Uvod u umjetnu inteligenciju – pismeni ispit (2020./2021.)

– NEKORIGIRANA VERZIJA –

*Ispit se sastoji od **20 pitanja** i ukupno nosi **20 bodova**. Točan odgovor nosi 1 bod a netočan -1/3 boda. Trajanje ispita je **150 minuta**. Primjerak ispita trebate predati zajedno sa svojim rješenjima.*

1. Pretraživanje prostora stanja i igranje igara (5 pitanja)

1 (T) Za razliku od slijepog pretraživanja prostora stanja, heurističko je pretraživanje usmjereno procjenom udaljenosti trenutačnog stanja do ciljnog stanja. **Koja je prednost heurističkog pretraživanja prostora stanja nad slijepim pretraživanjem?**

- A Reducira apriornu vremensku složenost pretraživanja, uz uvjet da se koristi skup posjećenih stanja
- B Smanjuje očekivan broj koraka izvođenja algoritma, a time i broj proširenih čvorova te zauzeće memorije
- C Ima gornju ogradu na apriornu vremensku i prostornu složenost pretraživanja
- D Ako je heuristika optimistična, osigurava potpunost, odnosno nalaženje ciljnog stanja ako ono postoji

2 (R) Prostor stanja pretražujemo algoritmom A^*. Skup stanja je $S = \{a, b, c, d, e, f\}$, a funkcija sljedbenika je $succ(a) = \{(b, 2), (c, 2)\}$, $succ(b) = \{(c, 5), (d, 2)\}$, $succ(c) = \{(d, 1), (f, 20)\}$, $succ(d) = \{(e, 2)\}$, $succ(e) = \{(f, 14)\}$ te $succ(f) = \emptyset$. Heurističke vrijednosti stanja su $h(a) = 16$, $h(b) = 6$, $h(c) = 14$, $h(d) = 4$, $h(e) = 2$, $h(f) = 0$. Početno stanje je a, a ciljno f. Izvedite korake algoritma A^*, bilježeći u svakom koraku sadržaj liste otvorenih čvorova O i skupa zatvorenih čvorova C. U nultom koraku algoritma vrijedi $O = [(a, 0)]$ i $C = \emptyset$. **Koji je sadržaj liste O i skupa C nakon šestog koraka izvođenja algoritma?**

- A Algoritam ne dostiže šesti korak
- B $O = [(c, 2), (f, 20)]$, $C = \{(a, 0), (b, 2), (d, 4), (e, 6)\}$
- C $O = [(e, 5), (f, 20)]$, $C = \{(a, 0), (b, 2), (c, 7)\}$
- D $O = [(e, 5), (f, 20)]$, $C = \{(a, 0), (b, 2), (c, 2), (d, 3)\}$

3 (P) Razmatramo problem slagalice dimenzija 5×5 s prosječnom dubinom rješenja $d = 10$. Za nalaženje rješenja algoritam pretraživanja u dubinu u prosjeku koristi 1 KB memorije. **Koliko bi memorije u prosjeku koristio algoritam pretraživanja u širinu na ovome problemu?**

A 25.6 MB B 5 MB C 3.4 MB D 34 MB

4 (R) Stablo igre definirano je prijelazima $succ(A) = \{B, C\}, succ(B) = \{D, E\}, succ(D) = \{H, I\}, succ(E) = \{J, K, L\}, succ(C) = \{F, G\}, succ(F) = \{M, N, O\}, succ(G) = \{P, Q\}$. Heurističke vrijednosti listova su $h(H) = -12, h(I) = -4, h(J) = -14, h(K) = 17, h(L) = 18, h(M) = -10, h(N) = -15, h(O) = -13, h(P) = 0, h(Q) = 11$. Optimalna strategija određuje se algoritmom minimax uz alfa-beta podrezivanje. **Koji će čvorovi pritom biti podrezani (preskočeni pri izračunu minimax vrijednosti)?**

A P, Q B N, O, P, Q C L, G, P, Q D I, K, L, F, M, N, O, Q

5 (T) Algoritam minimax koristi heurističku funkciju koja se poziva nad svim nezavršnim stanjima u stablu igre na određenoj dubini k. Igraći algoritmi općenito koriste različite heurističke funkcije. No, **što bi se dogodilo kada bi u igri sudjelovala dva algoritma koji imaju identične heurističke funkcije?**

- A Ako je mehanizam odabira poteza kod izjednačavanja minimax-vrijednosti poznat i determinističan, ishod igre mogao bi se izračunati na početku igre
- B Prvi igrač mogao bi unaprijed izračunati sve poteze protivnika do dubine k te tako odrediti globalno optimalnu minimax-strategiju
- C Ništa se posebno ne bi dogodilo, jer i dalje oba algoritma ne bi mogla pretraživanjem u dubinu dosegnuti konačna stanja igre s definiranim isplatama
- D Igrač koji je prvi na potezu bi pobijedio, osim ako u igri ima elemenata slučajnosti, i tada bi igra bila izjednačena

2. Prikazivanje znanja, automatsko zaključivanje i logičko programiranje (5 pitanja)

6 (T) Rezolucija opovrgavanjem je ispravno i potpuno pravilo zaključivanja u FOL. **Što to znači?**

- A Ako formula nije logička posljedica, onda to ne možemo dokazati rezolucijom opovrgavanjem
- B Postupak izvodi klauzulu NIL ako i samo ako je negiran cilj logička posljedica premisa
- C Kad god je skup premisa nekonzistentan, rezolucijsko pravilo izvodi NIL klauzulu
- D Postupak završava u konačnom broju koraka s odlukom je li formula logička posljedica premisa ili nije

7 (P) Za automatsko zaključivanje u PL koristimo rezoluciju opovrgavanjem uz strategiju skupa potpore (SOS). Na takav postupak zaključivanja možemo gledati kao na problem pretraživanje prostora stanja, gdje stanja odgovaraju skupu klauzula (onih zadanih i onih izvedenih), a prijelazi između stanja odgovaraju primjeni rezolucijskog pravila na jedan par klauzula. Takav problem pretraživanje ima i svoj faktor grananja, koji ovisi o dubini stabla, tj. o koraku zaključivanja. Neka skup premisa sadrži 10 klauzula, a negirani cilj 5 klauzula. **Koliko iznosi gornja ograda na faktor grananja u drugom koraku zaključivanja?** (Napomena: Jednom razriješeni par klauzula više se ne razrješava.)

A 60   B 104   C 74   D 119

8 (P) Važno svojstvo pravila zaključivanje jest da je ono ispravno. **Koje je od sljedećih pravila zaključivanja ispravno?**

- A $A \to (B \to C) \vdash (A \wedge B) \to C$
- C $A \to B, B \to C \vdash C \to A$
- B $\neg A, A \to B \vdash \neg B$
- D $A \vee B \vdash B$

9 (R) Zadane su sljedeće dvije premise:

$$\exists x\Big(\exists y R(y) \vee \forall z \neg Q(x,z)\Big) \to P(a),\ \exists x \forall y\Big(Q(x,y) \to R(y)\Big)$$

Koju je od sljedećih formula moguće dokazati iz ovih premisa rezolucijom opovrgavanjem?

A $\forall x(\exists y P(y) \vee R(x))$ B $Q(a,b)$ C $\forall x P(x)$ D $\exists x \forall y P(y)$

10 (T) Programi u Prologu sastoje se od definitnih klauzula, koje odgovaraju pravilima i činjenicama. **Što je razlika između pravila i činjenica u Prologu?**

- A Pravila imaju barem jedan negativan literal i točno jedan pozitivan literal, dok činjenice odgovaraju pravilima s uvijek istinitim antecedentom
- B Pravila imaju antecedent s točno jednim pozitivnim literalom, dok su činjenice jedinične klauzule sa samo jednim pozitivnim literalom
- C Pravila imaju antecedent s barem jednim pozitivnim literalom, dok su činjenice klauzule sa samo negativnim literalima
- D Pravila imaju najviše jedan negativan literal i barem jedan pozitivan literal, dok činjenice odgovaraju pravilima s uvijek lažnim antecedentom

3. Modeliranje neizvjesnosti (2 pitanja)

11 (P) Razmotrimo jezičnu varijablu *starost kanarinca*, definiranu s izrazima *mlad* i *star*. Značenje tih izraza modeliramo neizrazitim skupovima M odnosno S, definiranima nad univerzalnim skupom $\mathbb{R}^+$, koji odgovara godinama. Funkcije pripadnosti μ_M i μ_S definiramo kao po dijelovima linearne funkcije. Funkcija $\mu_M(x)$ ima vrijednost 1 za $0 \leq x \leq 5$, vrijednost 0 za $x \geq 9$, te linearno pada za $5 < x < 9$. Funkcija $\mu_S(x)$ ima vrijednost 0 za $0 \leq x \leq 6$, vrijednost 1 za $x \geq 12$, te linearno raste za $6 < x < 12$. Uporabom Zadehovih operatora definiramo dva neizrazita skupa: skup X_1 sa značenjem *mlad ili star kanarinac* te skup X_2 sa značenjem *mlad ili ne mlad kanarinac*. **Za koje se sve elemente $x \in \mathbb{R}^+$ pripadnost ovim dvama neizrazitim skupovima razlikuje?**

A $7 \leq x \leq 9$ B $7 \leq x \leq 12$ C $9 \leq x \leq 12$ D $5 \leq x \leq 6$

12 (R) Bayesovom shemom zaključujemo o vjerojatnosti da neka država provodi nuklearne pokuse. Apriornu vjerojatnost da država provodi nuklearne pokuse, $P(N)$, izračunavamo kao procjenu najveće izglednosti (MLE) na temelju informacije da je za devet od 195 država poznato da imaju nuklearno oružje te da svaka treća od njih aktivno provodi nuklearne pokuse. Kao dokaze provođenja nuklearnog pokusa koristimo podatak o detekciji snažne sezmičke aktivnosti (S) te podatak o intenzivnoj proizvodnji obogaćenog uranija (U). Znamo da je vjerojatnost detekcije seizmičke aktivnosti uslijed nuklearnog pokusa jednaka 0.8, a vjerojatnost sezmičke aktivnosti bez nuklearnog pokusa svega 0.002. Također znamo je vjerojatnost proizvodnje obogaćenog uranija ako je proveden nuklearni pokus 0.5, dok bez provođenja nuklearnog pokusa ta vjerojatnost iznosi 0.1. Izračunajte aposteriornu vjerojatnost provođenja nuklearnog pokusa, i to prvo samo uz dokaz U, a zatim uz dodatni dokaz S, pretpostavljajući pritom uvjetnu nezavisnost tih dvaju dokaza. **Koliko iznosi porast aposteriorne vjerojatnosti nakon dodavanja drugog dokaza?**

A 0.853 B 0.897 C 0.785 D 0.714

4. Strojno učenje i umjetne neuronske mreže (5 pitanja)

13 (T) Stabla odluke i Bayesov klasifikator dva su algoritma nadziranog učenja. Međutim, ti se algoritmi vrlo razlikuju. **Što je prednost, a što nedostatak stabla odluke u odnosu na Bayesov klasifikator?**

A Prednost je što primjere možemo klasificirati u više od jedne klase, a nedostatak što se stablo odluke može lako prenaučiti

B Prednost je što su stabla odluke otporna na male promjene u ulaznom skupu podataka, a nedostatak što pretpostavljamo uvjetnu nezavisnost značajki

C Prednost je što stablo odluke možemo podrezati kako bismo spriječili prenaučenost, a nedostatak što može doći do podljeva pri računanju vjerojatnosti

D Prednost je što možemo bolje objasniti zašto je primjer klasificiran u neku klasu, a nedostatak što nemamo vjerojatnost klasifikacijske odluke

14 (R) Mali je Ivica svakog svakog ljeta u zadnjih sedam godina naučio jedan novi programski jezik. Svoja vrijedna iskustva sažeo je u listu *"Programski jezik koji mi se sviđa"*, gdje je svaki jezik opisao četirima značajkama, te je naznačio je li mu se dotični jezik svidio ($y = 1$) ili nije ($y = 0$). Ta lista izgleda ovako:

i	Evaluacija (E)	Izvođenje (I)	Paradigma (P)	Provjera tipova (T)	y
1	lijena	kompajler	imperativna	statička	0
2	striktna	interpreter	deklarativna	dinamička	0
3	lijena	kompajler	imperativna	dinamička	0
4	lijena	interpreter	hibridna	statička	0
5	striktna	interpreter	imperativna	statička	1
6	lijena	kompajler	hibridna	dinamička	1
7	striktna	kompajler	hibridna	dinamička	1

Ovog ljeta Mali Ivica želi puno jesti i spavati te opet naučiti novi programski jezik. Međutim, Mali Ivica ne želi izgubiti ljetno na programskom jeziku koji mu se ne sviđa, pa je odlučio izgraditi stablo odluke za klasifikaciju programskih jezika koje još nije naučio. Pomozite Malom Ivici te na gornjem skupu primjera primjenite algoritam ID3. U slučaju da dvije značajke ili više njih imaju jednaku vrijednost informacijske dobiti, prednost dajte značajki koje je u gornjoj tablici navedena prva (ljevija značajka). **Koje se značajke nalaze u drugoj razini stabla odluke dobivenog algoritmom ID3?**

A E, I B E, I, P C E, P, T D E, P

15 (P) Na računalu implementiramo unaprijednu potpuno povezanu slojevitu umjetnu neuronsku mrežu arhitekture $3 \times 40 \times 10 \times 5 \times 2$. Neuroni kao prijenosne funkcije koriste zglobnicu. U memoriji težine mreže čuvaju se kao tip `double` koji zauzima 8 okteta. **Koliki je ukupni utrošak memorije za parametre ove mreže?**

A 5096 B 4640 C 8726 D 2856

16 (T) U nadziranom strojnom učenju, skup podataka često dijelimo u tri međusobno disjunktna podskupa: skup za učenje, skup za provjeru i skup za ispitivanje. Model učimo isključivo na skupu za učenje. **Čemu nam služe**

skup za provjeru i skup za ispitivanje?

