# EKSPERTNI SUSTAVI-PITANJA ZA MI

## I. Uvod

1. Što su ekspertni sustavi (ES) i što im je temeljni sadržaj ?
2. Koja četiri pristupa oblikovanju ES razlikujemo i koji je u fokusu ovog predmeta?
3. Navedite i ukratko objasni zajedničke značajke i razlike između 5 vrsta racionalnih agenata.
4. Koje su polazne hipoteze u oblikovanju ES utemeljenih na obradi simbola ?
5. Navedite razine apstrakcija u računalnom sustavu počevši od najniže razine (sklopovlje). Koji je glavni nedostatak programskih jezika više razine?
6. Navedite i objasnite temeljne probleme pri izgradnji i izvođenju ekspertnih sustava

**1.**

To su programski proizvodi namijenjeni rješavanju složenih problema u uskoj domeni primjene, za što je potrebno stručno (ekspertno) znanje.

Temeljni sadržaj ekspernih sustava je predstavljanje i obrada znanja

**2.**

1) Sustav koji rasuđuje poput ljudi- modeliranje kognitivnih procesa, introspekcija, eksperimentalna psihologija

2) Sustav koji djeluje poput ljudi- Turingov test

3) Sustav koji rasuđuje racionalno- formalno, matematički zasnovano predstavljanje znanja i rasuđivanja

4) <u>Sustav koji djeluje racionalno</u> –racionalan agent- opaža okolinu i djeluje slijedeći svoje ciljeve temeljem stvarne slike svijeta

**3.**

1) jednostavni reaktivni agent- ne pamti evoluciju svijeta, preslikava trenutno stanje svijeta u moguće akcije na temelju pravila u obliku uvjet-akcija

2) reaktivni agent zaslovan na modelu- osim informacija iz senzora, pamćenje prošlih stanja, ima interni model koji apstrahira svijet oko agenta

3) agent zasnivan na modelu i cilju- agent ima ugrađeni cilj, uvođenje pretraživanja i planiranja kako bi se postigao taj cilj

4) agent zasnovan na modelu i korisnosti cilja- uvođenje mjere kvalitete rješenja, odnosno kvalitete stanja u kojem se agent našao

5) agent koji uči- na početku može biti nepoznato okruženje, agent uči i usavršava početno znanje

-learning element- odgovoran za poboljšanja

-performance element- odgovoran za izbor vanjskih akcija

-critic- daje informaciju kako agentu ide

- problem generator- sugerira akcije koje dovode do novog iskustva

**4.**

1) rasuđivanje je obrada (manipulacija) simbola

2) svaka 'obrada' simbola se može izvesti na Turingovom stroju

**5.**

1) sklopovlje

2) strojni jezik

3) simbolički strojni jezik (assemblerska razina)

4) imperativni programski jezik

5) algoritmi i strukture podataka, apstraktni tipovi

6) razina simbola (LISP)

7) razina znanja (dokazivanje teorema, Prolog, OWL...)

→glavni nedostatak programskih jezika više razine- sporost u izvođenju

**6.**

1) dobivanje znanja- eksperti imaju ograničeno vrijeme i skupi su

- eksperti ne znaju sve što znaju

- eksperti ne žele izraziti sve što znaju

2) integracija s drugim sustavima i pristup velikim bazama podataka –potrebni međujezici za komunikaciju s drugim sustavima i bazama

3) performanse- sporo rasuđivanje u logici prvog reda i neučinkoviti jezici i alati

## 2. LISP

1. Navedite razliku između imperativnih i funkcijskih programskih jezika.
2. Navedite barem 4 značajke LISP-a.
3. Objasniti sintaksu i semantiku LISP-a (petlja izvođenja).
4. Navedite rezultat izvođenja LISP koda (za neki jednostavni primjer).
5. S kojom funkcijom osiguravamo lokalno vezanje simbola?
6. Što su to lambda izrazi ?
7. Navesti funkcije i dati primjer za pridruživanje i dobavljanje vrijednosti obilježjima simbola.
8. Prikažite navedeni izraz točkastim parovima.
9. Kako se LISP-ovi s-izrazi (liste) smještaju u memoriji računala? Prikazati na nekom jednostavnom primjeru.
10. Navesti glavne značajke Clojurea, kao primjera suvremenog dijalekta LISP-a.

**1.**

Imperativni jezici (C,Fortran, Pascal, Java, ...) – intenzivno korištenje pridruživanja

Funkcijski jezici (Haskell, LISP, Scheme, ...) – gradbeni blokovi su izrazi koji se evaluiraju kroz primjenu funkcije

- u čistim funkcijskim jezicima nema pridruživanja; funkcije obuhvaćaju ulazne parametre temeljem read-only. Nema promjene ranijih vrijednosti stanja

**2.**

1) viša razina apstrakcije od ostalih imperativnih jezika – izvođenje ovisi o tipovima parametara, a program i podaci se mogu mijenjati tijekom izvođenja

