

Uvod u umjetnu inteligenciju – pismeni ispit (2022./2023.) – 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 (4 pitanja)

- 1** (R) Neka su definirani skup stanja $S = \{a, b, c, d, e, f\}$ i funkcija sljedbenika $\text{succ}(a) = \{(b, 1), (c, 2)\}$, $\text{succ}(b) = \text{succ}(f) = \emptyset$, $\text{succ}(c) = \{(d, 3), (e, 5)\}$, $\text{succ}(d) = \{(b, 8), (f, 5)\}$ te $\text{succ}(e) = \{(d, 1), (f, 2)\}$. Heurističke vrijednosti čvorova neka su $h(a) = 7$, $h(b) = 4$, $h(c) = 6$, $h(d) = h(e) = 2$, $h(f) = 0$. Početno stanje neka je a , a ciljno f . Izvršite pretraživanje algoritmom A^* , bilježeći pritom sadržaje liste otvorenih čvorova O i zatvorenih čvorova C u svakom koraku algoritma (u nultom koraku $O = [(a, 0)]$ i $C = \emptyset$). **Koji su sadržaji listi O i C nakon četvrtog koraka izvođenja algoritma A^* ?**

- ☐ A $O = [(e, 7), (f, 10)]$, $C = \{(a, 0), (c, 2), (d, 5)\}$
☐ B $O = [(f, 0), (e, 2), (b, 4)]$, $C = \{(a, 0), (b, 1), (c, 2), (d, 5)\}$
☐ C $O = [(e, 7), (f, 10)]$, $C = \{(a, 0), (b, 1), (c, 2), (d, 5)\}$
☐ D Algoritam ne dostiže četvrti korak

- 2** (P) Razmatramo dva igrača algoritma: jedan s heuristikom h_1 i dubinom pretraživanja d_1 te drugi s heuristikom h_2 i dubinom pretraživanja d_2 . Oba algoritma koriste strategiju minimax. **Kada će prvi algoritam češće pobjeđivati drugi algoritam?**

- ☐ A Ako $d_1 = d_2$ i $h_1 < h_2$ ☐ B Ako $d_1 > d_2$ ☐ C Ako $h_2 > h_1$ ☐ D Ako $h_1 = h_2$ i $d_1 > d_2$

- 3** (T) Algoritam minimax s dubinskim ograničenjem uobičajeno je izveden pomoću dvije, međusobno rekurzivne funkcije. Neka su to funkcije $\text{maxValue}(s, d)$ i $\text{minValue}(s, d)$, gdje je s stanje igre, a d je brojač razine u stablu igre. Početni poziv algoritma je $\text{maxValue}(s_0, k)$, gdje je s_0 trenutačno stanje igre, a k je dubinsko ograničenje. Funkcija $\text{maxValue}(s, d)$ koristi petlju s iteratorom $t \in \text{succ}(s)$ kako bi izračunala minimax-vrijednost m , koja je izvan petlje inicijalizirana na $-\infty$. **Kako glasi tijelo petlje u funkciji maxValue ?**

- ☐ A $m \leftarrow \max(m, \text{maxValue}(t, d + 1))$ ☐ C $m \leftarrow \max(m, \text{minValue}(t, d - 1))$
☐ B $m \leftarrow \min(m, \text{minValue}(t, d + 1))$ ☐ D $m \leftarrow \min(m, \text{maxValue}(t, d - 1))$

- 4** (P) Rješavamo problem pretraživanja prostora stanja za koji vrijedi $m = 10$, $d = 7$ i $b = 4$. Koristimo pretraživanje u širinu (BFS) i pretraživanje u dubinu (DFS). Za oba algoritma razmotrite najgori slučaj vremenske složenosti. **Koliko će u takvom slučaju algoritam DFS generirati više čvorova od algoritma BFS?**

- ☐ A 1376256 ☐ B 1310716 ☐ C 1310640 ☐ D 1376176

2. Prikazivanje znanja i automatsko zaključivanje (4 pitanja)

- 5** (P) Ispravnost je važno svojstvo pravila zaključivanja. **Koje je od sljedećih pravila zaključivanja ispravno?**

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

- 6** (R) Dani su FOL atomi $P(f(x), y, g(w, b))$ i $P(y, f(g(a, z)), x)$, gdje je P predikat, a i b su konstante, a f i g su funkcije. **Što je najopćenitiji zajednički unifikator ovih dvaju atoma?**

- ☐ A $P(f(y), f(g(a, z)), g(w, b))$ ☐ C $P(f(g(a, b)), f(g(a, b)), g(a, b))$
☐ B Atome nije moguće unificirati ☐ D $P(f(x), f(g(x, y)), g(a, b))$

7 (T) Relacija logičke posljedice između formula F i G definirana je na temelju vrijednosti istinitosti formula F i G kroz sve moguće interpretacije tih formula. Dokazivanje logičke posljedice od praktičnog je značaja za umjetnu inteligenciju. **Na koji način dokazujemo logičku posljedicu u FOL?**

- ☐ A Pomoću ispravnih i potpunih pravila zaključivanja
- ☐ B Primjenom izravne rezolucije i faktorizacije
- ☐ C Konstrukcijom tablice istinitosti i provjerom modela
- ☐ D Pretvorbom u klauzalni oblik i skolemizacijom

8 (R) Zadane su sljedeće dvije premise:

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

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

- ☐ A $\forall x P(x)$
- ☐ B $\exists x P(x)$
- ☐ C $\exists x P(x) \wedge P(b)$
- ☐ D $\forall x R(x)$

3. Logičko programiranje i ekspertni sustavi (2 pitanja)

9 (T) Kao i sustavi formalne logike, ekspertni se sustavi temelje na formalnom prikazu znanja i izvođenju novog znanja primjenom pravila zaključivanja. Međutim, između tih sustava postoje razlike. **Po čemu se ekspertni sustavi razlikuju od općenitih sustava formalne logike?**

- ☐ A Znanje je prikazano u obliku ako-onda pravila, a ne Hornovih ili definitnih klauzula
- ☐ B Za dokazivanje se koristi rezolucija opovrgavanjem, a ne izravna rezolucija
- ☐ C Za dokazivanje se koriste logička pravila koja su potpuna, a ne samo ispravna
- ☐ D Znanje je prikazano u obliku činjenica i pravila, a ne proizvoljnih logičkih formula

10 (T) Izvođenje logičkih programa u Prologu odgovara dokazivanju teorema rezolucijom opovrgavanjem. Stanje izvođenja u Prologu definirano je stogom, koji početno sadrži atom upita. **Što sadrži Prologov stog u nastavku izvođenja programa?**

- ☐ A Atome iz tijela logičkih pravila
- ☐ B Atome dviju roditeljskih klauzula
- ☐ C Činjenice iz logičkog programa
- ☐ D Atome iz glave logičkih pravila

4. Modeliranje neizvjesnosti (2 pitanja)

11 (R) Nad univerzalnim skupom $\{a, b, c, d\}$ zadani su neizraziti skupovi $X = \{0.1/a + 0/b + 0.3/c + 1/d\}$, $Y = \{0.5/a + 0.4/b + 0/c + 0.2/d\}$ i $Z = \{1/a + 0/b + 0/c + 0.5/d\}$. Koristeći Zadehove operatore izvodimo neizraziti skup X tako da on odgovara jezičnome izrazu “Ne X ili (Y i Z)”. **Kako glasi neizraziti skup X ?**