A Na skupu za provjeru ispitujemo točnost modela, a na skupu za ispitivanje određujemo predikcijsku pogrešku modela najveće složenosti

B Skup za provjeru služi za učenje značajki modela, dok skup za ispitivanje služi za određivanje pogreške generalizacije modela različitih složenosti

C Na skupu za provjeru određujemo koliko model dobro generalizira, a na skupu za ispitivanje određujemo pogrešku modela na još neviđenim podatcima

D Na skupu za provjeru izračunavamo točnost modela različitih složenosti, a na skupu za ispitivanje vrednujemo točnost modela optimalne složenosti

17 (R) Zadan je skup primjeraka za učenje oblika $\{((x_{2,i}, x_{1,i}), y_i)\}$, koji sadrži pet primjeraka:

$$\{((3,2),1),((1,1),-1),((2,1),-1),((3,1),1),((1,2),-1)\}$$

Time je definirano preslikavanje $y_i = f(x_2, x_1)$. Ovo preslikavanje želimo naučiti TLU-perceptronom koji koristi prijenosnu funkciju skoka -1,1 (neka je izlaz -1 ako je $net < 0$, 1 inače). Učimo Rosenblattovim postupkom, pri čemu je iznos stope učenja jednak 1. Neka je početni vektor težina $[w_2, w_1, w_0] = [7, 3, -12]$. **Koliko će se puta provesti korigiranje vektora težina i koja je njegova konačna vrijednost?**

A 2 puta, $[7, 2, -12]$ B 3 puta, $[7, -1, -14]$ C 2 puta, $[6, 3, -15]$ D 3 puta, $[14, -2, -28]$

5. Prirodom inspirirani optimizacijski algoritmi (3 pitanja)

18 (R) Generacijskim genetskim algoritmom tražimo maksimum funkcije $f(x, y) = 15 - (x - 1)^2 - (y - 4)^2$. Kao reprezentaciju rješenja koristimo 6-bitovni kromosom, pri čemu se prva tri bita koriste za kodiranje vrijednosti varijable x, a preostala tri bita za kodiranje vrijednosti varijable y. Domena nad kojom se pretražuju vrijednosti od x je $[-1, 6]$ te od y je $[0, 7]$. Populacija se sastoji od četiri jedinke: J1=000110, J2=011001, J3=100111, J4=001110. Pretpostavite da se roditelji biraju proporcionalnom selekcijom te da su u jednom koraku kao roditelji izvučena dva rješenja koja imaju najmanju vjerojatnost odabira. Provedite nad njima postupak križanja s jednom točkom prijeloma (točka prijeloma je nakon prva dva bita); pretpostavite da u ovom koraku operator mutacije svaki puta djeluje na posljednja dva bita kromosoma. **Odredite iznos funkcije f u rješenju koje odgovara djetetu koje će biti poslano u sljedeću generaciju.** (Ako ih operatori križanja i mutacije generiraju više, u novu generaciju treba proslijediti najbolje.)

A 13 B 2 C 15 D 6

19 (P) Algoritmom kolonije mrava rješavamo problem trgovačkog putnika nad odabranim gradovima u Republici Hrvatskoj. Za algoritam su poznati sljedeći parametri: $\tau_0 = 100$, $\alpha = 2$, $\beta = 3$, $\rho = 0.1$, kolonija se sastoji od 100 mrava te ažuriranje u svakoj epohi obavlja samo mrav koji je u toj epohi pronašao najbolje rješenje. Razmotrimo brid koji povezuje gradove Zagreb i Dubrovnik. Pretpostavimo da niti u jednoj epohi ta dva grada nisu sljedbenici u pronađenom putu. **Nakon koliko će epoha algoritma količina feromonskih tragova na spomenutom bridu postati manja od 1?**

A 44 B 53 C 21 D 2

20 (T) Genetski algoritmi pripadaju grupi metaheurističkih stohastičkih optimizacijskih algoritama. **Što je kod njega nužno da bi algoritam konvergirao prema traženom rješenju?**

A Generacijska izvedba B Operator križanja C Velika populacija D Operator selekcije

Uvod u umjetnu inteligenciju – pismeni ispit (2020./2021.)

– NEKORIGIRANA VERZIJA –

*Ispit se sastoji od **20 pitanja** i ukupno nosi **20 bodova**. Točan odgovor nosi 1 bod a netočan -1/3 boda. Trajanje ispita je **150 minuta**. Primjerak ispita trebate predati zajedno sa svojim rješenjima.*

1. Pretraživanje prostora stanja i igranje igara (5 pitanja)

1 (P) Razmatramo problem slagalice dimenzija 3×3 s prosječnom dubinom rješenja $d = 11$. Za nalaženje rješenja algoritam pretraživanja u dubinu u prosjeku koristi 720 B memorije. **Koliko bi memorije u prosjeku koristio algoritam pretraživanja u širinu na ovome problemu?**

A 65 MB   B 720 MB   C 1.1 MB   D 5.4 MB

2 (R) Prostor stanja pretražujemo algoritmom A^*. Skup stanja je $S = \{a, b, c, d, e, f\}$, a funkcija sljedbenika je $succ(a) = \{(b, 2), (c, 2)\}$, $succ(b) = \{(c, 5), (d, 2)\}$, $succ(c) = \{(d, 1), (f, 20)\}$, $succ(d) = \{(e, 2)\}$, $succ(e) = \{(f, 14)\}$ te $succ(f) = \emptyset$. Heurističke vrijednosti stanja su $h(a) = 16$, $h(b) = 6$, $h(c) = 14$, $h(d) = 4$, $h(e) = 2$, $h(f) = 0$. Početno stanje je a, a ciljno f. Izvedite korake algoritma A^*, bilježeći u svakom koraku sadržaj liste otvorenih čvorova O i skupa zatvorenih čvorova C. U nultom koraku algoritma vrijedi $O = [(a, 0)]$ i $C = \emptyset$. **Koji je sadržaj listi O i skupa C nakon petog koraka izvođenja algoritma?**

A $O = [(d, 3), (f, 20)],\ C = \{(a, 0), (b, 2), (c, 2), (e, 6)\}$

B $O = [(d, 3), (f, 20)],\ C = \{(a, 0), (b, 2), (c, 2)\}$

C Algoritam ne dostiže peti korak

D $O = [(d, 3)],\ C = \{(a, 0), (b, 2), (c, 2), (e, 6)\}$

3 (T) Algoritam minimax koristi heurističku funkciju koja se poziva nad svim nezavršnim stanjima u stablu igre na određenoj dubini k. Igraći algoritmi općenito koriste različite heurističke funkcije. No, **što bi se dogodilo kada bi u igri sudjelovala dva algoritma koji imaju identične heurističke funkcije?**

A Ako je mehanizam odabira poteza kod izjednačavanja minimax-vrijednosti poznat i determinističan, ishod igre mogao bi se izračunati na početku igre

B Igrač koji je prvi na potezu bi pobijedio, osim ako u igri ima elemenata slučajnosti, i tada bi igra bila izjednačena

C Prvi igrač mogao bi unaprijed izračunati sve poteze protivnika do dubine k te tako odrediti globalno optimalnu minimax-strategiju

D Ništa se posebno ne bi dogodilo, jer i dalje oba algoritma ne bi mogla pretraživanjem u dubinu dosegnuti konačna stanja igre s definiranim isplatama

4 (T) Za razliku od slijepog pretraživanja prostora stanja, heurističko je pretraživanje usmjereno procjenom udaljenosti trenutačnog stanja do ciljnog stanja. **Koja je prednost heurističkog pretraživanja prostora stanja nad slijepim pretraživanjem?**

A Ima gornju ogradu na apriornu vremensku i prostornu složenost pretraživanja

B Ako je heuristika optimistična, osigurava potpunost, odnosno nalaženje ciljnog stanja ako ono postoji

C Smanjuje očekivan broj koraka izvođenja algoritma, a time i broj proširenih čvorova te zauzeće memorije

D Reducira apriornu vremensku složenost pretraživanja, uz uvjet da se koristi skup posjećenih stanja

5 (R) Stablo igre definirano je prijelazima $succ(A) = \{B, C\}, succ(B) = \{D, E\}, succ(D) = \{H, I\}, succ(E) = \{J, K, L\}, succ(C) = \{F, G\}, succ(F) = \{M, N, O\}, succ(G) = \{P, Q\}$. Heurističke vrijednosti listova su $h(H) = 10, h(I) = -h(O) = 3, h(J) = 13, h(K) = h(N) = -9, h(L) = -11, h(M) = 5, h(P) = 7, h(Q) = -16$. Optimalna strategija određuje se algoritmom minimax uz alfa-beta podrezivanje. **Koji će čvorovi pritom biti podrezani (preskočeni pri izračunu minimax vrijednosti)?**

A I, K, L, F, M, N, O, Q B K, L, G, P, Q C I, L D L, N, O, Q

2. Prikazivanje znanja, automatsko zaključivanje i logičko programiranje (5 pitanja)

6 (T) Rezolucija opovrgavanjem je ispravno i potpuno pravilo zaključivanja u FOL. **Što to znači?**

- A Postupak završava u konačnom broju koraka s odlukom je li formula logička posljedica premisa ili nije
- B Postupak izvodi klauzulu NIL ako i samo ako je negiran cilj logička posljedica premisa
- C Ako formula nije logička posljedica, onda to ne možemo dokazati rezolucijom opovrgavanjem
- D Kad god je skup premisa nekonzistentan, rezolucijsko pravilo izvodi NIL klauzulu

7 (P) Važno svojstvo pravila zaključivanje jest da je ono ispravno. **Koje je od sljedećih pravila zaključivanja ispravno?**

- A $B, A \to B \vdash A$
- C $\neg A, A \to B \vdash \neg B$
- B $A \to (B \lor C), \neg C \vdash A \to B$
- D $A \to B, B \to C \vdash C \to \neg A$

8 (T) Programi u Prologu sastoje se od definitnih klauzula, koje odgovaraju pravilima i činjenicama. **Što je razlika između pravila i činjenica u Prologu?**

- A Pravila imaju barem jedan negativan literal i točno jedan pozitivan literal, dok su činjenice jedinične klauzule sa barem jednim negativnim i točno jednim pozitivnim literalom
- B Pravila imaju najviše jedan negativan literal i barem jedan pozitivan literal, dok činjenice odgovaraju pravilima s uvijek lažnim antecedentom
- C Pravila imaju barem jedan negativan literal i točno jedan pozitivan literal, dok činjenice imaju jedan ili više pozitivnih literala
- D Pravila imaju barem jedan negativan literal i točno jedan pozitivan literal, dok činjenice odgovaraju pravilima s uvijek istinitim antecedentom

9 (R) Zadane su sljedeće dvije premise:

$$\exists x\Big(\exists y R(y) \lor \forall z \neg Q(x,z)\Big) \to P(a),\ \exists x \forall y\Big(Q(x,y) \to R(y)\Big)$$

Koju je od sljedećih formula moguće dokazati iz ovih premisa rezolucijom opovrgavanjem?

- A $\exists x(P(x) \lor R(x))$
- B $\forall x Q(a,x)$
- C $\exists x P(x) \land P(b)$
- D $\exists x \neg P(x)$

10 (P) Za automatsko zaključivanje u PL koristimo rezoluciju opovrgavanjem uz strategiju skupa potpore (SOS). Na takav postupak zaključivanja možemo gledati kao na problem pretraživanje prostora stanja, gdje stanja odgovaraju skupu klauzula (onih zadanih i onih izvedenih), a prijelazi između stanja odgovaraju primjeni rezolucijskog pravila na jedan par klauzula. Takav problem pretraživanje ima i svoj faktor grananja, koji ovisi o dubini stabla, tj. o koraku zaključivanja. Neka skup premisa sadrži 10 klauzula, a negirani cilj 5 klauzula. **Koliko iznosi gornja ograda na faktor grananja u drugom koraku zaključivanja?** (Napomena: Jednom razriješeni par klauzula više se ne razrješava.)

