$$opi: x_i \rightarrow x_j$$

Ovdje je bitno primijetiti da se ne može uvijek svaki operator primijeniti na svako stanje. Uz svaki operator se najčešće vezuju određeni uslovi koje mora zadovoljavati stanje u kojem se nalazi agent, da bi se taj operator mogao primijeniti. Operatori se mogu opisivati na različite načine u zavisnosti od konkretnog problema. Jedan od mogućih načina opisivanja operatora je preko funkcije sljedbenika  $opi: (trenutno\_stanje, sljedeće\_stanje), uslovi\_primjene(niz\_uslova)$ .

U našem primjeru bi imali tri operatora koja se mogu primijeniti na početno stanje, i to  $\{(Budva, Petrovac), (Budva, Kotor), (Budva, Cetinje)\}$ . Ne bi se mogao primijeniti operator  $(Budva, Herceg Novi)$ , nijesu susjedni gradovi – Što je i uslov za primjenu operatora.

Početno stanje uz pomoć funkcija sljedbenika za početno i sva iz njega dostupna stanja definiše prostor stanja za razmatrani problem – skup svih stanja do kojih se može doći iz početnog stanja. S obzirom na način definisanja prostora stanja, kao logičan izbor za prikaz prostora stanja se odabiraju grafovi u kojima su čvorovi stanja, a lukovi (grane) koji povezuju pojedine čvorove su operatori. Graf se formira tako što se pođe od početnog stanja, a zatim se korišćenjem nekog od dostupnih operatora generiše novo stanje (čvor) sljedbenik stanja (čvora) iz kojeg je generisan. Polazni čvor (roditelj) se spaja sa generisanim čvorom granom koja se označava imenom operatora koji je primijenjen za njegovo generisanje. Proces generisanja čvorova se nastavlja sve dok se ne dostigne ciljno stanje. Putanja u ovako generisanom grafu je sekvenca stanja koja su povezana sekvencom operatora. Putanja koja povezuje početno sa ciljnim stanjem čini rješenje problema. Politiku generisanja čvorova određuje **strategija pretraživanja**.

- **Testa na ciljno stanje** kojim se provjerava da li je trenutno stanje ciljno stanje. U našem primjeru ciljno stanje je jasno definisano – biti u Podgorici. U ovakvim slučajevima, kada imamo jasno definisano jedno ili više mogućih ciljnih stanja, test na ciljno stanje je jednostavan – provjerava se da li smo u nekom od mogućih ciljnih stanja. Međutim, cilj nije uvijek definisan kao skup stanja, već se može određivati nekim svojstvima koja moraju biti zadovoljena kada se dođe do cilja (na primjer pozicija na šahovskoj tabli takva da napadamo protivničkog kralja, a da on nema načina da se odbrani).

- **Funkcije cijene puta** koja svakoj sekvenci stanja (putanji) pridružuje neku numeričku vrijednost, kako bi se iz skupa pronađenih rješenja moglo odrediti najbolje. Dakle, sekvenca čvorova koja vodi od početnog do ciljnog stanja će predstavljati rješenje, dok će se kvalitet rješenja ocjenjivati funkcijom cijene puta, a rješenje sa najmanjom cijenom će se smatrati *optimalnim rješenjem*. Numerička vrijednost pridružena nekoj putanji treba da reflektuje kvalitet te putanje. U našem primjeru, kvalitetnija bi bila kraća (brža) putanja, jer želimo stići na vrijeme na let, pa će funkcija cijene puta biti zbir rastojanja između pojedinih gradova. Prilikom analize raznih strategija pretraživanja razmatraćemo slučajeve u kojima se cijena putanje dobija kao suma cijena pojedinih prelaza u sekvenci čvorova (stanja) koje ta putanja sadrži.

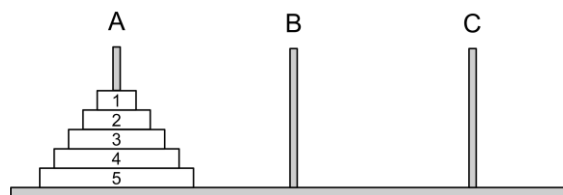
### 3.2. Primjeri problema koji se mogu riješiti pretraživanjem

Bez obzira na „jednostavnost“ pristupa rješavanja problema pretraživanjem, on se može primijeniti na veliki broj problema. U ranim fazama razvoja VI smatralo se da se bilo koji problem može riješiti pretraživanjem, međutim razvoj teorije složenosti pokazao je jasna ograničenja ovog pristupa. Ovdje će biti pobrojani neki od često korišćenih problema za poređenje pojedinih metoda pretraživanja. Veći broj primjera se može naći u [3] i [6].

Uobičajeno se problemi koji se mogu riješiti pretraživanjem dijele na idealne slučajeve (toy) koji su jasno definisani i egzaktni i realne slučajeve. Idealni slučajevi se koriste za ilustraciju i testiranje algoritama za rješavanje problema, kako bi se kasnije primijenili za rješavanje realnih problema (što nam je krajnji cilj).

Često korišćeni idealni problemi su: Hanojske kule, slagalica, 8 dama.