- ☐ A $\{0/a + 0.4/b + 0.3/c + 0.5/d\}$
- ☐ B $\{0.5/a + 0/b + 0/c + 0/d\}$
- ☐ C $\{0.9/a + 1.0/b + 0.7/c + 0.2/d\}$
- ☐ D $\{0.9/a + 0.4/b + 0/c + 0/d\}$

12 (T) Bayesovo pravilo možemo koristiti za zaključivanje abdukcijom. **Što to znači?**

- ☐ A Na temelju opažene hipoteze zaključujemo o dokazu
- ☐ B Na temelju opaženih dokaza zaključujemo o uzroku
- ☐ C Na temelju ostvarenog uzroka zaključujemo o dokazima
- ☐ D Na temelju ostvarenog uzroka zaključujemo o hipotezi

5. Strojno učenje, umjetne neuronske mreže i podržano učenje (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 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, $[6, 3, -15]$ ☐ B 3 puta, $[7, -1, -14]$ ☐ C 3 puta, $[14, -2, -28]$ ☐ D 2 puta, $[7, 2, -12]$

- 14** (P) Gradimo naivan Bayesov klasifikator za klasifikaciju poruka Twittera (tvitova) prema sentimentu. Svaki tvit želimo klasificirati u jednu od tri klase: pozitivan, negativan ili neutralan. Svaki tvit sastoji se od najviše 280 riječi i prikazujemo ga kao jedan binaran vektor značajki. Vektor se sastoji od 5000 značajki, i svaka značajka odgovara jednoj od 5000 riječi iz našeg vokabulara. Ako je značajka postavljena na 1, to znači da se dotična riječ pojavila u tvitu, inače je značajka postavljena na 0. Npr., ako je $x_{42} = 1$, onda to znači da se 42. riječ iz našeg vokabulara pojavila u tvitu. Učenje ovog modela svodi se na procjenu apriornih vjerojatnosti i izglednosti klase na temelju označenog skupa podataka. **Koliko ćemo ukupno vjerojatnosti trebati procijeniti za ovaj model?**

- ☐ A 569 ☐ B 1683 ☐ C 30003 ☐ D 31683

- 15** (R) Mali je Ivica svakog svakog ljeta u zadnjih sedam godina naučio jedan novi programski jezik. Svoja vrijedna iskustva sazeo 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	Eval (E)	Izvođenje (I)	Paradigma (P)	Tipovi (T)	y	i	Eval (E)	Izvođenje (I)	Paradigma (P)	Tipovi (T)	y
1	lijena	kompajler	imperativna	statička	0	5	striktna	interpreter	imperativna	statička	1
2	striktna	interpreter	deklarativna	dinamička	0	6	lijena	kompajler	hibridna	dinamička	1
3	lijena	kompajler	imperativna	dinamička	0	7	striktna	kompajler	hibridna	dinamička	1
4	lijena	interpreter	hibridna	statička	0						

Ovog ljeta Mali Ivica opet naučiti novi programski jezik, pa je odlučio izgraditi stablo odluke za klasifikaciju programskih jezika koje još nije naučio. Pomozite Malom Ivici te na gornjem skupu primjera primijenite 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?** (Korijski čvor je čvor na prvoj razini.)

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

- 16** (T) Model Bayesovog klasifikatora učimo tako da procijenimo njegove parametre. Za to koristimo procjenu navjeće izglednosti s Laplaceovim zaglađivanjem. **U kojim situacijama je važno koristiti Laplaceovo zaglađivanje?**

- ☐ A Kada je skup za učenje velik ☐ C Kada je skup za učenje malen
☐ B Kada je skup za ispitivanje malen ☐ D Kada je skup za provjeru malen

- 17** (P) Rešetkasti svijet sastoji se od pet ćelija (numerirane 1 do 5) koje su smještene slijeva udesno (ćelija 1 je prva, ćelija 5 je posljednja). Ćelija 5 je završna. Na ćelijama 1 do 4 robot može poduzeti jednu od tri akcije: $a_1 = \text{lijevo}$, $a_2 = \text{desno}$, $a_3 = \text{sagni-se-i-pokupi-bocu}$. Boce se nalaze na ćelijama 2 i 4. Okolina robotu dodjeljuje nagrade kako slijedi. Za prelazak na ćeliju 5 robot dobiva 10 bodova. Za saginjanje i skupljanje boce robot dobiva 5 bodova, no ako se sagne i pokuša pokupiti bocu na ćeliji koja nema bocu, robot dobiva -5 bodova. Za sve ostale akcije isporučuje se nagrada življenja od -1 bod. U ovom svijetu robot uči optimalnu politiku algoritmom q-učenja. Neka su $\gamma = 1$ i $\alpha = 0.25$. U nekom trenutku naučene q-vrijednosti su sljedeće: $q(1, a_1) = 0$, $q(2, a_1) = 1$, $q(3, a_1) = 0$, $q(4, a_1) = 1$, $q(1, a_2) = 1$, $q(2, a_2) = 1$, $q(3, a_2) = 1$, $q(4, a_2) = 2$, $q(1, a_3) = 2$, $q(2, a_3) = -1$, $q(3, a_3) = -2$, $q(4, a_3) = 3$. Robot se nalazi na ćeliji 3 i poduzima akciju a_2 . **Koja će biti nova vrijednost za $q(3, a_2)$?**

- ☐ A 2.75 ☐ B 1.5 ☐ C 3 ☐ D 1.25

6. Prirodom inspirirani optimizacijski algoritmi (3 pitanja)

- 18 (P) Genetičkim algoritmom traži se minimum funkcije $f(x) = (x - \frac{7}{8})^2$ na intervalu $[-3, 3.3]$. Pri tome se koriste 6-bitovni kromosomi. Označimo s \hat{x} najbolje rješenje koje GA može pronaći, a s x^* vrijednost za koju funkcija poprima minimum. **Koliko iznosi $|x^* - \hat{x}|$?**
- ☐ A 0.75 ☐ B 0.025 ☐ C 0.875 ☐ D 0.25
- 19 (R) Mravljim algoritmom uz parametre $\alpha = \beta = 1$ traži se put (hamiltonov ciklus) kroz graf koji se sastoji od 5 čvorova: A do E . Za bridove su poznate vrijednosti heurističkih informacija: $\eta_{AB} = 7$, $\eta_{AC} = 1$, $\eta_{BC} = 2$, $\eta_{BD} = 3$, $\eta_{CD} = 2$, $\eta_{DE} = 4$ i $\eta_{CE} = 3$. U nekoj iteraciji algoritma vrijednosti feromonskih tragova su: $\tau_{AB} = 1$, $\tau_{AC} = 3$, $\tau_{BC} = 2$, $\tau_{BD} = 2$, $\tau_{CD} = 1$, $\tau_{DE} = 2$ i $\tau_{CE} = 3$. Sve vrijednosti η i τ su ujedno i simetrične. Mrav konstrukciju puta započinje iz čvora A . Pretpostavite da će prilikom slučajnog odabira sljedećeg čvora mrav svaki puta odabrati onaj koji je najvjerojatniji (i u koji je dozvoljeno odnosno moguće prijeći). Odredite put koji će mrav konstruirati. **Koji je od sljedećih sljedova dio tog puta?**
- ☐ A $D \rightarrow C \rightarrow E$ ☐ B $B \rightarrow D \rightarrow C$ ☐ C $B \rightarrow C \rightarrow E$ ☐ D $E \rightarrow C \rightarrow A$
- 20 (T) Genetski algoritmi koriste operatore selekcije, križanja i mutacije te mehanizam elitizma kako bi pretražili prostor rješenja. **Što bi se dogodilo kada ne bismo koristili mehanizam elitizma?**
- ☐ A Algoritam bi mogao vratiti rješenje koje je lošije od nekog ranije generiranog rješenja
- ☐ B Povećala bi se prosječna vrijednost funkcije dobrote u populaciji
- ☐ C Porasla bi vjerojatnost da pretraga zapne u lokalnome optimumu
- ☐ D Algoritam bi generirao rješenja koja su neispravna (izvan intervala pretrage)