- A 60
- B 119
- C 104
- D 74

3. Modeliranje neizvjesnosti (2 pitanja)

11 (R) Bayesovom shemom zaključujemo o vjerojatnosti da neka država provodi nuklearne pokuse. Apriornu vjerojatnost da država provodi nuklearne pokuse, $P(N)$, izračunavamo kao procjenu najveće izglednosti (MLE) na temelju informacije da je za devet od 195 država poznato da imaju nuklearno oružje te da svaka treća od njih aktivno provodi nuklearne pokuse. Kao dokaze provođenja nuklearnog pokusa koristimo podatak o detekciji snažne sezmičke aktivnosti (S) te podatak o intenzivnoj proizvodnji obogaćenog uranija (U). Znamo da je vjerojatnost detekcije seizmičke aktivnosti uslijed nuklearnog pokusa jednaka 0.6, a vjerojatnost sezmičke aktivnosti bez nuklearnog pokusa svega 0.002. Također znamo je vjerojatnost proizvodnje obogaćenog uranija ako je proveden nuklearni pokus 0.3, dok bez provođenja nuklearnog pokusa ta vjerojatnost iznosi 0.2. Izračunajte aposteriornu vjerojatnost provođenja nuklearnog pokusa, i to prvo samo uz dokaz U, a zatim uz dodatni dokaz S, pretpostavljajući pritom uvjetnu nezavisnost tih dvaju dokaza. **Koliko iznosi porast aposteriorne vjerojatnosti nakon dodavanja drugog dokaza?**

- A 0.897
- B 0.785
- C 0.853
- D 0.714

12 (P) Razmotrimo jezičnu varijablu *starost kanarinca*, definiranu s izrazima *mlad* i *star*. Značenje tih izraza modeliramo neizrazitim skupovima M odnosno S, definiranima nad univerzalnim skupom $\mathbb{R}^+$, koji odgovara godinama. Funkcije pripadnosti μ_M i μ_S definiramo kao po dijelovima linearne funkcije. Funkcija $\mu_M(x)$ ima vrijednost 1 za $0 \leq x \leq 5$, vrijednost 0 za $x \geq 9$, te linearno pada za $5 < x < 9$. Funkcija $\mu_S(x)$ ima vrijednost 0 za $0 \leq x \leq 6$, vrijednost 1 za $x \geq 12$, te linearno raste za $6 < x < 12$. Uporabom Zadehovih operatora definiramo dva neizrazita skupa: skup X_1 sa značenjem *mlad ili star kanarinac* te skup X_2 sa značenjem *mlad ili ne mlad kanarinac*. **Za koje se sve elemente $x \in \mathbb{R}^+$ pripadnost ovim dvama neizrazitim skupovima razlikuje?**

A $7 \leq x \leq 12$ B $9 \leq x \leq 12$ C $7 \leq x \leq 9$ D $5 \leq x \leq 6$

4. Strojno učenje i umjetne neuronske mreže (5 pitanja)

13 (R) Zadan je skup primjeraka za učenje oblika $\{((x_{2,i}, x_{1,i}), y_i)\}$, koji sadrži pet primjeraka:

$$\{((3,2),1),((1,1),-1),((2,1),-1),((3,1),1),((1,2),-1)\}$$

Time je definirano preslikavanje $y_i = f(x_2, x_1)$. Ovo preslikavanje želimo naučiti TLU-perceptronom koji koristi prijenosnu funkciju skoka -1,1 (neka je izlaz -1 ako je *net* < 0, 1 inače). Učimo Rosenblattovim postupkom, pri čemu je iznos stope učenja jednak 0.5. Neka je početni vektor težina $[w_2, w_1, w_0] = [7, 3, -12]$. **Koliko će se puta provesti korigiranje vektora težina i koja je njegova konačna vrijednost?**

A 1 puta, $[10, 4, -26]$ B 1 puta, $[5, 2, -13]$ C 2 puta, $[7, 2, -12]$ D 2 puta, $[8, -1, -11]$

14 (T) U nadziranom strojnom učenju, skup podataka često dijelimo u tri međusobno disjunktna podskupa: skup za učenje, skup za provjeru i skup za ispitivanje. Model učimo isključivo na skupu za učenje. **Čemu nam služe skup za provjeru i skup za ispitivanje?**

A Na skupu za provjeru ispitujemo točnost modela, a na skupu za ispitivanje određujemo predikcijsku pogrešku modela najveće složenosti

B Na skupu za provjeru određujemo koliko model dobro generalizira, a na skupu za ispitivanje određujemo pogrešku modela na još neviđenim podatcima

C Skup za provjeru služi za učenje značajki modela, dok skup za ispitivanje služi za određivanje pogreške generalizacije modela različitih složenosti

D Na skupu za provjeru izračunavamo točnost modela različitih složenosti, a na skupu za ispitivanje vrednujemo točnost modela optimalne složenosti

15 (T) Stabla odluke i Bayesov klasifikator dva su algoritma nadziranog učenja. Međutim, ti se algoritmi vrlo razlikuju. **Što je prednost, a što nedostatak stabla odluke u odnosu na Bayesov klasifikator?**

A Prednost je što primjere možemo klasificirati u više od jedne klase, a nedostatak što se stablo odluke može lako prenaučiti

B Prednost je što su stabla odluke otporna na male promjene u ulaznom skupu podataka, a nedostatak što pretpostavljamo uvjetnu nezavisnost značajki

C Prednost je što stablo odluke možemo podrezati kako bismo spriječili prenaučenost, a nedostatak što može doći do podljeva pri računanju vjerojatnosti

D Prednost je što možemo bolje objasniti zašto je primjer klasificiran u neku klasu, a nedostatak što nemamo vjerojatnost klasifikacijske odluke

16 (P) Na računalu implementiramo unaprijednu potpuno povezanu slojevitu umjetnu neuronsku mrežu arhitekture $3 \times 20 \times 10 \times 5 \times 2$. Neuroni kao prijenosne funkcije koriste zglobnicu. U memoriji težine mreže čuvaju se kao tip `double` koji zauzima 8 okteta. **Koliki je ukupni utrošak memorije za parametre ove mreže?**

A 2560 B 4218 C 2856 D 5096

17 (R) Mali je Ivica svakog svakog ljeta u zadnjih sedam godina naučio jedan novi programski jezik. Svoja vrijedna iskustva sažeo je u listu *"Programski jezik koji mi se sviđa"*, gdje je svaki jezik opisao četirima značajkama, te je naznačio je li mu se dotični jezik svidio ($y = 1$) ili nije ($y = 0$). Ta lista izgleda ovako:

i	Evaluacija (E)	Izvođenje (I)	Paradigma (P)	Provjera tipova (T)	y
1	lijena	kompajler	imperativna	statička	0
2	striktna	interpreter	deklarativna	dinamička	0
3	lijena	kompajler	imperativna	dinamička	0
4	lijena	interpreter	hibridna	statička	0
5	striktna	interpreter	imperativna	statička	1
6	lijena	kompajler	hibridna	dinamička	1
7	striktna	kompajler	hibridna	dinamička	1

Ovog ljeta Mali Ivica želi puno jesti i spavati te opet naučiti novi programski jezik. Međutim, Mali Ivica ne želi izgubiti ljetno na programskom jeziku koji mu se ne sviđa, pa je odlučio izgraditi stablo odluke za klasifikaciju programskih jezika koje još nije naučio. Pomozite Malom Ivici te na gornjem skupu primjera primjenite algoritam ID3. U slučaju da dvije značajke ili više njih imaju jednaku vrijednost informacijske dobiti, prednost dajte značajki koje je u gornjoj tablici navedena prva (ljevija značajka). **Koje se značajke nalaze u drugoj razini stabla odluke dobivenog algoritmom ID3?**

A E, P, T B E, I, P C E, P D E, I

5. Prirodom inspirirani optimizacijski algoritmi (3 pitanja)

18 (P) Algoritmom kolonije mrava rješavamo problem trgovačkog putnika nad odabranim gradovima u Republici Hrvatskoj. Za algoritam su poznati sljedeći parametri: $\tau_0 = 100$, $\alpha = 2$, $\beta = 3$, $\rho = 0.1$, kolonija se sastoji od 100 mrava te ažuriranje u svakoj epohi obavlja samo mrav koji je u toj epohi pronašao najbolje rješenje. Razmotrimo brid koji povezuje gradove Zagreb i Dubrovnik. Pretpostavimo da niti u jednoj epohi ta dva grada nisu sljedbenici u pronađenom putu. **Nakon koliko će epoha algoritma količina feromonskih tragova na spomenutom bridu postati manja od 1?**

A 21 B 44 C 2 D 53

19 (T) Genetski algoritmi pripadaju grupi metaheurističkih stohastičkih optimizacijskih algoritama. **Što je kod njega nužno da bi algoritam konvergirao prema traženom rješenju?**

A Operator selekcije B Velika populacija C Operator križanja D Generacijska izvedba

20 (R) Generacijskim genetskim algoritmom tražimo maksimum funkcije $f(x, y) = 20 - (x-3)^2 - (y-4)^2$. Kao reprezentaciju rješenja koristimo 6-bitovni kromosom, pri čemu se prva tri bita koriste za kodiranje vrijednosti varijable x, a preostala tri bita za kodiranje vrijednosti varijable y. Domena nad kojom se pretražuju vrijednosti od x je $[1, 8]$ te od y je $[0, 7]$. Populacija se sastoji od četiri jedinke: J1=000110, J2=011001, J3=100111, J4=001110. Pretpostavite da se roditelji biraju proporcionalnom selekcijom te da su u jednom koraku kao roditelji izvučena dva rješenja koja imaju najmanju vjerojatnost odabira. Provedite nad njima postupak križanja s jednom točkom prijeloma (točka prijeloma je nakon prva dva bita); pretpostavite da u ovom koraku operator mutacije svaki puta djeluje na posljednja dva bita kromosoma. **Odredite iznos funkcije f u rješenju koje odgovara djetetu koje će biti poslano u sljedeću generaciju.** (Ako ih operatori križanja i mutacije generiraju više, u novu generaciju treba proslijediti najbolje.)

A 18 B 20 C 7 D 11

Uvod u umjetnu inteligenciju – pismeni ispit (2020./2021.)

– NEKORIGIRANA VERZIJA –

*Ispit se sastoji od **20 pitanja** i ukupno nosi **20 bodova**. Točan odgovor nosi 1 bod a netočan -1/3 boda. Trajanje ispita je **150 minuta**. Primjerak ispita trebate predati zajedno sa svojim rješenjima.*

1. Pretraživanje prostora stanja i igranje igara (5 pitanja)

1 (R) Prostor stanja pretražujemo algoritmom A^*. Skup stanja je $S = \{a, b, c, d, e, f\}$, a funkcija sljedbenika je $succ(a) = \{(b, 2), (c, 2)\}$, $succ(b) = \{(c, 5), (d, 2)\}$, $succ(c) = \{(d, 1), (f, 20)\}$, $succ(d) = \{(e, 2)\}$, $succ(e) = \{(f, 14)\}$ te $succ(f) = \emptyset$. Heurističke vrijednosti stanja su $h(a) = 16$, $h(b) = 6$, $h(c) = 14$, $h(d) = 4$, $h(e) = 2$, $h(f) = 0$. Početno stanje je a, a ciljno f. Izvedite korake algoritma A^*, bilježeći u svakom koraku sadržaj liste otvorenih čvorova O i skupa zatvorenih čvorova C. U nultom koraku algoritma vrijedi $O = [(a, 0)]$ i $C = \emptyset$. **Koji je sadržaj liste O i skupa C nakon šestog koraka izvođenja algoritma?**

A $O = [(e, 5), (f, 22)]$, $C = \{(a, 0), (b, 2), (c, 7), (d, 3)\}$

B Algoritam ne dostiže šesti korak

C $O = []$, $C = \{(a, 0), (b, 2), (c, 2), (d, 3), (e, 5)\}$

D $O = [(e, 5), (f, 20)]$, $C = \{(a, 0), (b, 2), (c, 2), (d, 3)\}$

2 (P) Razmatramo problem slagalice dimenzija 3×4 s prosječnom dubinom rješenja $d = 11$. Za nalaženje rješenja algoritam pretraživanja u dubinu u prosjeku koristi 720 B memorije. **Koliko bi memorije u prosjeku koristio algoritam pretraživanja u širinu na ovome problemu?**

A 2 MB   B 720 KB   C 87 MB   D 65.5 MB

3 (T) Algoritam minimax koristi heurističku funkciju koja se poziva nad svim nezavršnim stanjima u stablu igre na određenoj dubini k. Igraći algoritmi općenito koriste različite heurističke funkcije. No, **što bi se dogodilo kada bi u igri sudjelovala dva algoritma koji imaju identične heurističke funkcije?**

A Prvi igrač mogao bi unaprijed izračunati sve poteze protivnika do dubine k te tako odrediti globalno optimalnu minimax-strategiju

B Ništa se posebno ne bi dogodilo, jer i dalje oba algoritma ne bi mogla pretraživanjem u dubinu dosegnuti konačna stanja igre s definiranim isplatama

C Igrač koji je prvi na potezu bi pobijedio, osim ako u igri ima elemenata slučajnosti, i tada bi igra bila izjednačena

D Ako je mehanizam odabira poteza kod izjednačavanja minimax-vrijednosti poznat i determinističan, ishod igre mogao bi se izračunati na početku igre

4 (T) Za razliku od slijepog pretraživanja prostora stanja, heurističko je pretraživanje usmjereno procjenom udaljenosti trenutačnog stanja do ciljnog stanja. **Koja je prednost heurističkog pretraživanja prostora stanja nad slijepim pretraživanjem?**

A Ima gornju ogradu na apriornu vremensku i prostornu složenost pretraživanja

B Smanjuje očekivan broj koraka izvođenja algoritma, a time i broj proširenih čvorova te zauzeće memorije

C Reducira apriornu vremensku složenost pretraživanja, uz uvjet da se koristi skup posjećenih stanja

D Ako je heuristika optimistična, osigurava potpunost, odnosno nalaženje ciljnog stanja ako ono postoji

5 (R) Stablo igre definirano je prijelazima $succ(A) = \{B, C\}, succ(B) = \{D, E\}, succ(D) = \{H, I\}, succ(E) = \{J, K, L\}, succ(C) = \{F, G\}, succ(F) = \{M, N, O\}, succ(G) = \{P, Q\}$. Heurističke vrijednosti listova su $h(H) = 3, h(I) = -14, h(J) = -3, h(K) = h(Q) = -5, h(L) = 7, h(M) = -1, h(N) = -10, h(O) = -1, h(P) = 17$. Optimalna strategija određuje se algoritmom minimax uz alfa-beta podrezivanje. **Koji će čvorovi pritom biti podrezani (preskočeni pri izračunu minimax vrijednosti)?**

A G, P, Q B P, Q C Q D N, O, P, Q

2. Prikazivanje znanja, automatsko zaključivanje i logičko programiranje (5 pitanja)

6 (R) Zadane su sljedeće dvije premise:

$$\exists x\Big(\exists y R(y) \lor \forall z \neg Q(x,z)\Big) \to P(a),\ \exists x \forall y\Big(Q(x,y) \to R(y)\Big)$$

Koju je od sljedećih formula moguće dokazati iz ovih premisa rezolucijom opovrgavanjem?