2) LISP je visoko rekurzivan glede strukture podataka i algoritama

3) održava jednakost programa i podataka – temeljna struktura podataka je lista

4) tip varijable se ne mora definirati unaprijed

5) implicitan rad s pokazivačima

6) LISP je pogodan za probleme koji početno nisu potpuno razumljivi

**3.**

Sintaksu čine s- izrazi (simbolički izrazi)- atomi ili liste

Atomi- brojevi, niz znakova, simboli

Liste- svaki atom je s- izraz, s-izraz (S1 S2 ...Sn) je lista

→semantiku određuje petlja izvođenja: čitaj – evaluiraj – ispiši

**4.**

>(+ 1 2 3)

6

**5.**

setq (set equal)

**6.**

(lambda (<formalni-parametri>) <tijelo>)

Funkcija koja se prenosi kao parametar mora biti definirana, definicija fje može se prenijeti izravno lambda izrazom

**7.**

get – vadi obilježja simbola

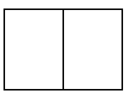
setf (setq) –pridruživanje

PR. (setf (get 'kosara' sadrzaj) '(kruh mlijeko sir))

**8.**


**9.**

LISP-ovi s-izrazi se smještaju u memoriju pomoću konstrukcijske ćelije –sadrže 2 kazala ('car' i 'cdr') liste u prikazu točkastim parovima



car cdr

**10.**

1) vrti se na Javinom virtualnom stroju, omogućuje pozivanje Javinog koda, a i Java omogućuje pozivanje koda iz Clojurea

2) podrška konkurentnom izvođenju, posebno u vidu višedretvenosti

3) strukture podataka su nepromjenjive i perzistentne

4) osim liste radi i s vektorima i mapama

5) umjesto točkastih parova koristi ISeq sučelje za prolazak po listi

## 3. Logika

1. Navesti i objasniti na primjeru sintaksu predikatne logike. Što je dobro oblikovana logička formula?
2. Kako je određena semantika predikatne logike?
3. Navesti teorem dedukcije i njegov korolar.
4. Riješiti neki jednostavni primjer dokazivanja obaranjem.
5. Navesti teorem razrješavanja (Robinson) i svođenje elementarnih pravila na razrješavanje.
6. Navesti algoritam za opći postupak dokazivanja teorema.
7. Preslikajte neku formulu predikatne logike u skup normaliziranih klauzula u konjunkcijskoj normalnoj formi (CNF).
8. Objasnite strategiju skupa potpore i usporedite ju sa strategijom u širinu za razrješavanje.
9. Objasnite i na primjeru pokažite linearnu ulaznu strategiju razrješavanja koju koristi PROLOG.
10. Objasniti koja je svrha izjednačavanja i na primjeru pokazati izjednačavanje atomičkih predikata. Što je to najopćenitiji unifikator?
11. Navesti i objasniti na primjeru postupke pojednostavljivanja u skupu formula predikatne logike.
12. Kako glasi Goedelov teorem nekompletnosti i koje su njegove posljedice?

**1.**

Predikatna logika analizira izjavu uvodeći relacije, obilježja, funkcije i kvantifikatore varijabli te time pobliže opisuje odnose elemenata

formalno: (fun_simb t1 t2....tn)

npr. (otac_ob kain abel) → otac_ob – veza između objekta u rečenici; kain abel- objekti

Dobro oblikovana logička formula – ne postoji automatizirano preslikavanje

1) prefiks notacija $\forall X(čovjek(X) \Rightarrow smrtan(X))$

2) infiks notacija $\forall X \forall Y(((otac(XY) \vee majka(XY)) \Rightarrow roditelji(XY))$

**2.**

Proces pridruživanja obilježja istinitosti dobro oblikovanim formulama; T- istina, F- neistina

Interpretacija I je proces preslikavanja iz domene D svakoj pojedinoj konstanti, varijabli, funkciji i atomičnom predikatu

npr. $\forall X\ P(X)\ je\ T\ ako\ je\ P(X) T\ za\ \forall X\ iz\ I, inače\ F$

**3.**

Teorem dedukcije- formula ω je logička posljedica skupa formula (ili jedne složene) Γ, tj
$\Gamma| = \omega$ akko je formula ($\Gamma \Rightarrow \omega$) tautologija
Korolar- budući da ( $\Gamma \Rightarrow \omega$) mora biti tautologija, njena negacija $not(\Gamma \Rightarrow \omega) = \mathrm{not}(\mathrm{not}\,\Gamma \vee \omega)$= ( $\Gamma \vee not\ \omega$) mora biti nezadovoljiva

**4.**

---

Neka su istinite (konjunkcijom povezane) formule u skupu $\Gamma$:

1. $P$
2. $(P \Rightarrow Q)$
3. $(Q \Rightarrow S)$

Skup formula u CNF-obliku: $\Gamma = [\ (P) \ \wedge (\neg P \vee Q) \ \wedge \ (\neg Q \vee S)\ ]$

Neka vrijede uobičajena pravila slaganja dviju formula logičkim vezicama te dodatna pravila Modus ponens i Modus tolens (vidi ranije naveden skup pravila zaključivanja)

Pokaži da je formula $S$ logička posljedica gornjih formula.