Uvod u umjetnu inteligenciju – pismeni ispit (2022./2023.) – 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 (4 pitanja)

- 1** (R) Neka su definirani skup stanja $S = \{a, b, c, d, e, f\}$ i funkcija sljedbenika $\text{succ}(a) = \{(b, 1), (c, 2)\}$, $\text{succ}(b) = \text{succ}(f) = \emptyset$, $\text{succ}(c) = \{(d, 3), (e, 5)\}$, $\text{succ}(d) = \{(b, 8), (f, 5)\}$ te $\text{succ}(e) = \{(d, 1), (f, 2)\}$. Heurističke vrijednosti čvorova neka su $h(a) = 7$, $h(b) = 4$, $h(c) = 6$, $h(d) = h(e) = 2$, $h(f) = 0$. Početno stanje neka je a , a ciljno f . Izvršite pretraživanje algoritmom A^* , bilježeći pritom sadržaje liste otvorenih čvorova O i zatvorenih čvorova C u svakom koraku algoritma (u nultom koraku $O = [(a, 0)]$ i $C = \emptyset$). **Koji su sadržaji listi O i C nakon četvrtog koraka izvođenja algoritma A^* ?**

- ☐ A $O = [(f, 9)]$, $C = \{(a, 0), (b, 1), (c, 2), (d, 5)\}$
☐ B $O = [(e, 7), (f, 10)]$, $C = \{(a, 0), (c, 2), (d, 5)\}$
☐ C $O = [(e, 7), (f, 10)]$, $C = \{(a, 0), (b, 1), (c, 2), (d, 5)\}$
☐ D Algoritam ne dostiže četvrti korak

- 2** (T) Algoritam minimax s dubinskim ograničenjem uobičajeno je izveden pomoću dvije, međusobno rekurzivne funkcije. Neka su to funkcije $\text{maxValue}(s, d)$ i $\text{minValue}(s, d)$, gdje je s stanje igre, a d je brojač razine u stablu igre. Početni poziv algoritma je $\text{maxValue}(s_0, k)$, gdje je s_0 trenutno stanje igre, a k je dubinsko ograničenje. Funkcija $\text{maxValue}(s, d)$ koristi petlju s iteratorom $t \in \text{succ}(s)$ kako bi izračunala minimax-vrijednost m , koja je izvan petlje inicijalizirana na $-\infty$. **Kako glasi tijelo petlje u funkciji maxValue ?**

- ☐ A $m \leftarrow \max(m, \text{minValue}(t, d - 1))$ ☐ C $m \leftarrow \max(m, \text{maxValue}(t, d + 1))$
☐ B $m \leftarrow \min(m, \text{minValue}(t, d + 1))$ ☐ D $m \leftarrow \min(m, \text{maxValue}(t, d - 1))$

- 3** (P) Rješavamo problem pretraživanja prostora stanja za koji vrijedi $m = 10$, $d = 7$ i $b = 4$. Koristimo pretraživanje u širinu (BFS) i pretraživanje u dubinu (DFS). Za oba algoritma razmotrite najgori slučaj vremenske složenosti. **Koliko će u takvom slučaju algoritam DFS generirati više čvorova od algoritma BFS?**

- ☐ A 1376256 ☐ B 1310640 ☐ C 1310716 ☐ D 1376176

- 4** (P) Razmatramo dva igrača algoritma: jedan s heuristikom h_1 i dubinom pretraživanja d_1 te drugi s heuristikom h_2 i dubinom pretraživanja d_2 . Oba algoritma koriste strategiju minimax. **Kada će prvi algoritam češće pobjeđivati drugi algoritam?**

- ☐ A Ako $d_1 = d_2$ i $h_1 < h_2$ ☐ B Ako $d_1 > d_2$ ☐ C Ako $h_1 = h_2$ i $d_1 > d_2$ ☐ D Ako $h_1 = h_2$ i $d_1 < d_2$

2. Prikazivanje znanja i automatsko zaključivanje (4 pitanja)

- 5** (T) Relacija logičke posljedice između formula F i G definirana je na temelju vrijednosti istinitosti formula F i G kroz sve moguće interpretacije tih formula. Dokazivanje logičke posljedice od praktičnog je značaja za umjetnu inteligenciju. **Na koji način dokazujemo logičku posljedicu u FOL?**

- ☐ A Pretvorbom u klauzalni oblik i skolemizacijom
☐ B Pomoću ispravnih i potpunih pravila zaključivanja
☐ C Primjenom izravne rezolucije i faktORIZACIJE
☐ D Konstrukcijom tablice istinitosti i provjerom modela

6 (R) Dani su FOL atomi $P(f(x), y, g(w, b))$ i $P(y, f(g(a, z)), x)$, gdje je P predikat, a i b su konstante, a f i g su funkcije. **Što je najopćenitiji zajednički unifikator ovih dvaju atoma?**

- ☐ A $P(f(x), f(g(x, y)), g(a, b))$ ☐ C $P(f(g(a, b)), f(g(a, b)), g(a, b))$
☐ B $P(f(g(a, b)), f(g(a, z)), g(w, b))$ ☐ D Atome nije moguće unificirati

7 (P) Ispravnost je važno svojstvo pravila zaključivanja. **Koje je od sljedećih pravila zaključivanja ispravno?**

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

8 (R) Zadane su sljedeće dvije premise:

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

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

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

3. Logičko programiranje i ekspertni sustavi (2 pitanja)

9 (T) Izvođenje logičkih programa u Prologu odgovara dokazivanju teorema rezolucijom opovrgavanjem. Stanje izvođenja u Prologu definirano je stogom, koji početno sadrži atom upita. **Što sadrži Prologov stog u nastavku izvođenja programa?**

- ☐ A Atome iz glave logičkih pravila ☐ C Činjenice iz logičkog programa
☐ B Atome dviju roditeljskih klauzula ☐ D Atome iz tijela logičkih pravila

10 (T) Kao i sustavi formalne logike, ekspertni se sustavi temelje na formalnom prikazu znanja i izvođenju novog znanja primjenom pravila zaključivanja. Međutim, između tih sustava postoje razlike. **Po čemu se ekspertni sustavi razlikuju od općenitih sustava formalne logike?**

- ☐ A Znanje je prikazano u obliku činjenica i pravila, a ne proizvoljnih logičkih formula
☐ B Znanje je prikazano u obliku ako-onda pravila, a ne Hornovih ili definitnih klauzula
☐ C Za dokazivanje se koriste logička pravila koja su potpuna, a ne samo ispravna
☐ D Za dokazivanje se koristi rezolucija opovrgavanjem, a ne izravna rezolucija

4. Modeliranje neizvjesnosti (2 pitanja)

11 (T) Bayesovo pravilo možemo koristiti za zaključivanje abdukcijom. **Što to znači?**

- ☐ A Na temelju ostvarenog uzroka zaključujemo o hipotezi
☐ B Na temelju ostvarenog uzroka zaključujemo o dokazima
☐ C Na temelju opažene hipoteze zaključujemo o dokazu
☐ D Na temelju opaženih dokaza zaključujemo o uzroku

12 Nad univerzalnim skupom $\{a, b, c, d\}$ zadani su neizraziti skupovi $X = \{0.1/a + 0/b + 0.3/c + 1/d\}$, $Y = \{0.5/a + 0.4/b + 0/c + 0.2/d\}$ i $Z = \{1/a + 0/b + 0/c + 0.5/d\}$. Koristeći Zadehove operatore odredite neizraziti skup koji odgovara jezičnome izrazu “Ne Z ili (X i Y)”.