A $\exists x R(x)$ B $\forall x P(x)$ C $\exists x P(x) \land P(b)$ D $\exists x P(x) \lor P(b)$

7 (P) Važno svojstvo pravila zaključivanje jest da je ono ispravno. **Koje je od sljedećih pravila zaključivanja ispravno?**

A $A \to B, B \to C \vdash C \to \neg A$

B $A \to (B \to C) \vdash (A \lor B) \to C$

C $A \to (B \to C) \vdash (A \land B) \to C$

D $\neg A, A \to B \vdash \neg B$

8 (T) Programi u Prologu sastoje se od definitnih klauzula, koje odgovaraju pravilima i činjenicama. **Što je razlika između pravila i činjenica u Prologu?**

A Pravila imaju barem jedan negativan literal i točno jedan pozitivan literal, dok su činjenice jedinične klauzule sa samo jednim pozitivnim literalom

B Pravila imaju antecedent s barem jednim pozitivnim literalom, dok su činjenice klauzule sa samo negativnim literalima

C Pravila imaju antecedent s točno jednim pozitivnim literalom, dok su činjenice jedinične klauzule sa samo jednim pozitivnim literalom

D Pravila imaju najviše jedan negativan literal i barem jedan pozitivan literal, dok činjenice odgovaraju pravilima s uvijek lažnim antecedentom

9 (T) Rezolucija opovrgavanjem je ispravno i potpuno pravilo zaključivanja u FOL. **Što to znači?**

A Kad god je skup premisa nekonzistentan, rezolucijsko pravilo izvodi NIL klauzulu

B Ako formula nije logička posljedica, onda to ne možemo dokazati rezolucijom opovrgavanjem

C Postupak izvodi klauzulu NIL ako i samo ako je negiran cilj logička posljedica premisa

D Postupak završava u konačnom broju koraka s odlukom je li formula logička posljedica premisa ili nije

10 (P) Za automatsko zaključivanje u PL koristimo rezoluciju opovrgavanjem uz strategiju skupa potpore (SOS). Na takav postupak zaključivanja možemo gledati kao na problem pretraživanje prostora stanja, gdje stanja odgovaraju skupu klauzula (onih zadanih i onih izvedenih), a prijelazi između stanja odgovaraju primjeni rezolucijskog pravila na jedan par klauzula. Takav problem pretraživanje ima i svoj faktor grananja, koji ovisi o dubini stabla, tj. o koraku zaključivanja. Neka skup premisa sadrži 10 klauzula, a negirani cilj 5 klauzula. **Koliko iznosi gornja ograda na faktor grananja u drugom koraku zaključivanja?** (Napomena: Jednom razriješeni par klauzula više se ne razrješava.)

A 119 B 74 C 104 D 60

3. Modeliranje neizvjesnosti (2 pitanja)

11 (R) Bayesovom shemom zaključujemo o vjerojatnosti da neka država provodi nuklearne pokuse. Apriornu vjerojatnost da država provodi nuklearne pokuse, $P(N)$, izračunavamo kao procjenu najveće izglednosti (MLE) na temelju informacije da je za devet od 195 država poznato da imaju nuklearno oružje te da svaka treća od njih aktivno provodi nuklearne pokuse. Kao dokaze provođenja nuklearnog pokusa koristimo podatak o detekciji snažne sezmičke aktivnosti (S) te podatak o intenzivnoj proizvodnji obogaćenog uranija (U). Znamo da je vjerojatnost detekcije seizmičke aktivnosti uslijed nuklearnog pokusa jednaka 0.8, a vjerojatnost sezmičke aktivnosti bez nuklearnog pokusa svega 0.002. Također znamo je vjerojatnost proizvodnje obogaćenog uranija ako je proveden nuklearni pokus 0.5, dok bez provođenja nuklearnog pokusa ta vjerojatnost iznosi 0.1. Izračunajte aposteriornu vjerojatnost provođenja nuklearnog pokusa, i to prvo samo uz dokaz U, a zatim uz dodatni dokaz S, pretpostavljajući pritom uvjetnu nezavisnost tih dvaju dokaza. **Koliko iznosi porast aposteriorne vjerojatnosti nakon dodavanja drugog dokaza?**

A 0.785 B 0.853 C 0.714 D 0.897

12 (P) Razmotrimo jezičnu varijablu *starost kanarinca*, definiranu s izrazima *mlad* i *star*. Značenje tih izraza modeliramo neizrazitim skupovima M odnosno S, definiranima nad univerzalnim skupom $\mathbb{R}^+$, koji odgovara godinama. Funkcije pripadnosti μ_M i μ_S definiramo kao po dijelovima linearne funkcije. Funkcija $\mu_M(x)$ ima vrijednost 1 za $0 \leq x \leq 5$, vrijednost 0 za $x \geq 9$, te linearno pada za $5 < x < 9$. Funkcija $\mu_S(x)$ ima vrijednost 0 za $0 \leq x \leq 6$, vrijednost 1 za $x \geq 12$, te linearno raste za $6 < x < 12$. Uporabom Zadehovih operatora definiramo dva neizrazita skupa: skup X_1 sa značenjem *mlad ili star kanarinac* te skup X_2 sa značenjem *mlad ili ne mlad kanarinac*. **Za koje se sve elemente $x \in \mathbb{R}^+$ pripadnost ovim dvama neizrazitim skupovima razlikuje?**

A $5 \leq x \leq 6$ B $7 \leq x \leq 9$ C $7 \leq x \leq 12$ D $9 \leq x \leq 12$

4. Strojno učenje i umjetne neuronske mreže (5 pitanja)

13 (R) Zadan je skup primjeraka za učenje oblika $\{((x_{2,i}, x_{1,i}), y_i)\}$, koji sadrži pet primjeraka:

$$\{((3,2),1),((1,1),-1),((2,1),-1),((3,1),1),((1,2),-1)\}$$

Time je definirano preslikavanje $y_i = f(x_2, x_1)$. Ovo preslikavanje želimo naučiti TLU-perceptronom koji koristi prijenosnu funkciju skoka -1,1 (neka je izlaz -1 ako je *net* < 0, 1 inače). Učimo Rosenblattovim postupkom, pri čemu je iznos stope učenja jednak 0.5. Neka je početni vektor težina $[w_2, w_1, w_0] = [7, 3, -12]$. **Koliko će se puta provesti korigiranje vektora težina i koja je njegova konačna vrijednost?**

A 1 puta, $[5, 2, -13]$ B 2 puta, $[7, 2, -12]$ C 2 puta, $[8, -1, -11]$ D 1 puta, $[10, 4, -26]$

14 (P) Na računalu implementiramo unaprijednu potpuno povezanu slojevitu umjetnu neuronsku mrežu arhitekture $3 \times 40 \times 10 \times 5 \times 2$. Neuroni kao prijenosne funkcije koriste zglobnicu. U memoriji težine mreže čuvaju se kao tip `double` koji zauzima 8 okteta. **Koliki je ukupni utrošak memorije za parametre ove mreže?**

A 4640 B 2856 C 5096 D 8726

15 (R) Mali je Ivica svakog svakog ljeta u zadnjih sedam godina naučio jedan novi programski jezik. Svoja vrijedna iskustva sažeo je u listu *"Programski jezik koji mi se sviđa"*, gdje je svaki jezik opisao četirima značajkama, te je naznačio je li mu se dotični jezik svidio ($y = 1$) ili nije ($y = 0$). Ta lista izgleda ovako:

i	Evaluacija (E)	Izvođenje (I)	Paradigma (P)	Provjera tipova (T)	y
1	lijena	kompajler	imperativna	statička	0
2	striktna	interpreter	deklarativna	dinamička	0
3	lijena	kompajler	imperativna	dinamička	0
4	lijena	interpreter	hibridna	statička	0
5	striktna	interpreter	imperativna	statička	1
6	lijena	kompajler	hibridna	dinamička	1
7	striktna	kompajler	hibridna	dinamička	1

Ovog ljeta Mali Ivica želi puno jesti i spavati te opet naučiti novi programski jezik. Međutim, Mali Ivica ne želi izgubiti ljetno na programskom jeziku koji mu se ne sviđa, pa je odlučio izgraditi stablo odluke za klasifikaciju programskih jezika koje još nije naučio. Pomozite Malom Ivici te na gornjem skupu primjera primjenite algoritam ID3. U slučaju da dvije značajke ili više njih imaju jednaku vrijednost informacijske dobiti, prednost dajte značajki koje je u gornjoj tablici navedena prva (ljevija značajka). **Koje se značajke nalaze u drugoj razini stabla odluke dobivenog algoritmom ID3?**

A E, I B E, P, T C E, I, P D E, P

16 (T) Stabla odluke i Bayesov klasifikator dva su algoritma nadziranog učenja. Međutim, ti se algoritmi vrlo razlikuju. **Što je prednost, a što nedostatak stabla odluke u odnosu na Bayesov klasifikator?**

A Prednost je što možemo bolje objasniti zašto je primjer klasificiran u neku klasu, a nedostatak što nemamo vjerojatnost klasifikacijske odluke

B Prednost je što su stabla odluke otporna na male promjene u ulaznom skupu podataka, a nedostatak što pretpostavljamo uvjetnu nezavisnost značajki

C Prednost je što primjere možemo klasificirati u više od jedne klase, a nedostatak što se stablo odluke može lako prenaučiti

D Prednost je što stablo odluke možemo podrezati kako bismo spriječili prenaučenost, a nedostatak što može doći do podljeva pri računanju vjerojatnosti

17 (T) U nadziranom strojnom učenju, skup podataka često dijelimo u tri međusobno disjunktna podskupa: skup za učenje, skup za provjeru i skup za ispitivanje. Model učimo isključivo na skupu za učenje. **Čemu nam služe skup za provjeru i skup za ispitivanje?**

- A Na skupu za provjeru izračunavamo točnost modela različitih složenosti, a na skupu za ispitivanje vrednujemo točnost modela optimalne složenosti
- B Na skupu za provjeru određujemo koliko model dobro generalizira, a na skupu za ispitivanje određujemo pogrešku modela na još neviđenim podatcima
- C Na skupu za provjeru ispitujemo točnost modela, a na skupu za ispitivanje određujemo predikcijsku pogrešku modela najveće složenosti
- D Skup za provjeru služi za učenje značajki modela, dok skup za ispitivanje služi za određivanje pogreške generalizacije modela različitih složenosti

5. Prirodom inspirirani optimizacijski algoritmi (3 pitanja)

18 (R) Generacijskim genetskim algoritmom tražimo maksimum funkcije $f(x, y) = 18 - (x - 4)^2 - (y - 2)^2$. Kao reprezentaciju rješenja koristimo 6-bitovni kromosom, pri čemu se prva tri bita koriste za kodiranje vrijednosti varijable x, a preostala tri bita za kodiranje vrijednosti varijable y. Domena nad kojom se pretražuju vrijednosti od x je $[2, 9]$ te od y je $[-2, 5]$. Populacija se sastoji od četiri jedinke: J1=000110, J2=011001, J3=100111, J4=001110. Pretpostavite da se roditelji biraju proporcionalnom selekcijom te da su u jednom koraku kao roditelji izvučena dva rješenja koja imaju najmanju vjerojatnost odabira. Provedite nad njima postupak križanja s jednom točkom prijeloma (točka prijeloma je nakon prva dva bita); pretpostavite da u ovom koraku operator mutacije svaki puta djeluje na posljednja dva bita kromosoma. **Odredite iznos funkcije f u rješenju koje odgovara djetetu koje će biti poslano u sljedeću generaciju.** (Ako ih operatori križanja i mutacije generiraju više, u novu generaciju treba proslijediti najbolje.)

A 9 B 5 C 18 D 16

19 (T) Genetski algoritmi pripadaju grupi metaheurističkih stohastičkih optimizacijskih algoritama. **Što je kod njega nužno da bi algoritam konvergirao prema traženom rješenju?**

A Operator selekcije B Generacijska izvedba C Velika populacija D Operator križanja

20 (P) Algoritmom kolonije mrava rješavamo problem trgovačkog putnika nad odabranim gradovima u Republici Hrvatskoj. Za algoritam su poznati sljedeći parametri: $\tau_0 = 100$, $\alpha = 2$, $\beta = 3$, $\rho = 0.1$, kolonija se sastoji od 100 mrava te ažuriranje u svakoj epohi obavlja samo mrav koji je u toj epohi pronašao najbolje rješenje. Razmotrimo brid koji povezuje gradove Zagreb i Dubrovnik. Pretpostavimo da niti u jednoj epohi ta dva grada nisu sljedbenici u pronađenom putu. **Nakon koliko će epoha algoritma količina feromonskih tragova na spomenutom bridu postati manja od 1?**

A 21 B 2 C 44 D 53

Uvod u umjetnu inteligenciju – pismeni ispit (2020./2021.)