**Hanojske kule** – posmatra se  $n$  različitih diskova i tri stuba. U početku su svi diskovi na prvom stubu, pri čemu su poređani tako da se na dnu nalazi najveći, a na vrhu najmanji disk. Cilj igre je da se svi diskovi prebace sa prvog na treći stub, koristeći drugi kao pomoćni, pri čemu je u jednom trenutku moguće prebaciti samo jedan disk, i ni u jednom trenutku se ne smije naći veći disk na manjem disku.



Slika 3 Hanojske kule - početno stanje za  $n=5$

*Početno stanje* za slučaj sa tri diska bi bilo (A,A,A) – sva tri diska su na istom stubu, dok bi *ciljno stanje* bilo (C,C,C), prvi član uređene trojke se odnosi na poziciju najvaćeg diska, drugi na poziciju srednjeg diska, a treći na poziciju najmanjeg. Kada stanje definišemo sa (A,A,A), podrazumijevamo da se manji disk nalazi na većem, o čemu ćemo morati voditi računa prilikom definisanja operatora i funkcija prelaza.

Prikom definisanja *skupa dozvoljenih prelaza – operatora*, mora se voditi računa o tome da se u jednom trenutku može pomjeriti samo jedan disk, i da se veći disk ne smije nalaziti na manjem. Mogući operatori (prelazi) su: prebacivanje najmanjeg diska disk1(disk2 ili disk3) na štap na kojem se trenutno ne nalazi, uz ograničenje da na tom štapu ili ne smije biti ništa, ili se nalazi jedan ili više manjih diskova i da se samo disk sa vrha može prebaciti. Dakle, ukoliko imamo stanje (A,B,A), treći,

najmanji disk se može prebaciti na štap B, ili na štap C. Drugi, srednji disk se može prebaciti na treći štap, ali ne i na prvi jer je na vrhu disk koji je manji od njega. Prvi disk se ne može pomjerati, najmanji disk je iznad njega.

Iz početnog stanja su moguća dva prelaza: najmanji disk se prebacuje na stub B i najmanji disk se prebacuje na stub C, pa su iz početnog stanja (A,A,A) mogući prelazi u stanje (A,A,C) ili (A,A,B). Možemo davati imena ovim operatorima, npr op1A za prebacivanje najmanjeg štapa na disk A, op2A, za prebacivanje najmanjeg diska na štap B, op3A, za prebacivanje najmanjeg diska na štap C, itd.

Iz stanja (A,A,C) moguć je prelaz u stanje (A,B,C) i (A,A,B), dok nije moguć prelaz u stanje (A,C,C) jer je disk koji se već nalazi na štapu C, manji od diska koji se prebacuje.

*Dozvoljena stanja* su sva stanja u kojima je manji disk na većem. S obzirom na način obilježavanja broj kombinacija je  $3^3=27$ , tri diska se raspoređuju, a mogu se rasporediti tako da budu na jednom od tri štapa. Ukoliko bi imali  $n$  diskova broj stanja bi bio  $3^n$ . Prostor stanja i dozvoljeni prelazi su prikazani tabelom Tabela 1. Ovdje je izostavljeno obilježavanje imena primijenjenih operatora, ali je jasno koji je operator primijenjen ukoliko se u porede polazno i novo stanje.

*Test na ciljno stanje* se svodi na testiranje da li se nalazimo u stanju (C,C,C).

*Funkcija cijene puta* se svodi na inkrementiranje cijene za jedan prilikom svakog prelaza iz trenutnog u naredno stanje. Putanja koja je sa najmanje prelaza dovela do ciljnog stanja ima najmanju funkciju prelaza i smatra se optimalnim rješenjem.

Dozvoljena stanja i operatori su prikazani tabelarno na sljedećoj strani. Obilježen je dio prelaza, čitalac bi trebalo da dopuni tabelu.

**Slagalica** – rješavanje slagalice se često koristi kao testni problem za nove algoritme. Na slici je prikazan primjer slagalice sa 8 polja

|   |   |   |
|---|---|---|
| 4 | 3 | 6 |
| 8 |   | 1 |
| 2 | 5 | 7 |

Početno stanje

|   |   |   |
|---|---|---|
|   | 1 | 2 |
| 3 | 4 | 5 |
| 6 | 7 | 8 |

Ciljno stanje

Početno stanje je dato na slici lijevo, ciljno stanje je prikazano desno. Operatori mogu biti pomjeranje praznog polja u četiri pravca (gore, dolje, lijevo i desno). Nije moguće izvršiti pomjeranje u četiri pravca iz svakog stanja. Kada je šupljina na ćoškovima, ima samo dvije mogućnosti za pomjeranjem, što će odgovarati faktoru grananja 2, kada bi se pravila tabela kao za prethodni primjer, ovo bi odgovaralo redovima sa samo dvije jedinice. Kao stanja bi u ovom slučaju se mogla uzeti 9-torka brojeva, pa bi početnom stanju odgovarala (4,3,6,8,0,1,2,5,7), pri čemu nula odgovara praznini, a bitan je redosljed. Cilj bi bio predstavljen kao (0,1,2,3,4,5,6,7,8). Mogući broj stanja je 9!