- ☐ A $\{0/a + 0.4/b + 0.3/c + 0.5/d\}$ ☐ C $\{0.9/a + 0.4/b + 0/c + 0/d\}$
☐ B $\{0.5/a + 0/b + 0/c + 0/d\}$ ☐ D $\{0.1/a + 1/b + 1/c + 0.5/d\}$

5. Strojno učenje, umjetne neuronske mreže i podržano učenje (5 pitanja)

- 13** (P) Gradimo naivan Bayesov klasifikator za klasifikaciju poruka Twittera (tvitova) prema sentimentu. Svaki tvit želimo klasificirati u jednu od tri klase: pozitivan, negativan ili neutralan. Svaki tvit sastoji se od najviše 280 riječi i prikazujemo ga kao jedan binaran vektor značajki. Vektor se sastoji od 5000 značajki, i svaka značajka odgovara jednoj od 5000 riječi iz našeg vokabulara. Ako je značajka postavljena na 1, to znači da se dotična riječ pojavila u tvitu, inače je značajka postavljena na 0. Npr., ako je $x_{42} = 1$, onda to znači da se 42. riječ iz našeg vokabulara pojavila u tvitu. Učenje ovog modela svodi se na procjenu apriornih vjerojatnosti i izglednosti klase na temelju označenog skupa podataka. **Koliko ćemo ukupno vjerojatnosti trebati procijeniti za ovaj model?**

☐ A 569 ☐ B 30003 ☐ C 31683 ☐ D 1683

- 14** (T) Model Bayesovog klasifikatora učimo tako da procijenimo njegove parametre. Za to koristimo procjenu navjeće izglednosti s Laplaceovim zaglađivanjem. **U kojim situacijama je važno koristiti Laplaceovo zaglađivanje?**

☐ A Kada je skup za provjeru malen ☐ C Kada je skup za učenje velik
☐ B Kada je skup za učenje malen ☐ D Kada je skup za ispitivanje malen

- 15** (R) Mali je Ivica svakog svakog ljeta u zadnjih sedam godina naučio jedan novi programski jezik. Svoja vrijedna iskustva sazeo 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	Eval (E)	Izvođenje (I)	Paradigma (P)	Tipovi (T)	y	i	Eval (E)	Izvođenje (I)	Paradigma (P)	Tipovi (T)	y
1	lijena	kompajler	imperativna	statička	0	5	striktna	interpreter	imperativna	statička	1
2	striktna	interpreter	deklarativna	dinamička	0	6	lijena	kompajler	hibridna	dinamička	1
3	lijena	kompajler	imperativna	dinamička	0	7	striktna	kompajler	hibridna	dinamička	1
4	lijena	interpreter	hibridna	statička	0						

Ovog ljeta Mali Ivica opet naučiti novi programski jezik, 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?** (Korijenski čvor je čvor na prvoj razini.)

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

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

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

Tim 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. Neke je početni vektor težina $[w_2, w_1, w_0] = [8, 4, -12]$. **Koliko će se puta provesti korigiranje vektora težina i koja je njegova konačna vrijednost?**

☐ A 3 puta, $[7, 2, -12]$ ☐ B 2 puta, $[5, 2, -14]$ ☐ C 2 puta, $[10, 4, -28]$ ☐ D 3 puta, $[8, -1, -11]$

- 17** (P) Rešetkasti svijet sastoji se od pet ćelija (numerirane 1 do 5) koje su smještene slijeva udesno (ćelija 1 je prva, ćelija 5 je posljednja). Ćelija 5 je završna. Na ćelijama 1 do 4 robot može poduzeti jednu od tri akcije: $a_1 = \text{lijevo}$, $a_2 = \text{desno}$, $a_3 = \text{sagni-se-i-pokupi-bocu}$. Boce se nalaze na ćelijama 2 i 4. Okolina robotu dodjeljuje nagrade kako slijedi. Za prelazak na ćeliju 5 robot dobiva 10 bodova. Za saginjanje i skupljanje boce robot dobiva 5 bodova, no ako se sagne i pokuša pokupiti bocu na ćeliji koja nema bocu, robot dobiva -5 bodova. Za sve ostale akcije isporučuje se nagrada življenja od -1 bod. U ovom svijetu robot uči optimalnu politiku algoritmom q-učenja. Neke su $\gamma = 1$ i $\alpha = 0.25$. U nekom trenutku naučene q-vrijednosti su sljedeće: $q(1, a_1) = 0$, $q(2, a_1) = 1$, $q(3, a_1) = 0$, $q(4, a_1) = 1$, $q(1, a_2) = 1$, $q(2, a_2) = 1$, $q(3, a_2) = 1$, $q(4, a_2) = 2$, $q(1, a_3) = 2$, $q(2, a_3) = -1$, $q(3, a_3) = -2$, $q(4, a_3) = 3$. Robot se nalazi na ćeliji 3 i poduzima akciju a_3 . **Koja će biti nova vrijednost za $q(3, a_3)$?**

☐ A 2 ☐ B -1 ☐ C 1.25 ☐ D -2.5

6. Prirodom inspirirani optimizacijski algoritmi (3 pitanja)

- 18 (R) Mravljim algoritmom uz parametre $\alpha = \beta = 1$ traži se put (hamiltonov ciklus) kroz graf koji se sastoji od 5 čvorova: A do E . Za bridove su poznate vrijednosti heurističkih informacija: $\eta_{AB} = 7$, $\eta_{AC} = 1$, $\eta_{BC} = 2$, $\eta_{BD} = 3$, $\eta_{CD} = 2$, $\eta_{DE} = 4$ i $\eta_{CE} = 3$. U nekoj iteraciji algoritma vrijednosti feromonskih tragova su: $\tau_{AB} = 1$, $\tau_{AC} = 3$, $\tau_{BC} = 2$, $\tau_{BD} = 2$, $\tau_{CD} = 1$, $\tau_{DE} = 2$ i $\tau_{CE} = 3$. Sve vrijednosti η i τ su ujedno i simetrične. Mrav konstrukciju puta započinje iz čvora A . Pretpostavite da će prilikom slučajnog odabira sljedećeg čvora mrav svaki puta odabrati onaj koji je najvjerojatniji (i u koji je dozvoljeno odnosno moguće prijeći). Odredite put koji će mrav konstruirati. **Koji je od sljedećih sljedova dio tog puta?**