Rješenje:

1. $P$
2. $(P \Rightarrow Q)$
3. $(Q \Rightarrow S)$
4. $\neg S$ ; teorem dedukcije: dodajemo negaciju formule S

Iz 1. i 2. Modusom ponensom slijedi $Q$ i dodaje se u $\Gamma$.
Iz 3. i 4. Modusom tolensom slijedi $\neg Q$ i dodaje se u $\Gamma$.
Konjunkcija tvrdnji $Q$ i $\neg Q$ je očita kontradikcija.

**Zaključak:** Formula $S$ je logička posljedica gornjih formula.

**5.**

teorem: Logička posljedica dviju istinitih konjukcijom vezanih, univerzalno kvantificiranih normaliziranih klauzula (sve formule moraju biti normalizirane klauzule) je normalizirana klauzula bez jednog komplementarnog para literala

**Modus ponens** je poseban slučaj razrješavanja:

$$\frac{P \qquad (P \Rightarrow Q)}{Q} \qquad\qquad \frac{P \qquad (\neg P \vee Q)}{Q}$$

**Modus tolens** je poseban slučaj razrješavanja:

$$\frac{\neg Q \qquad (P \Rightarrow Q)}{\neg P} \qquad\qquad \frac{\neg Q \qquad (\neg P \vee Q)}{\neg P}$$

**Lančano pravilo** je poseban slučaj razrješavanja:

$$\frac{(P \Rightarrow Q) \qquad (Q \Rightarrow R)}{(P \Rightarrow R)} \qquad\qquad \frac{(\neg P \vee Q) \qquad (\neg Q \vee R)}{(\neg P \vee R)}$$

**6.**

1) Definiraj skup aksioma E kao ispravne formule (wff)=stanje svijeta

2) definiraj teorem H kao ispravnu formulu (wff)

3) negirani H dodaj skupu aksioma E, dobiveni skup je formula F

4) preslikaj formulu F u skup univerzalno kvantificiranih normaliziranih klauzula K

**7.**

**Nemam pojma- to je u slajdovima!**

1. Eliminirati sve implikacije i ekvivalencije, te umjesto njih koristiti disjunkcije i konjunkcije, npr.:
   $(P \Rightarrow Q) = ((\neg P) \vee Q)$
   $(P \Leftrightarrow Q) = (((\neg P) \vee Q) \wedge (P \vee (\neg Q)))$
2. Pomaknuti negacije do jediničnih formula, npr.:
   $(\neg(\neg P)) = P$
   $(\neg(\forall X\, A(X))) = (\exists X\, (\neg A(X)))$
   $(\neg(\exists X\, A(X))) = (\forall X\, (\neg A(X)))$
3. Preimenovati varijable uz kvantifikatore u istoj složenoj formuli. Npr.:
   $(\forall X\, (P(X)) \Rightarrow (\exists X\, Q(X)))$
   Formule P i Q trebaju sadržati različite varijable (npr. X,Y),
   $(\forall X\, (P(X)) \Rightarrow (\exists Y\, Q(Y)))$
   Doseg varijable je samo unutar formula.
   Ponekad je korisno formule **standardizirati** (posebno označiti varijable).

4. Eliminirati kvantifikatore (skolemizirati - Thoralf Skolem).

4.1 Univerzalni kvantifikator (**podrazumijeva se, te se ispušta**)

(∀X ((prizma X) ⇒ (geom_tijelo X))) ;sve prizme su geom. tijela

pišemo jednostavnije:

((prizma X) ⇒ (geom_tijelo X))

4.2 Egzistencijski kvantifikator

Pretvorba varijable u novu konstantu ili funkciju, čiji su članovi univerzalno kvantificirane varijable.

4.2.1 Egzistencijski kvantifikator nije u dosegu univerzalnog:

(∃X((nudist X) ∧ (demokrat X)))

X supstitucija s novom jedinstvenom konstantom – **skolem konst.**
Dajemo ime nečemu što mora postojati,
jer ako je formula istinita, mora postojati barem jedna supstitucija.
Odaberemo npr: X = ab_1, te slijedi:
((nudist ab_1) ∧ (demokrat ab_1))

4.2.2 Egzistencijski kvantifikator je u dosegu univerzalnog:

**(∀X (∃Y P(X,Y)))** ; predikat P sadrži članove-varijable X i Y

Postoji neki Y i nekako ovisi o (odabranom) X.
Dajemo novi simbol za Y, (uvodimo funkcijski član
- skolem funkciju), jer je Y različit za svaki odabrani X.
Y = f(X), pa uz izostavljanje ∀ slijedi:

**P(X, f(X))** ; prefiks notacija

Npr. "Svaka osoba ima majku.“
∀X ∃Y (majka X Y) ; infiks notacija
Varijabla X označuje svaku osobu.
Y označuje određenu (ne svaku), jer ovisi o X.
Y zamjenjujemo s funk. članom (m X), pa (uz ispuštanje ∀):
(majka X (m X)) ; u infiks notaciji

5. konjunkcijom složene formule se rastavljaju i pišu se svaka posebno.