– NEKORIGIRANA VERZIJA –

*Ispit se sastoji od **20 pitanja** i ukupno nosi **20 bodova**. Točan odgovor nosi 1 bod a netočan -1/3 boda. Trajanje ispita je **150 minuta**. Primjerak ispita trebate predati zajedno sa svojim rješenjima.*

1. Pretraživanje prostora stanja i igranje igara (5 pitanja)

1 (R) Prostor stanja pretražujemo algoritmom A^*. Skup stanja je $S = \{a, b, c, d, e, f\}$, a funkcija sljedbenika je $succ(a) = \{(b, 2), (c, 2)\}$, $succ(b) = \{(c, 5), (d, 2)\}$, $succ(c) = \{(d, 1), (f, 20)\}$, $succ(d) = \{(e, 2)\}$, $succ(e) = \{(f, 14)\}$ te $succ(f) = \emptyset$. Heurističke vrijednosti stanja su $h(a) = 16$, $h(b) = 6$, $h(c) = 14$, $h(d) = 4$, $h(e) = 2$, $h(f) = 0$. Početno stanje je a, a ciljno f. Izvedite korake algoritma A^*, bilježeći u svakom koraku sadržaj liste otvorenih čvorova O i skupa zatvorenih čvorova C. U nultom koraku algoritma vrijedi $O = [(a, 0)]$ i $C = \emptyset$. **Koji je sadržaj listi O i skupa C nakon petog koraka izvođenja algoritma?**

A $O = [(d, 3), (f, 22)]$, $C = \{(a, 0), (b, 2), (c, 7), (e, 6)\}$

B $O = [(d, 3), (f, 20)]$, $C = \{(a, 0), (b, 2), (c, 2), (e, 6)\}$

C $O = [(d, 3), (f, 20)]$, $C = \{(a, 0), (b, 2), (c, 2)\}$

D Algoritam ne dostiže peti korak

2 (R) Stablo igre definirano je prijelazima $succ(A) = \{B, C\}, succ(B) = \{D, E\}, succ(D) = \{H, I\}, succ(E) = \{J, K, L\}, succ(C) = \{F, G\}, succ(F) = \{M, N, O\}, succ(G) = \{P, Q\}$. Heurističke vrijednosti listova su $h(H) = h(J) = 8, h(I) = -15, h(K) = -h(Q) = 17, h(L) = h(O) = 1, h(M) = -12, h(N) = -19, h(P) = -14$. Optimalna strategija određuje se algoritmom minimax uz alfa-beta podrezivanje. **Koji će čvorovi pritom biti podrezani (preskočeni pri izračunu minimax vrijednosti)?**

A K, L, G, P, Q B L, F, M, N, O C L, N, O, Q D L, Q

3 (P) Razmatramo problem slagalice dimenzija 3×3 s prosječnom dubinom rješenja $d = 11$. Za nalaženje rješenja algoritam pretraživanja u dubinu u prosjeku koristi 650 B memorije. **Koliko bi memorije u prosjeku koristio algoritam pretraživanja u širinu na ovome problemu?**

A 1.6 MB B 60 MB C 75.7 KB D 1 MB

4 (T) Za razliku od slijepog pretraživanja prostora stanja, heurističko je pretraživanje usmjereno procjenom udaljenosti trenutačnog stanja do ciljnog stanja. **Koja je prednost heurističkog pretraživanja prostora stanja nad slijepim pretraživanjem?**

A Reducira apriornu vremensku složenost pretraživanja, uz uvjet da se koristi skup posjećenih stanja

B Smanjuje očekivan broj koraka izvođenja algoritma, a time i broj proširenih čvorova te zauzeće memorije

C Ako je heuristika optimistična, osigurava potpunost, odnosno nalaženje ciljnog stanja ako ono postoji

D Ima gornju ogradu na apriornu vremensku i prostornu složenost pretraživanja

5 (T) Algoritam minimax koristi heurističku funkciju koja se poziva nad svim nezavršnim stanjima u stablu igre na određenoj dubini k. Igraći algoritmi općenito koriste različite heurističke funkcije. No, **što bi se dogodilo kada bi u igri sudjelovala dva algoritma koji imaju identične heurističke funkcije?**

A Prvi igrač mogao bi unaprijed izračunati sve poteze protivnika do dubine k te tako odrediti globalno optimalnu minimax-strategiju

B Ako je mehanizam odabira poteza kod izjednačavanja minimax-vrijednosti poznat i determinističan, ishod igre mogao bi se izračunati na početku igre

C Igrač koji je prvi na potezu bi pobijedio, osim ako u igri ima elemenata slučajnosti, i tada bi igra bila izjednačena

D Ništa se posebno ne bi dogodilo, jer i dalje oba algoritma ne bi mogla pretraživanjem u dubinu dosegnuti konačna stanja igre s definiranim isplatama

2. Prikazivanje znanja, automatsko zaključivanje i logičko programiranje (5 pitanja)

6 (P) Važno svojstvo pravila zaključivanje jest da je ono ispravno. **Koje je od sljedećih pravila zaključivanja ispravno?**

A $A \vdash A \wedge B$
C $A \rightarrow (B \vee C), \neg C \vdash A \rightarrow B$
B $A \rightarrow (B \rightarrow C) \vdash (A \vee B) \rightarrow C$
D $\neg A, A \rightarrow B \vdash \neg B$

7 (P) Za automatsko zaključivanje u PL koristimo rezoluciju opovrgavanjem uz strategiju skupa potpore (SOS). Na takav postupak zaključivanja možemo gledati kao na problem pretraživanje prostora stanja, gdje stanja odgovaraju skupu klauzula (onih zadanih i onih izvedenih), a prijelazi između stanja odgovaraju primjeni rezolucijskog pravila na jedan par klauzula. Takav problem pretraživanje ima i svoj faktor grananja, koji ovisi o dubini stabla, tj. o koraku zaključivanja. Neka skup premisa sadrži 10 klauzula, a negirani cilj 5 klauzula. **Koliko iznosi gornja ograda na faktor grananja u drugom koraku zaključivanja?** (Napomena: Jednom razriješeni par klauzula više se ne razrješava.)

A 119 B 60 C 74 D 104

8 (T) Rezolucija opovrgavanjem je ispravno i potpuno pravilo zaključivanja u FOL. **Što to znači?**

A Postupak završava u konačnom broju koraka s odlukom je li formula logička posljedica premisa ili nije
B Postupak izvodi klauzulu NIL ako i samo ako je negiran cilj logička posljedica premisa
C Kad god je skup premisa nekonzistentan, rezolucijsko pravilo izvodi NIL klauzulu
D Ako formula nije logička posljedica, onda to ne možemo dokazati rezolucijom opovrgavanjem

9 (T) Programi u Prologu sastoje se od definitnih klauzula, koje odgovaraju pravilima i činjenicama. **Što je razlika između pravila i činjenica u Prologu?**

A Pravila imaju antecedent s točno jednim pozitivnim literalom, dok su činjenice jedinične klauzule sa samo jednim pozitivnim literalom
B Pravila imaju antecedent s barem jednim pozitivnim literalom, dok su činjenice jedinične klauzule sa samo jednim pozitivnim literalom
C Pravila imaju najviše jedan negativan literal i barem jedan pozitivan literal, dok su činjenice jedinične klauzule sa samo jednim pozitivnim literalom
D Pravila imaju antecedent s najviše jednim pozitivnim literalom, dok činjenice odgovaraju pravilima s uvijek lažnim antecedentom

10 (R) Zadane su sljedeće dvije premise:

$$\exists x\Big(\exists y R(y) \vee \forall z \neg Q(x,z)\Big) \rightarrow P(a),\ \exists x \forall y\Big(Q(x,y) \rightarrow R(y)\Big)$$

Koju je od sljedećih formula moguće dokazati iz ovih premisa rezolucijom opovrgavanjem?

A $\neg\exists x P(x)$ B $\forall x P(x)$ C $Q(a,b)$ D $\forall x(\exists y P(y) \vee R(x))$

3. Modeliranje neizvjesnosti (2 pitanja)

11 (R) Bayesovom shemom zaključujemo o vjerojatnosti da neka država provodi nuklearne pokuse. Apriornu vjerojatnost da država provodi nuklearne pokuse, $P(N)$, izračunavamo kao procjenu najveće izglednosti (MLE) na temelju informacije da je za devet od 195 država poznato da imaju nuklearno oružje te da svaka treća od njih aktivno provodi nuklearne pokuse. Kao dokaze provođenja nuklearnog pokusa koristimo podatak o detekciji snažne sezmičke aktivnosti (S) te podatak o intenzivnoj proizvodnji obogaćenog uranija (U). Znamo da je vjerojatnost detekcije seizmičke aktivnosti uslijed nuklearnog pokusa jednaka 0.8, a vjerojatnost sezmičke aktivnosti bez nuklearnog pokusa svega 0.002. Također znamo je vjerojatnost proizvodnje obogaćenog uranija ako je proveden nuklearni pokus 0.5, dok bez provođenja nuklearnog pokusa ta vjerojatnost iznosi 0.1. Izračunajte aposteriornu vjerojatnost provođenja nuklearnog pokusa, i to prvo samo uz dokaz U, a zatim uz dodatni dokaz S, pretpostavljajući pritom uvjetnu nezavisnost tih dvaju dokaza. **Koliko iznosi porast aposteriorne vjerojatnosti nakon dodavanja drugog dokaza?**

A 0.714 B 0.785 C 0.897 D 0.853

12 (P) Razmotrimo jezičnu varijablu *starost kanarinca*, definiranu s izrazima *mlad* i *star*. Značenje tih izraza modeliramo neizrazitim skupovima M odnosno S, definiranima nad univerzalnim skupom $\mathbb{R}^+$, koji odgovara godinama. Funkcije pripadnosti μ_M i μ_S definiramo kao po dijelovima linearne funkcije. Funkcija $\mu_M(x)$ ima vrijednost 1 za $0 \leq x \leq 5$, vrijednost 0 za $x \geq 9$, te linearno pada za $5 < x < 9$. Funkcija $\mu_S(x)$ ima vrijednost 0 za $0 \leq x \leq 6$, vrijednost 1 za $x \geq 12$, te linearno raste za $6 < x < 12$. Uporabom Zadehovih operatora definiramo dva neizrazita skupa: skup X_1 sa značenjem *mlad ili star kanarinac* te skup X_2 sa značenjem *mlad ili ne mlad kanarinac*. **Za koje se sve elemente $x \in \mathbb{R}^+$ pripadnost ovim dvama neizrazitim skupovima razlikuje?**

A $7 \leq x \leq 9$ B $7 \leq x \leq 12$ C $9 \leq x \leq 12$ D $5 \leq x \leq 6$

4. Strojno učenje i umjetne neuronske mreže (5 pitanja)

13 (T) U nadziranom strojnom učenju, skup podataka često dijelimo u tri međusobno disjunktna podskupa: skup za učenje, skup za provjeru i skup za ispitivanje. Model učimo isključivo na skupu za učenje. **Čemu nam služe skup za provjeru i skup za ispitivanje?**

A Skup za provjeru služi za učenje značajki modela, dok skup za ispitivanje služi za određivanje pogreške generalizacije modela različitih složenosti

B Na skupu za provjeru određujemo koliko model dobro generalizira, a na skupu za ispitivanje određujemo pogrešku modela na još neviđenim podatcima

C Na skupu za provjeru izračunavamo točnost modela različitih složenosti, a na skupu za ispitivanje vrednujemo točnost modela optimalne složenosti

D Na skupu za provjeru ispitujemo točnost modela, a na skupu za ispitivanje određujemo predikcijsku pogrešku modela najveće složenosti

14 (R) Mali je Ivica svakog svakog ljeta u zadnjih sedam godina naučio jedan novi programski jezik. Svoja vrijedna iskustva sažeo je u listu *"Programski jezik koji mi se sviđa"*, gdje je svaki jezik opisao četirima značajkama, te je naznačio je li mu se dotični jezik svidio ($y = 1$) ili nije ($y = 0$). Ta lista izgleda ovako:

i	Evaluacija (E)	Izvođenje (I)	Paradigma (P)	Provjera tipova (T)	y
1	lijena	kompajler	imperativna	statička	0
2	striktna	interpreter	deklarativna	dinamička	0
3	lijena	kompajler	imperativna	dinamička	0
4	lijena	interpreter	hibridna	statička	0
5	striktna	interpreter	imperativna	statička	1
6	lijena	kompajler	hibridna	dinamička	1
7	striktna	kompajler	hibridna	dinamička	1

Ovog ljeta Mali Ivica želi puno jesti i spavati te opet naučiti novi programski jezik. Međutim, Mali Ivica ne želi izgubiti ljetno na programskom jeziku koji mu se ne sviđa, pa je odlučio izgraditi stablo odluke za klasifikaciju programskih jezika koje još nije naučio. Pomozite Malom Ivici te na gornjem skupu primjera primjenite algoritam ID3. U slučaju da dvije značajke ili više njih imaju jednaku vrijednost informacijske dobiti, prednost dajte značajki koje je u gornjoj tablici navedena prva (ljevija značajka). **Koje se značajke nalaze u drugoj razini stabla odluke dobivenog algoritmom ID3?**

A E, I, P B E, I C E, P, T D E, P

15 (T) Stabla odluke i Bayesov klasifikator dva su algoritma nadziranog učenja. Međutim, ti se algoritmi vrlo razlikuju. **Što je prednost, a što nedostatak stabla odluke u odnosu na Bayesov klasifikator?**

A Prednost je što primjere možemo klasificirati u više od jedne klase, a nedostatak što se stablo odluke može lako prenaučiti

B Prednost je što stablo odluke možemo podrezati kako bismo spriječili prenaučenost, a nedostatak što može doći do podljeva pri računanju vjerojatnosti

C Prednost je što su stabla odluke otporna na male promjene u ulaznom skupu podataka, a nedostatak što pretpostavljamo uvjetnu nezavisnost značajki