☐ A $E \rightarrow C \rightarrow A$ ☐ B $B \rightarrow C \rightarrow E$ ☐ C $B \rightarrow D \rightarrow C$ ☐ D $D \rightarrow C \rightarrow E$

- 19 (T) Genetski algoritmi koriste operatore selekcije, križanja i mutacije te mehanizam elitizma kako bi pretražili prostor rješenja. **Što bi se dogodilo kada ne bismo koristili mehanizam elitizma?**

- ☐ A Povećala bi se prosječna vrijednost funkcije dobrote u populaciji
☐ B Algoritam bi mogao vratiti rješenje koje je lošije od nekog ranije generiranog rješenja
☐ C Algoritam bi generirao rješenja koja su neispravna (izvan intervala pretrage)
☐ D Porasla bi vjerojatnost da pretraga zapne u lokalnome optimumu

- 20 (P) Genetičkim algoritmom traži se minimum funkcije $f(x) = (x - \frac{11}{8})^2$ na intervalu $[-2.5, 3.8]$. Pri tome se koriste 6-bitovni kromosomi. Označimo s \hat{x} najbolje rješenje koje GA može pronaći, a s x^* vrijednost za koju funkcija poprima minimum. **Koliko iznosi $|x^* - \hat{x}|$?**

☐ A 0.3 ☐ B 0.875 ☐ C 0.025 ☐ D 0.125

Uvod u umjetnu inteligenciju – pismeni ispit (2022./2023.) – 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 (4 pitanja)

- 1** (P) Rješavamo problem pretraživanja prostora stanja za koji vrijedi $m = 10$, $d = 7$ i $b = 4$. Koristimo pretraživanje u širinu (BFS) i pretraživanje u dubinu (DFS). Za oba algoritma razmotrite najgori slučaj vremenske složenosti. **Koliko će u takvom slučaju algoritam DFS generirati više čvorova od algoritma BFS?**

☐ A 1376256 ☐ B 1376176 ☐ C 1310716 ☐ D 1310640

- 2** (R) Neka su definirani skup stanja $S = \{a, b, c, d, e, f\}$ i funkcija sljedbenika $\text{succ}(a) = \{(b, 1), (c, 2)\}$, $\text{succ}(b) = \text{succ}(f) = \emptyset$, $\text{succ}(c) = \{(d, 3), (e, 5)\}$, $\text{succ}(d) = \{(b, 8), (f, 5)\}$ te $\text{succ}(e) = \{(d, 1), (f, 2)\}$. Heurističke vrijednosti čvorova neka su $h(a) = 7$, $h(b) = 4$, $h(c) = 6$, $h(d) = h(e) = 2$, $h(f) = 0$. Početno stanje neka je a , a ciljno f . Izvršite pretraživanje algoritmom A^* , bilježeći pritom sadržaje liste otvorenih čvorova O i zatvorenih čvorova C u svakom koraku algoritma (u nultom koraku $O = [(a, 0)]$ i $C = \emptyset$). **Koji su sadržaji listi O i C nakon petog koraka izvođenja algoritma A^* ?**

☐ A Algoritam ne dostiže peti korak
☐ B $O = [(e, 7), (f, 10), (b, 13)]$, $C = \{(a, 0), (b, 1), (c, 2), (d, 5)\}$
☐ C $O = [(f, 10)]$, $C = \{(a, 0), (b, 1), (c, 2), (d, 5), (e, 7)\}$
☐ D $O = [(f, 9)]$, $C = \{(a, 0), (b, 1), (c, 2), (d, 5), (e, 7)\}$

- 3** (T) Algoritam minimax s dubinskim ograničenjem uobičajeno je izveden pomoću dvije, međusobno rekurzivne funkcije. Neka su to funkcije $\text{maxValue}(s, d)$ i $\text{minValue}(s, d)$, gdje je s stanje igre, a d je brojač razine u stablu igre. Početni poziv algoritma je $\text{maxValue}(s_0, k)$, gdje je s_0 trenutno stanje igre, a k je dubinsko ograničenje. Funkcija $\text{minValue}(s, d)$ koristi petlju s iteratorom $t \in \text{succ}(s)$ kako bi izračunala minimax-vrijednost m , koja je izvan petlje inicijalizirana na $+\infty$. **Kako glasi tijelo petlje u funkciji minValue ?**

☐ A $m \leftarrow \min(m, \text{minValue}(t, d + 1))$ ☐ C $m \leftarrow \max(m, \text{minValue}(t, d - 1))$
☐ B $m \leftarrow \max(m, \text{maxValue}(t, d + 1))$ ☐ D $m \leftarrow \min(m, \text{maxValue}(t, d - 1))$

- 4** (P) Razmatramo dva igrača algoritma: jedan s heuristikom h_1 i dubinom pretraživanja d_1 te drugi s heuristikom h_2 i dubinom pretraživanja d_2 . Oba algoritma koriste strategiju minimax. **Kada će prvi algoritam češće pobjeđivati drugi algoritam?**

☐ A Ako $h_1 = h_2$ i $d_1 > d_2$ ☐ B Ako $d_1 = d_2$ i $h_1 < h_2$ ☐ C Ako $h_1 = h_2$ i $d_1 < d_2$ ☐ D Ako $d_1 > d_2$

2. Prikazivanje znanja i automatsko zaključivanje (4 pitanja)

- 5** (T) Relacija logičke posljedice između formula F i G definirana je na temelju vrijednosti istinitosti formula F i G kroz sve moguće interpretacije tih formula. Dokazivanje logičke posljedice od praktičnog je značaja za umjetnu inteligenciju. **Na koji način dokazujemo logičku posljedicu u FOL?**

☐ A Pomoću ispravnih i potpunih pravila zaključivanja
☐ B Pretvorbom u klauzalni oblik i skolemizacijom
☐ C Primjenom izravne rezolucije i faktORIZACIJE
☐ D Konstrukcijom tablice istinitosti i provjerom modela

- 6 (R) Zadane su sljedeće dvije premise:

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

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

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

- 7 (R) Dani su FOL atomi $P(f(x), y, g(w, b))$ i $P(y, f(g(a, z)), x)$, gdje je P predikat, a i b su konstante, a f i g su funkcije. Što je najopćenitiji zajednički unifikator ovih dvaju atoma?

- ☐ A $P(f(g(a, b)), f(g(a, z)), g(w, b))$ ☐ C $P(f(g(a, b)), f(g(a, b)), g(a, b))$
☐ B Atome nije moguće unificirati ☐ D $P(f(y), f(g(a, z)), g(w, b))$

- 8 (P) Ispravnost je važno svojstvo pravila zaključivanja. Koje je od sljedećih pravila zaključivanja ispravno?

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

3. Logičko programiranje i ekspertni sustavi (2 pitanja)

- 9 (T) Izvođenje logičkih programa u Prologu odgovara dokazivanju teorema rezolucijom opovrgavanjem. Stanje izvođenja u Prologu definirano je stogom, koji početno sadrži atom upita. Što sadrži Prologov stog u nastavku izvođenja programa?

- ☐ A Atome iz tijela logičkih pravila ☐ C Činjenice iz logičkog programa
☐ B Atome iz glave logičkih pravila ☐ D Atome dviju roditeljskih klauzula

- 10 (T) Kao i sustavi formalne logike, ekspertni se sustavi temelje na formalnom prikazu znanja i izvođenju novog znanja primjenom pravila zaključivanja. Međutim, između tih sustava postoje razlike. Po čemu se ekspertni sustavi razlikuju od općenitih sustava formalne logike?