Veliki je broj realnih problema koji se koriste za poređenje performansi pojedinih algoritama. Problem koji smo analizirali kao ilustraciju, pronalaženje putanje za dolazak iz grada A, u grad B je jedan od realnih problema. Najčešće korišćeni je problem trgovačkog putnika, dok se često koriste i problemi dizajniranja VLSI kola i navigacija robotom.

Kada je u pitanju problem trgovačkog putnika, trgovac ima za zadatak da obiđe  $n$  gradova, pri čemu ne smije dva puta obići isti grad. Smatra se da između svakog para gradova postoji direktna veza. Dok je početno stanje i dalje definisano na sličan način kao kod problema određivanja rute, koji smo razmatrali (Budve-Podgorica), opis stanja je znatno složeniji. Sada nije dovoljno pamtiti grad u kojem se nalazimo, već i listu gradova koje smo već obišli, kako ne bi posjetili isti grad dva puta – čime se značajno povećava broj mogućih stanja. Ukoliko imamo  $N$  gradova, broj mogućih putanja je  $(N-1)!$ . Dalje, prilikom primjene operatora, prelaska iz jednog u drugi grad, potrebno je ispitati da li je grad već obiđen. Cilj je postignut kada se trgovački putnik nađe u početnom gradu, a obišao je sve ostale gradove. Optimalna putanja je najkraća putanja, takva da se njom obiđe svaki grad tačno po jedan put.

|     | 111 | 112 | 113 | 121 | 122 | 123 | 131 | 132 | 133 | 211 | 212 | 213 | 221 | 222 | 223 | 231 | 232 | 233 | 311 | 312 | 313 | 321 | 322 | 323 | 331 | 332 | 333 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 111 |     | 1   | 1   |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| 112 | 1   |     | 1   |     |     |     |     | 1   |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| 113 | 1   | 1   |     |     |     | 1   |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| 121 |     |     |     |     | 1   | 1   | 1   |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| 122 |     |     |     | 1   |     | 1   |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     | 1   |     |     |     |     |
| 123 |     | 1   | 1   |     | 1   |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| 131 |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| 132 |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| 133 |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| 211 |     |     |     |     |     |     |     |     |     |     | 1   | 1   |     |     |     |     |     |     |     | 1   |     |     |     |     |     |     |     |
| 212 |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| 213 |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| 221 |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| 222 |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| 223 |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| 231 |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| 232 |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| 233 |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| 311 |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| 312 |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| 313 |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| 321 |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| 322 |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| 323 |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| 331 |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| 332 |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| 333 |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |

**Tabela 1** Tabela sa dozvoljenim stanjima i prelazima za slučaj Hanojskih kula sa tri diska. Disk A je označen sa 1, disk B sa 2, i disk C sa 3.



## Minimum zahtijevanog znanja sa drugog termina

1. Definisati svojim riječima ekspertni sistem.
2. Koji oblici znanja postoje? Koja znanja su značajna za klasične ekspertne sisteme? Na koji način se do njih dolazi?
3. Šta je inženjerstvo znanja? Koji su zadaci inženjera znanja u procesu izgradnje ekspertnog sistema?
4. Koji su osnovni koraci u početnom razvoju ekspertnih sistema. Ko sve učestvuje u njegovom testiranju?
5. Koja su dva osnovna dijela ekspertnih sistema i od čega se sastoje?
6. Šta je baza znanja i koja joj je funkcija?
7. Šta se nalazi u radnoj memoriji u toku jedne sesije?
8. Šta je modul za zaključivanje i koja mu je funkcija?
9. Od čega se sastoji modul za zaključivanje?
10. Opišite cikluse u funkcionisanju modula za zaključivanje.
11. Šta je vezni modul ekspertnog sistema i koja mu je funkcija? Šta je modul za zahvatanje znanja?
12. Koja je funkcija modula za interpretaciju znanja? Iz kojih djelova se sastoji, i kakva je funkcija pojedinih njegovih djelova?
13. Koji su bitni zadaci zahvatanja znanja u ekspertnim sistemima?
14. Koje su prednosti ekspertnih sistema u odnosu na čovjeka?
15. Zadatak sa problemom koji je potrebno predstaviti u prostoru stanja.

## LITERATURA

- [1] Joseph C. Giarratano, Gary D. Riley.: *Expert Systems: Principles and Programming*, Prentice Hall, 2nd ed., 2002.
- [2] Polišćuk, E.J.: *Ekspertni sistemi*, Informatička literatura JEP (vlastito izdanje), Podgorica, 2004.
- [3] Russell S., Norvig P.: *Artificial Intelligence: A Modern Approach*, Prentice Hall, NJ, 1995.
- [4] Prof. dr Alempije Veljović: *Ekspertni sistemi*, CET Čitalište 72 – April 2009.
- [5] <http://ri4es.etf.rs/>, posljednji put pristupano, 08. 02. 2010. godine.
- [6] Konar A.: *Artificial Intelligence and Soft Computing: Behavioral and Cognitive Modeling of the Human Brain*, CRC Press, December 8, 1999.