$(A \lor (B \land C)) = (A \lor B) \land (A \lor C)$

Formule $(A \lor B)$ i $(A \lor C)$ pišu se kao odvojene klauzule.

- Nakon izvedenog preslikavanja slijedi:
  **skup normaliziranih klauzula u CNF** (konjunkcijskom normalnom) obliku - konjunkcija disjunkcija.

**8.**

Za skup ulaznih klauzula K može se definirati podskup T c K (skup potpore). Ova strategija zahtjeva da barem jedna od klauzula u razrješavanju bude ili ima pretka u T. Ako je K nezadovoljiv, a K-T zadovoljiv, strategija skupa potpore je kompletna. Zahtjeva da barem jedna od klauzula bude negirani zaključak.

Smanjuje prostor stanja strategije izbora u širinu, te traži samo sva razrješenja s negiranim zaključkom i njegovim potomcima, a ne sva kao u strategiji u širinu.

**9.**

Polazi od negiranog zaključka i jedne klauzule iz skupa aksioma. U svakom sljedećem koraku razrješava se uvijek s rezultatom iz prijašnjeg koraka i jednom novom klauzulom iz skupa aksioma. Nikad se ne koriste klauzule generirane u nekom od ranijih koraka. Nije kompletna!

PR.

Primjer: Ako ma koja klauzula predstavljala negirani zaključak, linearnom ulaznom strategijom nije moguće doći do prazne klauzule (dokaza).

**1. $(A \lor B)$**

**2. $(A \lor \neg B)$**

**3. $(\neg A \lor B)$**

**4. $(\neg A \lor \neg B)$**

- Očito da iz 1 i 2 slijedi **A,** a 3 i 4 slijedi **¬A**, te je skup nekonzistentan.
- Ako pak koristimo linearnu ulaznu strategiju, polazimo npr.
- od 1, te 1 i 2 daju **5: A**, 5 i 3 daje **6: B**, 6 i 4 daje **7: ¬A**, i nema ()
- Slično bez obzira na polaznu klauzulu.

## 10.

Najbitniji dio alata za automatizuirano rasuđivanje; u razrješavanju tražimo komplementarne parove; u predikatnoj logici jedinični predikati s istim predikatnim simbolom i jednakim brojem članova možda se mogu izjednačiti.

1. (r1 Y Z) ; r1 je predikat, Y, Z su varijable
2. ((¬(r1 a X)) ∨ (r2 X b)) ; X je varijabla, a i b su konst.

Supstitucija **Y=a, Z=a1** u prvoj klauzuli, te **X=a1** u drugoj daje:

1. (r1 a a1)
2. (¬(r1 a a1)) ∨ (r2 a1 b))

(…) su kompl. parovi. Razriješena klauzula je (r2 a1 b).

**Postoji i općenitije rješenje.**

- Ako u prvoj klauzuli supstituiramo: **Y=a, Z=Xnovi** (varijablu zamijenimo s varijablom), a u drugoj **X=Xnovi**, slijede klauzule:
  1. (r1 a Xnovi)
  2. (¬(r1 a Xnovi)) ∨ (r2 Xnovi b))
- Razriješena klauzula (r2 Xnovi b) je općenitija, sadrži varijablu Xnovi koja se može dalje koristiti (u slijedećim koracima).
- Traži se supstitucija koja daje **najopćenitije** rješenje, **unifikator** (supstitucijski parovi) je tada **MGU** (*engl. most general unifier*).

## 11.

1) upotreba temeljnih pravila:

| | |
|---|---|
| (P ∨ R ∨ P) | pojednostavi (spoji) u (P ∨ R) |
| (P ∨ ¬P ∨ Q) | izostavi cijelu jer je evidentno valjana (T) |

2) podrazumijevanje klauzula- klauzula $\omega_1$ podrazumijeva klauzulu $\omega_2$ ako su literali u $\omega_1$ podskup literala u $\omega_2$

npr.

(P ∨ $R$) podrazumijeva klauzule (P ∨ $R$ ∨ $Q$)

**12.**

U FOPL se uvodi skup prirodnih brojeva (N) i aritmetika

**teorem**: Ne postoji konzistentan i kompletan sustav dokazivanja za FOPL+N.

Aritmetika omogućuje izgradnju kodnog sustava (Gödelovi brojevi) za izraze u FOPL+N

Npr.: P = “P je nedokazljiv”

Ako je P istinit, tada je P nedokazljiv (**nekompletnost !**).

Ako je P neistinit, tada je P dokazljiv (**nekonzistentnost !**).

## 4. Pravila

1. Koje su razlike između sustava s pravilima i formalne logike u modeliranju znanja ?
2. Imaju li sustavi s pravilima proceduralan upravljački tijek ili neproceduralan, te zašto ?
3. Navedite barem tri potencijalne poteškoće pri zaključivanju s pravilima.
4. Koja su dva temeljna postupka zaključivanja u sustavima s pravilima? Koncizno opišite izvođenje oba postupka.
5. Što je to "konfliktni skup" kod ES zasnovanih na pravilima (u sustavu CLIPS "Agenda") ?
6. Kada treba odabrati ulančavanje unaprijed a kada unatrag ? (objasnite posebno za slučaj moguće analize pravila, a posebno za slučaj analize činjenica).
7. Kako algoritam Rete uspješno ubrzava zaključivanje u sustavima s pravilima?
8. Navedite četiri jezgrena modula u klasičnom ES temeljenom na pravilima.
9. Navedite barem pet prednosti sustava s pravilima.
10. Navedite barem četiri nedostatka sustava s pravilima.