D Prednost je što možemo bolje objasniti zašto je primjer klasificiran u neku klasu, a nedostatak što nemamo vjerojatnost klasifikacijske odluke

16 (P) Na računalu implementiramo unaprijednu potpuno povezanu slojevitu umjetnu neuronsku mrežu arhitekture $3 \times 20 \times 10 \times 5 \times 2$. Neuroni kao prijenosne funkcije koriste zglobnicu. U memoriji težine mreže čuvaju se kao tip `double` koji zauzima 8 okteta. **Koliki je ukupni utrošak memorije za parametre ove mreže?**

A 2560 B 4218 C 5096 D 2856

17 (R) Zadan je skup primjeraka za učenje oblika $\{((x_{2,i}, x_{1,i}), y_i)\}$, koji sadrži pet primjeraka:

$$\{((3,2),1),((1,1),-1),((2,1),-1),((3,1),1),((1,2),-1)\}$$

Time je definirano preslikavanje $y_i = f(x_2, x_1)$. Ovo preslikavanje želimo naučiti TLU-perceptronom koji koristi prijenosnu funkciju skoka -1,1 (neka je izlaz -1 ako je $net < 0$, 1 inače). Učimo Rosenblattovim postupkom, pri čemu je iznos stope učenja jednak 1. Neka je početni vektor težina $[w_2, w_1, w_0] = [7, 3, -12]$. **Koliko će se puta provesti korigiranje vektora težina i koja je njegova konačna vrijednost?**

A 2 puta, $[7, 2, -12]$ B 3 puta, $[14, -2, -28]$ C 3 puta, $[7, -1, -14]$ D 2 puta, $[6, 3, -15]$

5. Prirodom inspirirani optimizacijski algoritmi (3 pitanja)

18 (R) Generacijskim genetskim algoritmom tražimo maksimum funkcije $f(x, y) = 20 - (x - 3)^2 - (y - 4)^2$. Kao reprezentaciju rješenja koristimo 6-bitovni kromosom, pri čemu se prva tri bita koriste za kodiranje vrijednosti varijable x, a preostala tri bita za kodiranje vrijednosti varijable y. Domena nad kojom se pretražuju vrijednosti od x je $[1, 8]$ te od y je $[0, 7]$. Populacija se sastoji od četiri jedinke: J1=000110, J2=011001, J3=100111, J4=001110. Pretpostavite da se roditelji biraju proporcionalnom selekcijom te da su u jednom koraku kao roditelji izvučena dva rješenja koja imaju najmanju vjerojatnost odabira. Provedite nad njima postupak križanja s jednom točkom prijeloma (točka prijeloma je nakon prva dva bita); pretpostavite da u ovom koraku operator mutacije svaki puta djeluje na posljednja dva bita kromosoma. **Odredite iznos funkcije f u rješenju koje odgovara djetetu koje će biti poslano u sljedeću generaciju.** (Ako ih operatori križanja i mutacije generiraju više, u novu generaciju treba proslijediti najbolje.)

A 20 B 11 C 18 D 7

19 (P) Algoritmom kolonije mrava rješavamo problem trgovačkog putnika nad odabranim gradovima u Republici Hrvatskoj. Za algoritam su poznati sljedeći parametri: $\tau_0 = 20$, $\alpha = 2$, $\beta = 3$, $\rho = 0.2$, kolonija se sastoji od 40 mrava te ažuriranje u svakoj epohi obavlja samo mrav koji je u toj epohi pronašao najbolje rješenje. Razmotrimo brid koji povezuje gradove Zagreb i Dubrovnik. Pretpostavimo da niti u jednoj epohi ta dva grada nisu sljedbenici u pronađenom putu. **Nakon koliko će epoha algoritma količina feromonskih tragova na spomenutom bridu postati manja od 0.1?**

A 52 B 31 C 24 D 4

20 (T) Genetski algoritmi pripadaju grupi metaheurističkih stohastičkih optimizacijskih algoritama. **Što je kod njega nužno da bi algoritam konvergirao prema traženom rješenju?**

A Velika populacija B Operator selekcije C Operator križanja D Generacijska izvedba

Uvod u umjetnu inteligenciju – pismeni ispit (2020./2021.)

– NEKORIGIRANA VERZIJA –

*Ispit se sastoji od **20 pitanja** i ukupno nosi **20 bodova**. Točan odgovor nosi 1 bod a netočan -1/3 boda. Trajanje ispita je **150 minuta**. Primjerak ispita trebate predati zajedno sa svojim rješenjima.*

1. Pretraživanje prostora stanja i igranje igara (5 pitanja)

1 (R) Prostor stanja pretražujemo algoritmom A^*. Skup stanja je $S = \{a, b, c, d, e, f\}$, a funkcija sljedbenika je $succ(a) = \{(b, 2),(c, 2)\}$, $succ(b) = \{(c, 5),(d, 2)\}$, $succ(c) = \{(d, 1),(f, 20)\}$, $succ(d) = \{(e, 2)\}$, $succ(e) = \{(f, 14)\}$ te $succ(f) = \emptyset$. Heurističke vrijednosti stanja su $h(a) = 16$, $h(b) = 6$, $h(c) = 14$, $h(d) = 4$, $h(e) = 2$, $h(f) = 0$. Početno stanje je a, a ciljno f. Izvedite korake algoritma A^*, bilježeći u svakom koraku sadržaj liste otvorenih čvorova O i skupa zatvorenih čvorova C. U nultom koraku algoritma vrijedi $O = [(a, 0)]$ i $C = \emptyset$. **Koji je sadržaj listi O i skupa C nakon petog koraka izvođenja algoritma?**

- A $O = [(d, 3)]$, $C = \{(a, 0),(b, 2),(c, 2),(e, 6)\}$
- B Algoritam ne dostiže peti korak
- C $O = [(d, 3),(f, 20)]$, $C = \{(a, 0),(b, 2),(c, 2)\}$
- D $O = [(d, 3),(f, 20)]$, $C = \{(a, 0),(b, 2),(c, 2),(e, 6)\}$

2 (R) Stablo igre definirano je prijelazima $succ(A) = \{B, C\}, succ(B) = \{D, E\}, succ(D) = \{H, I\}, succ(E) = \{J, K, L\}, succ(C) = \{F, G\}, succ(F) = \{M, N, O\}, succ(G) = \{P, Q\}$. Heurističke vrijednosti listova su $h(H) = 3, h(I) = -14, h(J) = -3, h(K) = h(Q) = -5, h(L) = 7, h(M) = -1, h(N) = -10, h(O) = -1, h(P) = 17$. Optimalna strategija određuje se algoritmom minimax uz alfa-beta podrezivanje. **Koji će čvorovi pritom biti podrezani (preskočeni pri izračunu minimax vrijednosti)?**

A G, P, Q B L, O, G, P, Q C Q D L, F, M, N, O

3 (P) Razmatramo problem slagalice dimenzija 5×5 s prosječnom dubinom rješenja $d = 10$. Za nalaženje rješenja algoritam pretraživanja u dubinu u prosjeku koristi 1 KB memorije. **Koliko bi memorije u prosjeku koristio algoritam pretraživanja u širinu na ovome problemu?**

A 25.6 MB B 160 KB C 3.4 MB D 7.6 MB

4 (T) Algoritam minimax koristi heurističku funkciju koja se poziva nad svim nezavršnim stanjima u stablu igre na određenoj dubini k. Igraći algoritmi općenito koriste različite heurističke funkcije. No, **što bi se dogodilo kada bi u igri sudjelovala dva algoritma koji imaju identične heurističke funkcije?**

- A Ništa se posebno ne bi dogodilo, jer i dalje oba algoritma ne bi mogla pretraživanjem u dubinu dosegnuti konačna stanja igre s definiranim isplatama
- B Igrač koji je prvi na potezu bi pobijedio, osim ako u igri ima elemenata slučajnosti, i tada bi igra bila izjednačena
- C Ako je mehanizam odabira poteza kod izjednačavanja minimax-vrijednosti poznat i determinističan, ishod igre mogao bi se izračunati na početku igre
- D Prvi igrač mogao bi unaprijed izračunati sve poteze protivnika do dubine k te tako odrediti globalno optimalnu minimax-strategiju

5 (T) Za razliku od slijepog pretraživanja prostora stanja, heurističko je pretraživanje usmjereno procjenom udaljenosti trenutačnog stanja do ciljnog stanja. **Koja je prednost heurističkog pretraživanja prostora stanja nad slijepim pretraživanjem?**

- A Reducira apriornu vremensku složenost pretraživanja, uz uvjet da se koristi skup posjećenih stanja
- B Ako je heuristika optimistična, osigurava potpunost, odnosno nalaženje ciljnog stanja ako ono postoji
- C Smanjuje očekivan broj koraka izvođenja algoritma, a time i broj proširenih čvorova te zauzeće memorije
- D Ima gornju ogradu na apriornu vremensku i prostornu složenost pretraživanja

2. Prikazivanje znanja, automatsko zaključivanje i logičko programiranje (5 pitanja)

6 (P) Za automatsko zaključivanje u PL koristimo rezoluciju opovrgavanjem uz strategiju skupa potpore (SOS). Na takav postupak zaključivanja možemo gledati kao na problem pretraživanje prostora stanja, gdje stanja odgovaraju skupu klauzula (onih zadanih i onih izvedenih), a prijelazi između stanja odgovaraju primjeni rezolucijskog pravila na jedan par klauzula. Takav problem pretraživanje ima i svoj faktor grananja, koji ovisi o dubini stabla, tj. o koraku zaključivanja. Neka skup premisa sadrži 10 klauzula, a negirani cilj 5 klauzula. **Koliko iznosi gornja ograda na faktor grananja u drugom koraku zaključivanja?** (Napomena: Jednom razriješeni par klauzula više se ne razrješava.)

A 74 B 119 C 104 D 60

7 (T) Rezolucija opovrgavanjem je ispravno i potpuno pravilo zaključivanja u FOL. **Što to znači?**

A Postupak izvodi klauzulu NIL ako i samo ako je negiran cilj logička posljedica premisa

B Postupak završava u konačnom broju koraka s odlukom je li formula logička posljedica premisa ili nije

C Kad god je skup premisa nekonzistentan, rezolucijsko pravilo izvodi NIL klauzulu

D Ako formula nije logička posljedica, onda to ne možemo dokazati rezolucijom opovrgavanjem

8 (P) Važno svojstvo pravila zaključivanje jest da je ono ispravno. **Koje je od sljedećih pravila zaključivanja ispravno?**

A $A \to (B \to C) \vdash (A \land B) \to C$ C $A \to B, B \to C \vdash C \to \neg A$

B $\neg A, A \to B \vdash \neg B$ D $B, A \to B \vdash A$

9 (R) Zadane su sljedeće dvije premise:

$$\exists x\Big(\exists y R(y) \lor \forall z \neg Q(x,z)\Big) \to P(a),\ \exists x \forall y \Big(Q(x,y) \to R(y)\Big)$$

Koju je od sljedećih formula moguće dokazati iz ovih premisa rezolucijom opovrgavanjem?

A $\forall x R(x)$ B $\exists x(P(x) \lor R(x))$ C $\exists x P(x) \land P(b)$ D $\forall x P(x)$

10 (T) Programi u Prologu sastoje se od definitnih klauzula, koje odgovaraju pravilima i činjenicama. **Što je razlika između pravila i činjenica u Prologu?**

A Pravila imaju najviše jedan negativan literal i barem jedan pozitivan literal, dok su činjenice jedinične klauzule sa samo jednim pozitivnim literalom

B Pravila imaju antecedent s najviše jednim pozitivnim literalom, dok činjenice odgovaraju pravilima s uvijek lažnim antecedentom

C Pravila imaju antecedent s barem jednim pozitivnim literalom, dok su činjenice jedinične klauzule sa samo jednim pozitivnim literalom

D Pravila imaju najviše jedan negativan literal i barem jedan pozitivan literal, dok činjenice odgovaraju pravilima s uvijek lažnim antecedentom

3. Modeliranje neizvjesnosti (2 pitanja)

11 (R) Bayesovom shemom zaključujemo o vjerojatnosti da neka država provodi nuklearne pokuse. Apriornu vjerojatnost da država provodi nuklearne pokuse, $P(N)$, izračunavamo kao procjenu najveće izglednosti (MLE) na temelju informacije da je za devet od 195 država poznato da imaju nuklearno oružje te da svaka treća od njih aktivno provodi nuklearne pokuse. Kao dokaze provođenja nuklearnog pokusa koristimo podatak o detekciji snažne sezmičke aktivnosti (S) te podatak o intenzivnoj proizvodnji obogaćenog uranija (U). Znamo da je vjerojatnost detekcije seizmičke aktivnosti uslijed nuklearnog pokusa jednaka 0.6, a vjerojatnost sezmičke aktivnosti bez nuklearnog pokusa svega 0.002. Također znamo je vjerojatnost proizvodnje obogaćenog uranija ako je proveden nuklearni pokus 0.3, dok bez provođenja nuklearnog pokusa ta vjerojatnost iznosi 0.2. Izračunajte aposteriornu vjerojatnost provođenja nuklearnog pokusa, i to prvo samo uz dokaz U, a zatim uz dodatni dokaz S, pretpostavljajući pritom uvjetnu nezavisnost tih dvaju dokaza. **Koliko iznosi porast aposteriorne vjerojatnosti nakon dodavanja drugog dokaza?**