- ☐ A Za dokazivanje se koriste logička pravila koja su potpuna, a ne samo ispravna
☐ B Znanje je prikazano u obliku činjenica i pravila, a ne proizvoljnih logičkih formula
☐ C Za dokazivanje se koristi rezolucija opovrgavanjem, a ne izravna rezolucija
☐ D Znanje je prikazano u obliku ako-onda pravila, a ne Hornovih ili definitnih klauzula

4. Modeliranje neizvjesnosti (2 pitanja)

- 11 Nad univerzalnim skupom $\{a, b, c, d\}$ zadani su neizraziti skupovi $X = \{0.1/a + 0/b + 0.3/c + 1/d\}$, $Y = \{0.5/a + 0.4/b + 0/c + 0.2/d\}$ i $Z = \{1/a + 0/b + 0/c + 0.5/d\}$. Koristeći Zadehove operatore odredite neizraziti skup koji odgovara jezičnome izrazu "Ne Z i (X ili Y)".

- ☐ A $\{0/a + 0.4/b + 0.3/c + 0.5/d\}$ ☐ C $\{1/a + 1/b + 0.7/c + 0.5/d\}$
☐ B $\{0.9/a + 1.0/b + 0.7/c + 0.2/d\}$ ☐ D $\{0.5/a + 0/b + 0.3/c + 1/d\}$

- 12 (T) Bayesovo pravilo možemo koristiti za zaključivanje abdukcijom. Što to znači?

- ☐ A Na temelju ostvarenog uzroka zaključujemo o hipotezi
☐ B Na temelju opažene hipoteze zaključujemo o dokazu
☐ C Na temelju ostvarenog uzroka zaključujemo o dokazima
☐ D Na temelju opaženih dokaza zaključujemo o uzroku

5. Strojno učenje, umjetne neuronske mreže i podržano učenje (5 pitanja)

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

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

Tim 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 $[w_2, w_1, w_0] = [8, 4, -12]$. **Koliko će se puta provesti korigiranje vektora težina i koja je njegova konačna vrijednost?**

- ☐ A 2 puta, $[5, 2, -14]$ ☐ B 2 puta, $[10, 4, -28]$ ☐ C 3 puta, $[8, -1, -11]$ ☐ D 3 puta, $[7, 2, -12]$

- 14** (P) Gradimo naivan Bayesov klasifikator za klasifikaciju poruka Twittera (tvitova) prema sentimentu. Svaki tvit želimo klasificirati u jednu od tri klase: pozitivan, negativan ili neutralan. Svaki tvit sastoji se od najviše 280 riječi i prikazujemo ga kao jedan binaran vektor značajki. Vektor se sastoji od 5000 značajki, i svaka značajka odgovara jednoj od 5000 riječi iz našeg vokabulara. Ako je značajka postavljena na 1, to znači da se dotična riječ pojavila u tvitu, inače je značajka postavljena na 0. Npr., ako je $x_{42} = 1$, onda to znači da se 42. riječ iz našeg vokabulara pojavila u tvitu. Učenje ovog modela svodi se na procjenu apriornih vjerojatnosti i izglednosti klase na temelju označenog skupa podataka. **Koliko ćemo ukupno vjerojatnosti trebati procijeniti za ovaj model?**

- ☐ A 30003 ☐ B 569 ☐ C 1683 ☐ D 31683

- 15** (T) Model Bayesovog klasifikatora učimo tako da procijenimo njegove parametre. Za to koristimo procjenu navjeće izglednosti s Laplaceovim zaglađivanjem. **U kojim situacijama je važno koristiti Laplaceovo zaglađivanje?**

- ☐ A Kada je skup za učenje malen ☐ C Kada je skup za učenje velik
☐ B Kada je skup za provjeru malen ☐ D Kada je skup za ispitivanje malen

- 16** (R) Mali je Ivica svakog svakog ljeta u zadnjih sedam godina naučio jedan novi programski jezik. Svoja vrijedna iskustva sazeo 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	Eval (E)	Izvođenje (I)	Paradigma (P)	Tipovi (T)	y	i	Eval (E)	Izvođenje (I)	Paradigma (P)	Tipovi (T)	y
1	lijena	kompajler	imperativna	statička	0	5	striktna	interpreter	imperativna	statička	1
2	striktna	interpreter	deklarativna	dinamička	0	6	lijena	kompajler	hibridna	dinamička	1
3	lijena	kompajler	imperativna	dinamička	0	7	striktna	kompajler	hibridna	dinamička	1
4	lijena	interpreter	hibridna	statička	0						

Ovog ljeta Mali Ivica opet naučiti novi programski jezik, 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?** (Korijski čvor je čvor na prvoj razini.)

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

- 17** (P) Rešetkasti svijet sastoji se od pet ćelija (numerirane 1 do 5) koje su smještene slijeva udesno (ćelija 1 je prva, ćelija 5 je posljednja). Ćelija 5 je završna. Na ćelijama 1 do 4 robot može poduzeti jednu od tri akcije: $a_1 = \text{lijevo}$, $a_2 = \text{desno}$, $a_3 = \text{sagni-se-i-pokupi-bocu}$. Boce se nalaze na ćelijama 2 i 4. Okolina robotu dodjeljuje nagrade kako slijedi. Za prelazak na ćeliju 5 robot dobiva 10 bodova. Za saginjanje i skupljanje boce robot dobiva 5 bodova, no ako se sagne i pokuša pokupiti bocu na ćeliji koja nema bocu, robot dobiva -5 bodova. Za sve ostale akcije isporučuje se nagrada življenja od -1 bod. U ovom svijetu robot uči optimalnu politiku algoritmom q-učenja. Neka su $\gamma = 1$ i $\alpha = 0.25$. U nekom trenutku naučene q-vrijednosti su sljedeće: $q(1, a_1) = 0$, $q(2, a_1) = 1$, $q(3, a_1) = 0$, $q(4, a_1) = 1$, $q(1, a_2) = 1$, $q(2, a_2) = 1$, $q(3, a_2) = 1$, $q(4, a_2) = 2$, $q(1, a_3) = 2$, $q(2, a_3) = -1$, $q(3, a_3) = -2$, $q(4, a_3) = 3$. Robot se nalazi na ćeliji 4 i poduzima akciju a_3 . **Koja će biti nova vrijednost za $q(3, a_3)$?**

- ☐ A 4.25 ☐ B 2.25 ☐ C 3 ☐ D 8

6. Prirodom inspirirani optimizacijski algoritmi (3 pitanja)

- 18 (R) Mravljinim algoritmom uz parametre $\alpha = \beta = 1$ traži se put (hamiltonov ciklus) kroz graf koji se sastoji od 5 čvorova: A do E . Za bridove su poznate vrijednosti heurističkih informacija: $\eta_{AB} = 7$, $\eta_{AC} = 1$, $\eta_{BC} = 2$, $\eta_{BD} = 3$, $\eta_{CD} = 2$, $\eta_{DE} = 4$ i $\eta_{CE} = 3$. U nekoj iteraciji algoritma vrijednosti feromonskih tragova su: $\tau_{AB} = 1$, $\tau_{AC} = 3$, $\tau_{BC} = 2$, $\tau_{BD} = 2$, $\tau_{CD} = 1$, $\tau_{DE} = 2$ i $\tau_{CE} = 3$. Sve vrijednosti η i τ su ujedno i simetrične. Mrav konstrukciju puta započinje iz čvora A . Pretpostavite da će prilikom slučajnog odabira sljedećeg čvora mrav svaki puta odabrati onaj koji je najvjerojatniji (i u koji je dozvoljeno odnosno moguće prijeći). Odredite put koji će mrav konstruirati. **Koji je od sljedećih sljedova dio tog puta?**