**1.**

Kod formalne logike materijalna implikacija nije potpuno intuitivna prirodnom jeziku (modelira uvjetnu konstrukciju, a ne uzročno posljedičnu vezu) dok kod pravila imamo ako-onda princip.

**2.**

Sustavi s pravilima imaju neproceduralan tijek jer se traži slaganje činjenica s AKO stranama u pravilima, pravila koja postignu slaganje se izvode

**3.**

1) činjenice mogu biti složene strukture podataka (usporava poklapanje)

2) slaganje činjenica s više pravila (mogućnost konflikta)

3) konjukcija u AKO stranama pravila (produljeno vrijeme poklapanja)

4) povezani ciklusi zaključivanja (mogućnost ulaska u beskonačnu petlju)

**4.**

1) zaključivanje ulančavanjem unatrag- korisnik postavlja upit ili cilj koji se pretražuje u činjenicama, pa ako se tamo ne nalazi, nastoji se pretražiti u ONDA stranama čija se potvrda onda traži u AKO stranama istih pravila ili u činjenicama

$$P(a)$$

$$(P(X) \rightarrow Q(X))$$

? P(a) - ovdje odmah odgovor DA.

? R(b) - nema u činjenicama, a također je i **ONDA** strana pravila različita, odgovor je NE.

? Q(a) - nema u činjenicama, ali se **odgovarajućom supstitucijom** varijabli (X = a) može pronaći u ONDA strani pravila.

Supstitucija se mora provesti u cijelom pravilu. Novo stanje sustava je:

$$P(a)$$
$$(P(a) \rightarrow Q(a))$$

2) zaključivanje ulančavanjem unaprijed – postoji pravilo (P(X)→Q(X)). U trenutku upisa P(a) u sustav započinje automatska potraga za pravilom koje u svakom AKO dijelu ima strukturu koja bi se mogla izjednačiti s P(a). Supstitucijom X=a to se nalazi pa je stanje (P(a)→Q(a))

**5.**

Neka se u sustav zapišu činjenice: C1, C3, C5, a pravila su:

1. $((C1 \wedge C2) \rightarrow A1)$
2. $(C3 \rightarrow A2)$
3. $((C1 \wedge C3) \rightarrow A3)$
4. $(C4 \rightarrow A4)$
5. $(C5 \rightarrow A5)$

- Stanje činjenica zadovoljava (lijevu stranu) pravila: 2, 3, 5.
- Sva ta pravila čine tzv. **konfliktni skup** (u sustavu CLIPS – **agenda**).
- Potrebno je **izabrati jedno i samo jedno** koje će generirati neku akciju.
- Mogući postupci izbora: prvo iz konfliktnog skupa, slučajno, težine, … .
- U općem slučaju, faze **ciklusa zaključivanja** ulančavanjem unaprijed su:
  - **usporedi** (engl. *match*) AKO strane pravila s činjenicama,
  - **razriješi** (engl. *resolve*) konfliktni skup i
  - **djeluj** (engl. *act*), tj. aktiviraj odabrano pravilo.

**6.**

Činjenice:

1)ako imamo puno činjenica poznato, a potrebno je vidjeti kamo to vodi→ulančavanje unaprijed

2) ako imamo malo činjenica a cilj je potvrditi jednu od mnogih hipoteza→ulančavanje unatrag

Pravila:
1) ukoliko postoji mnogo uvjeta u 'tipičnom pravilu' →ulančavanje unaprijed

2) ukoliko postoji malo uvjeta → ulančavanje unatrag

**7.**

Koristi već dijelom izvršenu usporedbu u ranijim ciklusima, kao i to da u skupu pravila postoje neki identični dijelovi lijevih strana. Pri upisu pravila algoritam gradi mrežu uvjetnih zavisnosti, koja povezuje sva pravila s jednakim dijelovima AKO strane.

**8.**

1)skup pravila (baza znanja)

2) skup činjenica (radna memorija)

3) stroj za zaključivanje (inference engine)

4) sučelje s korisnikom

**9.**

1) izravni prirodni prikaz i korištenje iskustvenog znanja prikupljenog od eksperta

2) modularnost zbirke znanja i njeno jednostavno održavanje

3) efikasno izvođenje u uskim domenama

4) dobra obilježja razjašnjavanja procesa automatiziranog rasuđivanja

5) skup pravila se prirodno preslikava u problem pretraživanja prostora stanja

6) koraci tijekom rješavanja problema su transparentni

7) razdvajanje upr. toka i zbirke znanja olakšava projektiranje i izradu inteligentnih sustava