A 0.853 B 0.785 C 0.714 D 0.897

12 (P) Razmotrimo jezičnu varijablu *starost kanarinca*, definiranu s izrazima *mlad* i *star*. Značenje tih izraza modeliramo neizrazitim skupovima M odnosno S, definiranima nad univerzalnim skupom $\mathbb{R}^+$, koji odgovara godinama. Funkcije pripadnosti μ_M i μ_S definiramo kao po dijelovima linearne funkcije. Funkcija $\mu_M(x)$ ima vrijednost 1 za $0 \leq x \leq 5$, vrijednost 0 za $x \geq 9$, te linearno pada za $5 < x < 9$. Funkcija $\mu_S(x)$ ima vrijednost 0 za $0 \leq x \leq 6$, vrijednost 1 za $x \geq 12$, te linearno raste za $6 < x < 12$. Uporabom Zadehovih operatora definiramo dva neizrazita skupa: skup X_1 sa značenjem *mlad ili star kanarinac* te skup X_2 sa značenjem *mlad ili ne mlad kanarinac*. **Za koje se sve elemente $x \in \mathbb{R}^+$ pripadnost ovim dvama neizrazitim skupovima razlikuje?**

A $5 \leq x \leq 6$ B $9 \leq x \leq 12$ C $7 \leq x \leq 12$ D $7 \leq x \leq 9$

4. Strojno učenje i umjetne neuronske mreže (5 pitanja)

13 (P) Na računalu implementiramo unaprijednu potpuno povezanu slojevitu umjetnu neuronsku mrežu arhitekture $3 \times 20 \times 10 \times 5 \times 2$. Neuroni kao prijenosne funkcije koriste zglobnicu. U memoriji težine mreže čuvaju se kao tip `double` koji zauzima 8 okteta. **Koliki je ukupni utrošak memorije za parametre ove mreže?**

A 4218 B 5096 C 2560 D 2856

14 (R) Zadan je skup primjeraka za učenje oblika $\{((x_{2,i}, x_{1,i}), y_i)\}$, koji sadrži pet primjeraka:

$$\{((1,1),-1),((3,1),1),((2,1),-1),((1,2),-1),((3,2),1)\}$$

Time je definirano preslikavanje $y_i = f(x_2, x_1)$. Ovo preslikavanje želimo naučiti TLU-perceptronom koji koristi prijenosnu funkciju skoka -1,1 (neka je izlaz -1 ako je *net* < 0, 1 inače). Učimo Rosenblattovim postupkom, pri čemu je iznos stope učenja jednak 1. Neka je početni vektor težina $[w_2, w_1, w_0] = [8, 3, -12]$. **Koliko će se puta provesti korigiranje vektora težina i koja je njegova konačna vrijednost?**

A 3 puta, $[6, 1, -14]$ B 3 puta, $[12, 1, -28]$ C 2 puta, $[6, 3, -15]$ D 2 puta, $[7, 2, -12]$

15 (T) Stabla odluke i Bayesov klasifikator dva su algoritma nadziranog učenja. Međutim, ti se algoritmi vrlo razlikuju. **Što je prednost, a što nedostatak stabla odluke u odnosu na Bayesov klasifikator?**

A Prednost je što su stabla odluke otporna na male promjene u ulaznom skupu podataka, a nedostatak što pretpostavljamo uvjetnu nezavisnost značajki

B Prednost je što možemo bolje objasniti zašto je primjer klasificiran u neku klasu, a nedostatak što nemamo vjerojatnost klasifikacijske odluke

C Prednost je što primjere možemo klasificirati u više od jedne klase, a nedostatak što se stablo odluke može lako prenaučiti

D Prednost je što stablo odluke možemo podrezati kako bismo spriječili prenaučenost, a nedostatak što može doći do podljeva pri računanju vjerojatnosti

16 (R) Mali je Ivica svakog svakog ljeta u zadnjih sedam godina naučio jedan novi programski jezik. Svoja vrijedna iskustva sažeo je u listu *"Programski jezik koji mi se sviđa"*, gdje je svaki jezik opisao četirima značajkama, te je naznačio je li mu se dotični jezik svidio ($y = 1$) ili nije ($y = 0$). Ta lista izgleda ovako:

i	Evaluacija (E)	Izvođenje (I)	Paradigma (P)	Provjera tipova (T)	y
1	lijena	kompajler	imperativna	statička	0
2	striktna	interpreter	deklarativna	dinamička	0
3	lijena	kompajler	imperativna	dinamička	0
4	lijena	interpreter	hibridna	statička	0
5	striktna	interpreter	imperativna	statička	1
6	lijena	kompajler	hibridna	dinamička	1
7	striktna	kompajler	hibridna	dinamička	1

Ovog ljeta Mali Ivica želi puno jesti i spavati te opet naučiti novi programski jezik. Međutim, Mali Ivica ne želi izgubiti ljetno na programskom jeziku koji mu se ne sviđa, pa je odlučio izgraditi stablo odluke za klasifikaciju programskih jezika koje još nije naučio. Pomozite Malom Ivici te na gornjem skupu primjera primjenite algoritam ID3. U slučaju da dvije značajke ili više njih imaju jednaku vrijednost informacijske dobiti, prednost dajte značajki koje je u gornjoj tablici navedena prva (ljevija značajka). **Koje se značajke nalaze u drugoj razini stabla odluke dobivenog algoritmom ID3?**

A E, P, T B E, I C E, I, P D E, P

17 (T) U nadziranom strojnom učenju, skup podataka često dijelimo u tri međusobno disjunktna podskupa: skup za učenje, skup za provjeru i skup za ispitivanje. Model učimo isključivo na skupu za učenje. **Čemu nam služe skup za provjeru i skup za ispitivanje?**

- A Na skupu za provjeru ispitujemo točnost modela, a na skupu za ispitivanje određujemo predikcijsku pogrešku modela najveće složenosti
- B Na skupu za provjeru određujemo koliko model dobro generalizira, a na skupu za ispitivanje određujemo pogrešku modela na još neviđenim podatcima
- C Skup za provjeru služi za učenje značajki modela, dok skup za ispitivanje služi za određivanje pogreške generalizacije modela različitih složenosti
- D Na skupu za provjeru izračunavamo točnost modela različitih složenosti, a na skupu za ispitivanje vrednujemo točnost modela optimalne složenosti

5. Prirodom inspirirani optimizacijski algoritmi (3 pitanja)

18 (T) Genetski algoritmi pripadaju grupi metaheurističkih stohastičkih optimizacijskih algoritama. **Što je kod njega nužno da bi algoritam konvergirao prema traženom rješenju?**

A Generacijska izvedba B Operator križanja C Operator selekcije D Velika populacija

19 (P) Algoritmom kolonije mrava rješavamo problem trgovačkog putnika nad odabranim gradovima u Republici Hrvatskoj. Za algoritam su poznati sljedeći parametri: $\tau_0 = 40$, $\alpha = 2$, $\beta = 3$, $\rho = 0.15$, kolonija se sastoji od 55 mrava te ažuriranje u svakoj epohi obavlja samo mrav koji je u toj epohi pronašao najbolje rješenje. Razmotrimo brid koji povezuje gradove Zagreb i Dubrovnik. Pretpostavimo da niti u jednoj epohi ta dva grada nisu sljedbenici u pronađenom putu. **Nakon koliko će epoha algoritma količina feromonskih tragova na spomenutom bridu postati manja od 0.1?**

A 37 B 43 C 4 D 52

20 (R) Generacijskim genetskim algoritmom tražimo maksimum funkcije $f(x, y) = 18 - (x - 4)^2 - (y - 2)^2$. Kao reprezentaciju rješenja koristimo 6-bitovni kromosom, pri čemu se prva tri bita koriste za kodiranje vrijednosti varijable x, a preostala tri bita za kodiranje vrijednosti varijable y. Domena nad kojom se pretražuju vrijednosti od x je $[2, 9]$ te od y je $[-2, 5]$. Populacija se sastoji od četiri jedinke: J1=000110, J2=011001, J3=100111, J4=001110. Pretpostavite da se roditelji biraju proporcionalnom selekcijom te da su u jednom koraku kao roditelji izvučena dva rješenja koja imaju najmanju vjerojatnost odabira. Provedite nad njima postupak križanja s jednom točkom prijeloma (točka prijeloma je nakon prva dva bita); pretpostavite da u ovom koraku operator mutacije svaki puta djeluje na posljednja dva bita kromosoma. **Odredite iznos funkcije f u rješenju koje odgovara djetetu koje će biti poslano u sljedeću generaciju.** (Ako ih operatori križanja i mutacije generiraju više, u novu generaciju treba proslijediti najbolje.)

A 9 B 16 C 5 D 18

Uvod u umjetnu inteligenciju – pismeni ispit (2020./2021.)

– NEKORIGIRANA VERZIJA –

*Ispit se sastoji od **20 pitanja** i ukupno nosi **20 bodova**. Točan odgovor nosi 1 bod a netočan -1/3 boda. Trajanje ispita je **150 minuta**. Primjerak ispita trebate predati zajedno sa svojim rješenjima.*

1. Pretraživanje prostora stanja i igranje igara (5 pitanja)

1 (P) Razmatramo problem slagalice dimenzija 3×4 s prosječnom dubinom rješenja $d = 11$. Za nalaženje rješenja algoritam pretraživanja u dubinu u prosjeku koristi 720 B memorije. **Koliko bi memorije u prosjeku koristio algoritam pretraživanja u širinu na ovome problemu?**

A 720 KB B 3 KB C 2 MB D 65.5 MB

2 (R) Stablo igre definirano je prijelazima $succ(A) = \{B, C\}, succ(B) = \{D, E\}, succ(D) = \{H, I\}, succ(E) = \{J, K, L\}, succ(C) = \{F, G\}, succ(F) = \{M, N, O\}, succ(G) = \{P, Q\}$. Heurističke vrijednosti listova su $h(H) = 10, h(I) = -h(O) = 3, h(J) = 13, h(K) = h(N) = -9, h(L) = -11, h(M) = 5, h(P) = 7, h(Q) = -16$. Optimalna strategija određuje se algoritmom minimax uz alfa-beta podrezivanje. **Koji će čvorovi pritom biti podrezani (preskočeni pri izračunu minimax vrijednosti)?**

A O, P, Q B K, L, G, P, Q C L, F, M, N, O D I, K, L, F, M, N, O, Q

3 (T) Algoritam minimax koristi heurističku funkciju koja se poziva nad svim nezavršnim stanjima u stablu igre na određenoj dubini k. Igraći algoritmi općenito koriste različite heurističke funkcije. No, **što bi se dogodilo kada bi u igri sudjelovala dva algoritma koji imaju identične heurističke funkcije?**

A Ako je mehanizam odabira poteza kod izjednačavanja minimax-vrijednosti poznat i determinističan, ishod igre mogao bi se izračunati na početku igre

B Igrač koji je prvi na potezu bi pobijedio, osim ako u igri ima elemenata slučajnosti, i tada bi igra bila izjednačena

C Ništa se posebno ne bi dogodilo, jer i dalje oba algoritma ne bi mogla pretraživanjem u dubinu dosegnuti konačna stanja igre s definiranim isplatama

D Prvi igrač mogao bi unaprijed izračunati sve poteze protivnika do dubine k te tako odrediti globalno optimalnu minimax-strategiju

4 (T) Za razliku od slijepog pretraživanja prostora stanja, heurističko je pretraživanje usmjereno procjenom udaljenosti trenutačnog stanja do ciljnog stanja. **Koja je prednost heurističkog pretraživanja prostora stanja nad slijepim pretraživanjem?**

A Ima gornju ogradu na apriornu vremensku i prostornu složenost pretraživanja

B Ako je heuristika optimistična, osigurava potpunost, odnosno nalaženje ciljnog stanja ako ono postoji

C Smanjuje očekivan broj koraka izvođenja algoritma, a time i broj proširenih čvorova te zauzeće memorije

D Reducira apriornu vremensku složenost pretraživanja, uz uvjet da se koristi skup posjećenih stanja

5 (R) Prostor stanja pretražujemo algoritmom A^*. Skup stanja je $S = \{a, b, c, d, e, f\}$, a funkcija sljedbenika je $succ(a) = \{(b, 2), (c, 2)\}$, $succ(b) = \{(c, 5), (d, 2)\}$, $succ(c) = \{(d, 1), (f, 20)\}$, $succ(d) = \{(e, 2)\}$, $succ(e) = \{(f, 14)\}$ te $succ(f) = \emptyset$. Heurističke vrijednosti stanja su $h(a) = 16$, $h(b) = 6$, $h(c) = 14$, $h(d) = 4$, $h(e) = 2$, $h(f) = 0$. Početno stanje je a, a ciljno f. Izvedite korake algoritma A^*, bilježeći u svakom koraku sadržaj liste otvorenih čvorova O i skupa zatvorenih čvorova C. U nultom koraku algoritma vrijedi $O = [(a, 0)]$ i $C = \emptyset$. **Koji je sadržaj liste O i skupa C nakon šestog koraka izvođenja algoritma?**

A $O = [(e, 5), (f, 20)],\ C = \{(a, 0), (b, 2), (c, 7)\}$

B $O = [(e, 5), (f, 20)],\ C = \{(a, 0), (b, 2), (c, 2), (d, 3)\}$

C Algoritam ne dostiže šesti korak

D $O = [(e, 5), (f, 22)],\ C = \{(a, 0), (b, 2), (c, 7), (d, 3)\}$

2. Prikazivanje znanja, automatsko zaključivanje i logičko programiranje (5 pitanja)

6 (P) Važno svojstvo pravila zaključivanje jest da je ono ispravno. **Koje je od sljedećih pravila zaključivanja ispravno?**

A $A \to (B \to C) \vdash (A \lor B) \to C$
C $A \to (B \to C) \vdash (A \land B) \to C$
B $A \vdash A \land B$
D $A \lor B \vdash B$

7 (P) Za automatsko zaključivanje u PL koristimo rezoluciju opovrgavanjem uz strategiju skupa potpore (SOS). Na takav postupak zaključivanja možemo gledati kao na problem pretraživanje prostora stanja, gdje stanja odgovaraju skupu klauzula (onih zadanih i onih izvedenih), a prijelazi između stanja odgovaraju primjeni rezolucijskog pravila na jedan par klauzula. Takav problem pretraživanje ima i svoj faktor grananja, koji ovisi o dubini stabla, tj. o koraku zaključivanja. Neka skup premisa sadrži 10 klauzula, a negirani cilj 5 klauzula. **Koliko iznosi gornja ograda na faktor grananja u drugom koraku zaključivanja?** (Napomena: Jednom razriješeni par klauzula više se ne razrješava.)