☐ A $B \rightarrow C \rightarrow E$ ☐ B $E \rightarrow C \rightarrow A$ ☐ C $D \rightarrow C \rightarrow E$ ☐ D $B \rightarrow D \rightarrow C$

- 19 (R) Genetskim algoritmom traži se maksimum funkcije $f(x, y) = 3x^2 + 5y^2$. Svaka varijabla kodira se s 4 bita, pri čemu se x pretražuje iz intervala $[-4, 11]$, a y iz intervala $[-6, 9]$. U kromosomu najprije dolaze bitovi koji kodiraju x pa potom bitovi koji kodiraju y . Za odabir roditelja koristi se selekcija proporcionalna dobroti (engl. *roulette wheel selection*). Populacija se sastoji od sljedeće četiri jedinke:

$$K_1 = 10110011, \quad K_2 = 11010101, \quad K_3 = 00101010, \quad K_4 = 00110001$$

Koristi se križanje s jednom točkom prijeloma (točno na polovici kromosoma). Pretpostavite da se u trenutačnoj iteraciji algoritma slučajno odaberu baš dvije najmanje vjerojatne jedinke te da pri izgradnji djece mutacija igrom slučaja ne promijeni niti jedan bit. Označimo s F zbroj vjerojatnosti odabira odabranih jedinki, a s G dobrotu djeteta koje će biti umetnuto u novu populaciju (ako križanjem/mutacijom nastaje više djece, samo najbolje dijete ide dalje). **Što od sljedećega vrijedi za F i G ?**

☐ A $F < 0.5$, $90 < G < 140$ ☐ B $F > 0.3$, $70 < G < 110$ ☐ C $F > 0.4$, $G > 90$ ☐ D $F < 0.4$, $G < 100$

- 20 (P) Genetičkim algoritmom traži se minimum funkcije $f(x) = (x - \frac{11}{8})^2$ na intervalu $[-2.5, 3.8]$. Pri tome se koriste 6-bitovni kromosomi. Označimo s \hat{x} najbolje rješenje koje GA može pronaći, a s x^* vrijednost za koju funkcija poprima minimum. **Koliko iznosi $|x^* - \hat{x}|$?**

☐ A 0.25 ☐ B 0.75 ☐ C 0.625 ☐ D 0.025

Uvod u umjetnu inteligenciju – pismeni ispit (2022./2023.) – 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 (4 pitanja)

- 1** (T) Algoritam minimax s dubinskim ograničenjem uobičajeno je izveden pomoću dvije, međusobno rekurzivne funkcije. Neka su to funkcije $\text{maxValue}(s, d)$ i $\text{minValue}(s, d)$, gdje je s stanje igre, a d je brojač razine u stablu igre. Početni poziv algoritma je $\text{maxValue}(s_0, k)$, gdje je s_0 trenutačno stanje igre, a k je dubinsko ograničenje. Funkcija $\text{maxValue}(s, d)$ koristi petlju s iteratorom $t \in \text{succ}(s)$ kako bi izračunala minimax-vrijednost m , koja je izvan petlje inicijalizirana na $-\infty$. **Kako glasi tijelo petlje u funkciji maxValue ?**

- ☐ A $m \leftarrow \max(m, \text{minValue}(t, d - 1))$ ☐ C $m \leftarrow \max(m, \text{maxValue}(t, d + 1))$
☐ B $m \leftarrow \min(m, \text{minValue}(t, d + 1))$ ☐ D $m \leftarrow \min(m, \text{maxValue}(t, d - 1))$

- 2** (P) Rješavamo problem pretraživanja prostora stanja za koji vrijedi $m = 9$, $d = 6$ i $b = 4$. Koristimo pretraživanje u širinu (BFS) i pretraživanje u dubinu (DFS). Za oba algoritma razmotrite najgori slučaj vremenske složenosti. **Koliko će u takvom slučaju algoritam DFS generirati više čvorova od algoritma BFS?**

- ☐ A 327676 ☐ B 327600 ☐ C 343984 ☐ D 344064

- 3** (P) Razmatramo dva igrača algoritma: jedan s heuristikom h_1 i dubinom pretraživanja d_1 te drugi s heuristikom h_2 i dubinom pretraživanja d_2 . Oba algoritma koriste strategiju minimax. **Kada će prvi algoritam češće pobjeđivati drugi algoritam?**

- ☐ A Ako $h_2 > h_1$ ☐ B Ako $d_1 > d_2$ ☐ C Ako $h_1 = h_2$ i $d_1 > d_2$ ☐ D Ako $h_1 > h_2$

- 4** (R) Neka su definirani skup stanja $S = \{a, b, c, d, e, f\}$ i funkcija sljedbenika $\text{succ}(a) = \{(b, 1), (c, 2)\}$, $\text{succ}(b) = \text{succ}(f) = \emptyset$, $\text{succ}(c) = \{(d, 3), (e, 5)\}$, $\text{succ}(d) = \{(b, 8), (f, 5)\}$ te $\text{succ}(e) = \{(d, 1), (f, 2)\}$. Heurističke vrijednosti čvorova neka su $h(a) = 7$, $h(b) = 4$, $h(c) = 6$, $h(d) = h(e) = 2$, $h(f) = 0$. Početno stanje neka je a , a ciljno f . Izvršite pretraživanje algoritmom A^* , bilježeći pritom sadržaje liste otvorenih čvorova O i zatvorenih čvorova C u svakom koraku algoritma (u nultom koraku $O = [(a, 0)]$ i $C = \emptyset$). **Koji su sadržaji listi O i C nakon petog koraka izvođenja algoritma A^* ?**

- ☐ A Algoritam ne dostiže peti korak
☐ B $O = [(e, 7), (f, 10)]$, $C = \{(a, 0), (b, 1), (c, 2), (d, 5)\}$
☐ C $O = [(f, 10)]$, $C = \{(a, 0), (b, 1), (c, 2), (d, 5), (e, 7)\}$
☐ D $O = [(f, 9)]$, $C = \{(a, 0), (b, 1), (c, 2), (d, 5), (e, 7)\}$

2. Prikazivanje znanja i automatsko zaključivanje (4 pitanja)

- 5** (R) Dani su FOL atomi $P(f(x), y, g(w, f(z)))$ i $P(y, f(g(a, z)), x)$, gdje je P predikat, a a je konstanta, a f i g su funkcije. **Što je najopćenitiji zajednički unifikator ovih dvaju atoma?**

- ☐ A $P(f(g(a, b)), f(g(a, b)), g(a, b))$ ☐ C $P(f(x), f(g(x, y)), g(a, b))$
☐ B $P(f(g(a, b)), f(g(a, z)), g(w, b))$ ☐ D Atome nije moguće unificirati