**10.**

1) pravila prikupljena od eksperta su heuristička i fragmentirana i ne objašnjavaju cjelovite odnose u procesu

2) sustav s pravilima naglo prestaje bit uporabiv na rubu domene

3) razjašnjenja, koja slijede iz prikaza ulančavanja, ne daju kauzalne veze u procesu

4) u razvoju sustava postoji poteškoća razumijevanja između inženjera i eksperta

## 5. Nesavršeno znanje

1. Objasnite na primjerima dva načina razumjevanja semantike težinskih faktora pridruženih pravilima.
2. Što su neizrazite (lingvističke) varijable i njihovi modifikatori? Čime je opisana lingvistička varijabla? Navesti primjere.
3. Navedite definicije logičkih operacija unije, presjeka i komplementa u neizrazitoj logici.
4. Koje su "inženjerske" definicije implikacije u neizrazitoj logici? Zašto su one uvedene?
5. Kako se zaključuje (postupak izvođenja zaključka) u poopćenom Modus ponensu?
6. Objasnite postupak zaključivanja s neizrazitim pravilima:
   - Činjenice izrazite, AKO strana pravila neizrazita, TADA strana pravila izrazita.
   - Činjenice izrazite, AKO strana pravila neizrazita, TADA strana pravila neizrazita.
   - Činjenice neizrazite, AKO strana pravila neizrazita, TADA strana pravila izrazita.
   - Činjenice neizrazite, AKO strana pravila neizrazita, TADA strana pravila neizrazita.

   Postupak treba objasniti sve do uključivo izvođenja izrazitog zaključka temeljem svih aktiviranih pravila.

**1.**

1) neizrazitost- npr. 'Temperatura je **visoka'** –neizrazita vrijednost, ali traži pripadnu mjeru

2) neizvjesnost- npr. 'Temperatura pare je 280°C (**0.8)'**- mjera nepotpunog poznavanja temperature

**2.**

Lingvistička varijabla je subjektivan opis stanja (toplo, nisko, visoko, itd. umjesto precizno 25°C, 2m, itd.), a njezini modifikatori su rangovi tog stanja (vrlo, malo, manje-više, itd.)

Lingvistička varijabla je opisana vrijednostima karakteristične funkcije (diskretnim ili kontinuiranim)

PR. A~=(0/1, 1/1, 2/1, 3/0.8, 4/0.6 ...) →X/1 potpuno pripadaju, X/0.8 manje pripadaju, X/0.6 još manje....

**3.**

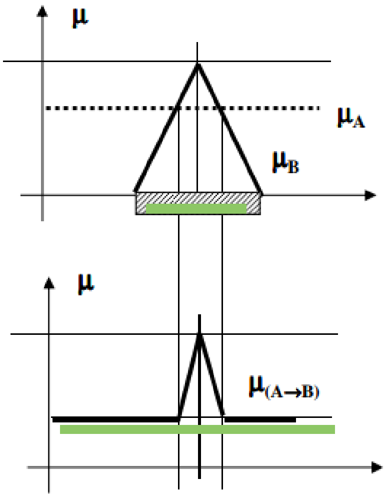
| | |
|---|---|
| $\mu A' = 1 - \mu A$ | (komplement) |
| $\mu(A \cap B) = \min[\mu A, \mu B]$ | (presjek) |
| $\mu(A \cup B) = \max[\mu A, \mu B]$ | (unija) |

**4.**

- Problem implikacije: $\mu(A \to B) = I - \min[\mu(A), (I - \mu(B))]$

Implikacija preslikava pravilo: **Ako A onda B.**
Ako je domena od **B** konačna i implikacija mora rezultirati u nečem konačnom.

Protuprimjer (domena implikacije je beskonačna):




Predložene su jednostavnije **inženjerske definicije** implikacije:

**1.** $\mu(A \to B) = \min[\mu(A), \mu(B)]$

**min**, Mamdani, 1974

**2.** $\mu(A \to B) = [\mu(A) \cdot \mu(B)]$

**produkt**, Larsen, 1980

16

**5.**

Premise kao neizraziti skupovi predstavljene su karakterističnim funkcijama, a zaključak je njihova kompozicija: $\mu(B2) = \mu(A1)°\mu(A2 \to B1)$

**6.**

## prezentacija 5, slajdovi 23-29

1) činjenice izrazite, AKO strana pravila neizrazita, TADA strana pravila izrazita

2) činjenice izrazite, AKO strana pravila neizrazita, TADA strana pravila neizrazita

3) činjenice neizrazite, AKO strana pravila neizrazita, TADA strana pravila izrazita

4) činjenice neizrazite, AKO strana pravila neizrazita, TADA strana pravila neizrazita

## 5. Nesavršeno znanje

7. Navedite dva postupka kompozicije svih pravila u izvođenju neizrazitog zaključka.
8. Pretpostavite da imate sustav s izrazitim činjenicama koje imaju faktore izvjesnosti i izrazitim pravilima koja imaju faktore izvjesnosti. Kako se računa faktor izvjesnosti zaključka jednog pravila?
9. Pretpostavite da imate sustav s neizrazitim činjenicama koje imaju faktore izvjesnosti i neizrazitim pravilima koja imaju faktore izvjesnosti. Kako se računa faktor izvjesnosti zaključka jednog pravila?
10. Objasnite riječima što je to "mjera sličnosti" u ES s neizrazitim pravilima i kako se upotrebljava.
11. U ekspertnom sustavu s faktorima izvjesnosti (činjenica i pravila), kako se računa faktor izvjesnosti činjenice koja se temeljem aktivacije nekog pravila upisuje u skup činjenica?
12. Kako se izračunava višestruki doprinos jednoj hipotezi (višestruko upisivanje iste činjenice (izrazite ili neizrazite) ?