A 74 B 104 C 119 D 60

8 (T) Rezolucija opovrgavanjem je ispravno i potpuno pravilo zaključivanja u FOL. **Što to znači?**

A Postupak završava u konačnom broju koraka s odlukom je li formula logička posljedica premisa ili nije
B Postupak izvodi klauzulu NIL ako i samo ako je negiran cilj logička posljedica premisa
C Ako formula nije logička posljedica, onda to ne možemo dokazati rezolucijom opovrgavanjem
D Kad god je skup premisa nekonzistentan, rezolucijsko pravilo izvodi NIL klauzulu

9 (R) Zadane su sljedeće dvije premise:

$$\exists x\Big(\exists y R(y) \lor \forall z \neg Q(x,z)\Big) \to P(a),\ \exists x \forall y\Big(Q(x,y) \to R(y)\Big)$$

Koju je od sljedećih formula moguće dokazati iz ovih premisa rezolucijom opovrgavanjem?

A $\exists x \neg P(x)$ B $\exists x P(x) \land P(b)$ C $\exists x P(x) \lor P(b)$ D $\neg \exists x P(x)$

10 (T) Programi u Prologu sastoje se od definitnih klauzula, koje odgovaraju pravilima i činjenicama. **Što je razlika između pravila i činjenica u Prologu?**

A Pravila imaju najviše jedan negativan literal i barem jedan pozitivan literal, dok su činjenice jedinične klauzule sa samo jednim pozitivnim literalom
B Pravila imaju barem jedan negativan literal i točno jedan pozitivan literal, dok su činjenice jedinične klauzule sa samo jednim pozitivnim literalom
C Pravila imaju barem jedan negativan literal i točno jedan pozitivan literal, dok činjenice odgovaraju pravilima s uvijek istinitim konzekvensom
D Pravila imaju barem jedan negativan literal i točno jedan pozitivan literal, dok su činjenice jedinične klauzule sa barem jednim negativnim i točno jednim pozitivnim literalom

3. Modeliranje neizvjesnosti (2 pitanja)

11 (R) Bayesovom shemom zaključujemo o vjerojatnosti da neka država provodi nuklearne pokuse. Apriornu vjerojatnost da država provodi nuklearne pokuse, $P(N)$, izračunavamo kao procjenu najveće izglednosti (MLE) na temelju informacije da je za devet od 195 država poznato da imaju nuklearno oružje te da svaka treća od njih aktivno provodi nuklearne pokuse. Kao dokaze provođenja nuklearnog pokusa koristimo podatak o detekciji snažne sezmičke aktivnosti (S) te podatak o intenzivnoj proizvodnji obogaćenog uranija (U). Znamo da je vjerojatnost detekcije seizmičke aktivnosti uslijed nuklearnog pokusa jednaka 0.8, a vjerojatnost sezmičke aktivnosti bez nuklearnog pokusa svega 0.002. Također znamo je vjerojatnost proizvodnje obogaćenog uranija ako je proveden nuklearni pokus 0.5, dok bez provođenja nuklearnog pokusa ta vjerojatnost iznosi 0.1. Izračunajte aposteriornu vjerojatnost provođenja nuklearnog pokusa, i to prvo samo uz dokaz U, a zatim uz dodatni dokaz S, pretpostavljajući pritom uvjetnu nezavisnost tih dvaju dokaza. **Koliko iznosi porast aposteriorne vjerojatnosti nakon dodavanja drugog dokaza?**

A 0.897 B 0.714 C 0.853 D 0.785

12 (P) Razmotrimo jezičnu varijablu *starost kanarinca*, definiranu s izrazima *mlad* i *star*. Značenje tih izraza modeliramo neizrazitim skupovima M odnosno S, definiranima nad univerzalnim skupom $\mathbb{R}^+$, koji odgovara godinama. Funkcije pripadnosti μ_M i μ_S definiramo kao po dijelovima linearne funkcije. Funkcija $\mu_M(x)$ ima vrijednost 1 za $0 \leq x \leq 5$, vrijednost 0 za $x \geq 9$, te linearno pada za $5 < x < 9$. Funkcija $\mu_S(x)$ ima vrijednost 0 za $0 \leq x \leq 6$, vrijednost 1 za $x \geq 12$, te linearno raste za $6 < x < 12$. Uporabom Zadehovih operatora definiramo dva neizrazita skupa: skup X_1 sa značenjem *mlad ili star kanarinac* te skup X_2 sa značenjem *mlad ili ne mlad kanarinac*. **Za koje se sve elemente $x \in \mathbb{R}^+$ pripadnost ovim dvama neizrazitim skupovima razlikuje?**

A $7 \leq x \leq 12$ B $5 \leq x \leq 6$ C $9 \leq x \leq 12$ D $7 \leq x \leq 9$

4. Strojno učenje i umjetne neuronske mreže (5 pitanja)

13 (P) Na računalu implementiramo unaprijednu potpuno povezanu slojevitu umjetnu neuronsku mrežu arhitekture $3 \times 40 \times 10 \times 5 \times 2$. Neuroni kao prijenosne funkcije koriste zglobnicu. U memoriji težine mreže čuvaju se kao tip `double` koji zauzima 8 okteta. **Koliki je ukupni utrošak memorije za parametre ove mreže?**

A 8726 B 4640 C 5096 D 2856

14 (T) Stabla odluke i Bayesov klasifikator dva su algoritma nadziranog učenja. Međutim, ti se algoritmi vrlo razlikuju. **Što je prednost, a što nedostatak stabla odluke u odnosu na Bayesov klasifikator?**

A Prednost je što primjere možemo klasificirati u više od jedne klase, a nedostatak što se stablo odluke može lako prenaučiti

B Prednost je što su stabla odluke otporna na male promjene u ulaznom skupu podataka, a nedostatak što pretpostavljamo uvjetnu nezavisnost značajki

C Prednost je što stablo odluke možemo podrezati kako bismo spriječili prenaučenost, a nedostatak što može doći do podljeva pri računanju vjerojatnosti

D Prednost je što možemo bolje objasniti zašto je primjer klasificiran u neku klasu, a nedostatak što nemamo vjerojatnost klasifikacijske odluke

15 (R) Mali je Ivica svakog svakog ljeta u zadnjih sedam godina naučio jedan novi programski jezik. Svoja vrijedna iskustva sažeo je u listu *"Programski jezik koji mi se sviđa"*, gdje je svaki jezik opisao četirima značajkama, te je naznačio je li mu se dotični jezik svidio ($y = 1$) ili nije ($y = 0$). Ta lista izgleda ovako:

i	Evaluacija (E)	Izvođenje (I)	Paradigma (P)	Provjera tipova (T)	y
1	lijena	kompajler	imperativna	statička	0
2	striktna	interpreter	deklarativna	dinamička	0
3	lijena	kompajler	imperativna	dinamička	0
4	lijena	interpreter	hibridna	statička	0
5	striktna	interpreter	imperativna	statička	1
6	lijena	kompajler	hibridna	dinamička	1
7	striktna	kompajler	hibridna	dinamička	1

Ovog ljeta Mali Ivica želi puno jesti i spavati te opet naučiti novi programski jezik. Međutim, Mali Ivica ne želi izgubiti ljetno na programskom jeziku koji mu se ne sviđa, pa je odlučio izgraditi stablo odluke za klasifikaciju programskih jezika koje još nije naučio. Pomozite Malom Ivici te na gornjem skupu primjera primjenite algoritam ID3. U slučaju da dvije značajke ili više njih imaju jednaku vrijednost informacijske dobiti, prednost dajte značajki koje je u gornjoj tablici navedena prva (ljevija značajka). **Koje se značajke nalaze u drugoj razini stabla odluke dobivenog algoritmom ID3?**

A E, I, P B E, P C E, P, T D E, I

16 (R) Zadan je skup primjeraka za učenje oblika $\{((x_{2,i}, x_{1,i}), y_i)\}$, koji sadrži pet primjeraka:

$$\{((3,2),1), ((1,1),-1), ((2,1),-1), ((3,1),1), ((1,2),-1)\}$$

Time je definirano preslikavanje $y_i = f(x_2, x_1)$. Ovo preslikavanje želimo naučiti TLU-perceptronom koji koristi prijenosnu funkciju skoka -1,1 (neka je izlaz -1 ako je *net* < 0, 1 inače). Učimo Rosenblattovim postupkom, pri čemu je iznos stope učenja jednak 0.5. Neka je početni vektor težina $[w_2, w_1, w_0] = [7, 3, -12]$. **Koliko će se puta provesti korigiranje vektora težina i koja je njegova konačna vrijednost?**

A 1 puta, $[10, 4, -26]$ B 2 puta, $[8, -1, -11]$ C 1 puta, $[5, 2, -13]$ D 2 puta, $[7, 2, -12]$

17 (T) U nadziranom strojnom učenju, skup podataka često dijelimo u tri međusobno disjunktna podskupa: skup za učenje, skup za provjeru i skup za ispitivanje. Model učimo isključivo na skupu za učenje. **Čemu nam služe skup za provjeru i skup za ispitivanje?**

- A Na skupu za provjeru ispitujemo točnost modela, a na skupu za ispitivanje određujemo predikcijsku pogrešku modela najveće složenosti
- B Na skupu za provjeru izračunavamo točnost modela različitih složenosti, a na skupu za ispitivanje vrednujemo točnost modela optimalne složenosti
- C Na skupu za provjeru određujemo koliko model dobro generalizira, a na skupu za ispitivanje određujemo pogrešku modela na još neviđenim podatcima
- D Skup za provjeru služi za učenje značajki modela, dok skup za ispitivanje služi za određivanje pogreške generalizacije modela različitih složenosti

5. Prirodom inspirirani optimizacijski algoritmi (3 pitanja)

18 (P) Algoritmom kolonije mrava rješavamo problem trgovačkog putnika nad odabranim gradovima u Republici Hrvatskoj. Za algoritam su poznati sljedeći parametri: $\tau_0 = 100$, $\alpha = 2$, $\beta = 3$, $\rho = 0.1$, kolonija se sastoji od 100 mrava te ažuriranje u svakoj epohi obavlja samo mrav koji je u toj epohi pronašao najbolje rješenje. Razmotrimo brid koji povezuje gradove Zagreb i Dubrovnik. Pretpostavimo da niti u jednoj epohi ta dva grada nisu sljedbenici u pronađenom putu. **Nakon koliko će epoha algoritma količina feromonskih tragova na spomenutom bridu postati manja od 1?**

A 21 B 2 C 44 D 53

19 (T) Genetski algoritmi pripadaju grupi metaheurističkih stohastičkih optimizacijskih algoritama. **Što je kod njega nužno da bi algoritam konvergirao prema traženom rješenju?**

A Operator selekcije B Generacijska izvedba C Velika populacija D Operator križanja

20 (R) Generacijskim genetskim algoritmom tražimo maksimum funkcije $f(x, y) = 13 - (x - 2)^2 - (y - 3)^2$. Kao reprezentaciju rješenja koristimo 6-bitovni kromosom, pri čemu se prva tri bita koriste za kodiranje vrijednosti varijable x, a preostala tri bita za kodiranje vrijednosti varijable y. Domena nad kojom se pretražuju vrijednosti od x je $[0, 7]$ te od y je $[-1, 6]$. Populacija se sastoji od četiri jedinke: J1=000110, J2=011001, J3=100111, J4=001110. Pretpostavite da se roditelji biraju proporcionalnom selekcijom te da su u jednom koraku kao roditelji izvučena dva rješenja koja imaju najmanju vjerojatnost odabira. Provedite nad njima postupak križanja s jednom točkom prijeloma (točka prijeloma je nakon prva dva bita); pretpostavite da u ovom koraku operator mutacije svaki puta djeluje na posljednja dva bita kromosoma. **Odredite iznos funkcije f u rješenju koje odgovara djetetu koje će biti poslano u sljedeću generaciju.** (Ako ih operatori križanja i mutacije generiraju više, u novu generaciju treba proslijediti najbolje.)

A 0 B 13 C 11 D 4

	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20
Grupa A	B	D	C	C	A	B	C	A	A	A	B	B	D	A	A	D	B	C	A	D
Grupa B	C	A	A	C	B	B	B	D	A	D	C	A	B	D	D	C	D	B	A	B
Grupa C	D	A	D	B	A	D	C	A	C	B	D	C	A	C	A	A	A	C	A	C
Grupa D	B	A	D	B	B	C	C	B	B	D	C	B	C	B	D	D	C	A	C	B
Grupa E	D	A	C	C	C	A	A	A	B	C	A	C	D	A	B	B	D	C	A	D
Grupa F	C	B	A	C	B	C	A	B	C	B	A	A	C	D	D	C	B	C	A	B