**7.**

max-min i max-produkt

**8.**

Činjenice:**izrazite** s faktorima izvjesnosti CF1 ... CFn

Pravilo: AKO strane **izrazite**, ONDA strana izrazita ili ne, faktor izvjesnosti pravila $CF_R$ .

Novi faktor izvjesnosti pravila:

$$CF_C = CF_R \cdot \min[CF1, \dots, CFn]$$

**9.**

Činjenica: jedna **neizrazita**, faktor izvjesnosti $CF\tilde{}_I$ (npr. 0.8)

Pravilo: AKO strana **neizrazita**,
ONDA strana izrazita ili ne,
faktor izvjesnosti pravila $CF\tilde{}_R$ (npr. 0.7)

Novi faktor izvjesnosti pravila:

$$CF_C = (CF\tilde{} \cdot S_I) \cdot CF\tilde{}_R$$

gdje je $S_I$ = **mjera sličnosti** činjenice i AKO dijela s kojim se činjenica prekriva.

**10.**

Preklapanje $\mu F$ (karakteristična funkcija činjenice) i komplementa $\mu R$ (1- $\mu R$)

$\mu R$- karakteristična funkcija jednog konjukcijskog dijela AKO strane

**11.**

Pravilo se aktivira ($F1 \leftrightarrow AKO\ F1$) i nova činjenica F2 upisuje se u skup činjenica, ali s promijenjenim faktorom $CF_{F2}$

**12.**

CF= max [$CF_{Fn}, CF_F$]

### 6. Bayesove mreže

1. Dati definicije: slučajne varijable, uvjetne vjerojatnosti, marginalne razdiobe, pravila zbrajanja, lančanog pravila, Bayesovog pravila.
2. Što je to Bayesova mreža?
3. Što je to Markovljeva uvjetna nezavisnost?
4. Navedite postupak oblikovanja Bayesove mreže.
5. Zašto u Bayesoj mreži nije potrebno poznavati svih $2^n$ vjerojatnosti za $\boldsymbol{n}$ Booleovih slučajnih varijabli koje opisuju neku domenu?
6. Objasnite rasuđivanje u Bayesovoj mreži. Koji je najčešći cilj tog rasuđivanja?
7. Za neku jednostavnu Bayesovu mrežu izračunajte:
   - Bezuvjetne (marginalne) razdiobe vjerojatnosti.
   - Uzročno djelovanje.
   - Dijagnostičko djelovanje.
8. Za koji oblik Bayesove mreže je moguća polinomska složenost rasuđivanja?
9. Pri proširenju Bayesovih mreža na mreže utjecaja i odlučivanja, koje vrste čvorova možemo naći u mreži? Objasnite njihovu ulogu.
10. Za neku jednostavnu mrežu utjecaja i odlučivanja, potrebno je izračunati maksimalnu očekivanu korisnost i na temelju nje donijeti racionalnu odluku.

**1.**

Slučajna varijabla- varijabla koja može poprimiti vrijednost iz skupa isključivih i potpunih vrijednosti s određenom vjerojatnošću

Uvjetna vjerojatnost P(A|B)- vjerojatnost da će se dogoditi A ako se dogodio B

$$P(A|B) = \frac{P(A,B)}{P(B)}$$

Marginalne razdiobe- razdioba nekog fiksiranog podskupa tih varijabli, odnosno razdioba tog podskupa bez obzira na vrijednosti u ostatku skupa

Pravilo zbrajanja- P(A=a)=$\sum_a P(A=a|B=b) \cdot P(B=b)$

Lančano pravilo - Rastavljanjem zajedničke razdiobe na faktore uvjetnih vjerojatnosti smanjuje se dimenzionalnost ( N na N-1)

Bayesovo pravilo – $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$

**2.**

Bayesova mreža je usmjereni aciklički graf G=(V,E) gdje su:

V- čvorovi, slučajne varijable

E- lukovi, pokazuju izravnu uzročnu vezu

**3.**

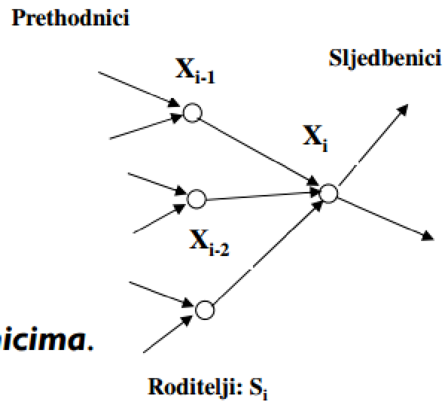
**Markovljeva uvjetna nezavisnost**

Promatramo jedan čvor $X_i$ :

Ako je skup roditelja ***$S_i$ poznat*** (poznate su vjerojatnosti svih vrijednosti), tada:

$P(X_i \mid X_{i-1}, X_{i-2}, \ldots, X_2, X_1) = P(X_i \mid S_i)$

tj. vjerojatnost od $X_i$ ***ne ovisi o ostalim prethodnicima***.




Zajednička razdioba cijele mreže dana je lančanim pravilom:

$$\mathbf{P(X_1, \ldots, X_n)} = P(X_n \mid X_{n-1}, \ldots, X_1)\, P(X_{n-1} \mid X_{n-2}, \ldots, X_1) \ldots P(X_2 \mid X_1)\, P(X_1)$$
$$= \prod_{i=n..1} P(X_i \mid X_{i-1}, X_{i-2}, \ldots, X_2, X_1)$$
$$= \prod_{i=n..1} \mathbf{P(X_i \mid S(X_i))}$$

Opažamo smanjenje složenosti mreže: ***zajednička razdioba je produkt lokalnih uvjetnih razdioba.***

**4.**

1) odaberi skup sluč. var. koje opisuju domenu problema

2) uredi skup varijabli (povlačenje lukova) tako da najprije odrediš najranije prethodnike (var. koje nemaju roditelja), a zatim var. na koje one izravno utječu (neposredne lok. uzročne veze)

3) ponavljaj 2. do krajnjih varijabli ( djece)

4) definiraj tablicu lokalnih uvjetnih vjerojatnosti svake varijable. Pri tome broj roditelja neke varijable određuje dimenzionalnost njene lokalne tablice vjerojatnosti

**5.**

Jer je za varijablu bez roditelja potrebno odrediti samo jednu vjerojatnost

**6.**

1) temeljem tablica lokalnih uvjetnih vjerojatnosti varijabli, mogu se izračunati bezuvjetne (marginalne) razdiobe svih varijabli

2) uz poznate vrijednosti skupa evidencijskih varijabli $\{E_e\}$ mogu se odrediti vjerojatnosti skupa upitnih varijabli $\{Q_q\}$

→tipovi uvjeta: dijagnostika, uzročno rasuđivanje, međuuzročno rasuđivanje, miješano rasuđivanje

→cilj: donošenje odluka

**7.**


definirano: P(A=T), P(B=T|A=T), P(B=T|A=F)

1. P(B=T)=$\sum_a P(B = T|A = a)P(A = a)$ a={T,F}

2. P(B=T|A=T) očitamo

3. $\mathrm{P(A = T|B = T)} = \frac{\mathrm{P(B=T|A=T)P(A=T)}}{P(B=T)}$

**8.**

Polinomska složenost rasuđivanja je moguća za topologiju jednostruko povezanih čvorova (polistablo)- postoji samo jedan put između 2 čvora (uvijek samo jedan roditelj)

**9.**

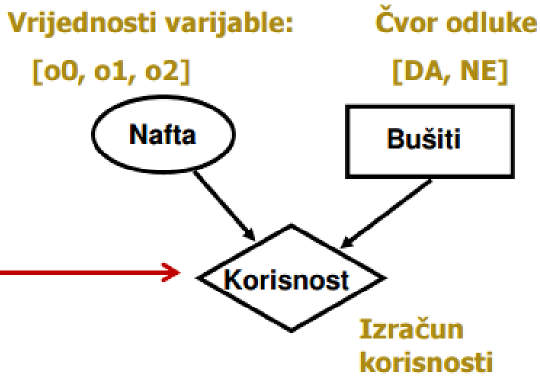
1) čvorovi odluke- točke u kojima agent ima izbor mogućih akcija

2) čvorovi izglednosti- predstavljaju slučajne varijable, roditelji ovih čvorova mogu biti čvorovi izglednosti i odluke

3) čvorovi korisnosti- predstavljaju dijelove agentove funkcije korisnosti

**10.**

Poduzetnik treba odlučiti da li da buši u traženju nafte na određenom mjestu ili ne. Pretpostavljamo da je korisnost = novac.




Na tom mjestu postoje vjerojatnosti:

Suho (nema nafte) o0 = 0.5

Mokro (malo nafte) o1 = 0.3

Razmočeno (mnogo nafte) o2 = 0.2

Tablica korisnosti $U(D_i)$ = novac :

| | o0 | o1 | o2 |
|---|---|---|---|
| D1 (bušiti): | -70 | 50 | 200 |
| D2 (ne bušiti): | 0 | 0 | 0 |

; npr. u M kuna

Očekivana vrijednost/korisnost za svaku akciju:

$EU = \Sigma_i\, p_i\, U(D_i)$

EU(ne bušiti) = 0

EU(bušiti) = 0.5 x (-70) + 0.3 x 50 + 0.2 x 200 = 20

***MEU = 20*** , zaključak: treba bušiti

(by TooSheik)