6 (T) Relacija logičke posljedice između formula F i G definirana je na temelju vrijednosti istinitosti formula F i G kroz sve moguće interpretacije tih formula. Dokazivanje logičke posljedice od praktičnog je značaja za umjetnu inteligenciju. **Na koji način dokazujemo logičku posljedicu u FOL?**

- ☐ A Pretvorbom u klauzalni oblik i skolemizacijom
- ☐ B Konstrukcijom tablice istinitosti i provjerom modela
- ☐ C Pomoću ispravnih i potpunih pravila zaključivanja
- ☐ D Primjenom izravne rezolucije i faktorizacije

7 (P) Ispravnost je važno svojstvo pravila zaključivanja. **Koje je od sljedećih pravila zaključivanja ispravno?**

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

8 (R) Zadane su sljedeće dvije premise:

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

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

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

3. Logičko programiranje i ekspertni sustavi (2 pitanja)

9 (T) Kao i sustavi formalne logike, ekspertni se sustavi temelje na formalnom prikazu znanja i izvođenju novog znanja primjenom pravila zaključivanja. Međutim, između tih sustava postoje razlike. **Po čemu se ekspertni sustavi razlikuju od općenitih sustava formalne logike?**

- ☐ A Za dokazivanje se koriste logička pravila koja su potpuna, a ne samo ispravna
- ☐ B Za dokazivanje se koristi rezolucija opovrgavanjem, a ne izravna rezolucija
- ☐ C Znanje je prikazano u obliku činjenica i pravila, a ne proizvoljnih logičkih formula
- ☐ D Znanje je prikazano u obliku ako-onda pravila, a ne Hornovih ili definitnih klauzula

10 (T) Izvođenje logičkih programa u Prologu odgovara dokazivanju teorema rezolucijom opovrgavanjem. Stanje izvođenja u Prologu definirano je stogom, koji početno sadrži atom upita. **Što sadrži Prologov stog u nastavku izvođenja programa?**

- ☐ A Činjenice iz logičkog programa ☐ C Atome iz glave logičkih pravila
☐ B Atome iz tijela logičkih pravila ☐ D Atome dviju roditeljskih klauzula

4. Modeliranje neizvjesnosti (2 pitanja)

11 (R) Nad univerzalnim skupom $\{a, b, c, d\}$ zadani su neizraziti skupovi $X = \{0.1/a + 0/b + 0.3/c + 1/d\}$, $Y = \{0.5/a + 0.4/b + 0/c + 0.2/d\}$ i $Z = \{1/a + 0/b + 0/c + 0.5/d\}$. Koristeći Zadehove operatore izvodimo neizraziti skup X tako da on odgovara jezičnome izrazu " $Ne X$ ili (Y i Z)". **Kako glasi neizraziti skup X ?**

- ☐ A $\{0.5/a + 0/b + 0.3/c + 1/d\}$ ☐ C $\{0.9/a + 1.0/b + 0.7/c + 0.2/d\}$
☐ B $\{0.9/a + 0.4/b + 0/c + 0/d\}$ ☐ D $\{0.5/a + 0/b + 0/c + 0/d\}$

12 (T) Bayesovo pravilo možemo koristiti za zaključivanje abdukcijom. **Što to znači?**

- ☐ A Na temelju opažene hipoteze zaključujemo o dokazu
- ☐ B Na temelju opaženih dokaza zaključujemo o uzroku
- ☐ C Na temelju ostvarenog uzroka zaključujemo o dokazima
- ☐ D Na temelju ostvarenog uzroka zaključujemo o hipotezi

5. Strojno učenje, umjetne neuronske mreže i podržano učenje (5 pitanja)

- 13** (P) Gradimo naivan Bayesov klasifikator za klasifikaciju poruka Twittera (tvitova) prema sentimentu. Svaki tvit želimo klasificirati u jednu od tri klase: pozitivan, negativan ili neutralan. Svaki tvit sastoji se od najviše 280 riječi i prikazujemo ga kao jedan binaran vektor značajki. Vektor se sastoji od 5000 značajki, i svaka značajka odgovara jednoj od 5000 riječi iz našeg vokabulara. Ako je značajka postavljena na 1, to znači da se dotična riječ pojavila u tvitu, inače je značajka postavljena na 0. Npr., ako je $x_{42} = 1$, onda to znači da se 42. riječ iz našeg vokabulara pojavila u tvitu. Učenje ovog modela svodi se na procjenu apriornih vjerojatnosti i izglednosti klase na temelju označenog skupa podataka. **Koliko ćemo ukupno vjerojatnosti trebati procijeniti za ovaj model?**

☐ A 30003 ☐ B 569 ☐ C 1683 ☐ D 31683

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

- 15** (T) Model Bayesovog klasifikatora učimo tako da procijenimo njegove parametre. Za to koristimo procjenu navjeće izglednosti s Laplaceovim zaglađivanjem. **U kojim situacijama je važno koristiti Laplaceovo zaglađivanje?**

☐ A Kada je skup za ispitivanje malen ☐ C Kada je skup za učenje velik
☐ B Kada je skup za učenje malen ☐ D Kada je skup za provjeru malen

- 16** (R) Mali je Ivica svakog svakog ljeta u zadnjih sedam godina naučio jedan novi programski jezik. Svoja vrijedna iskustva sazeo 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	Eval (E)	Izvođenje (I)	Paradigma (P)	Tipovi (T)	y	i	Eval (E)	Izvođenje (I)	Paradigma (P)	Tipovi (T)	y
1	lijena	kompajler	imperativna	statička	0	5	striktna	interpreter	imperativna	statička	1
2	striktna	interpreter	deklarativna	dinamička	0	6	lijena	kompajler	hibridna	dinamička	1
3	lijena	kompajler	imperativna	dinamička	0	7	striktna	kompajler	hibridna	dinamička	1
4	lijena	interpreter	hibridna	statička	0						

Ovog ljeta Mali Ivica opet naučiti novi programski jezik, 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?** (Korijski čvor je čvor na prvoj razini.)

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

- 17** (P) Rešetkasti svijet sastoji se od pet ćelija (numerirane 1 do 5) koje su smještene slijeva udesno (ćelija 1 je prva, ćelija 5 je posljednja). Ćelija 5 je završna. Na ćelijama 1 do 4 robot može poduzeti jednu od tri akcije: $a_1 = \text{lijevo}$, $a_2 = \text{desno}$, $a_3 = \text{sagni-se-i-pokupi-bocu}$. Boce se nalaze na ćelijama 2 i 4. Okolina robotu dodjeljuje nagrade kako slijedi. Za prelazak na ćeliju 5 robot dobiva 10 bodova. Za saginjanje i skupljanje boce robot dobiva 5 bodova, no ako se sagne i pokuša pokupiti bocu na ćeliji koja nema bocu, robot dobiva -5 bodova. Za sve ostale akcije isporučuje se nagrada življenja od -1 bod. U ovom svijetu robot uči optimalnu politiku algoritmom q-učenja. Neke su $\gamma = 1$ i $\alpha = 0.25$. U nekom trenutku naučene q-vrijednosti su sljedeće: $q(1, a_1) = 0$, $q(2, a_1) = 1$, $q(3, a_1) = 0$, $q(4, a_1) = 1$, $q(1, a_2) = 1$, $q(2, a_2) = 1$, $q(3, a_2) = 1$, $q(4, a_2) = 2$, $q(1, a_3) = 2$, $q(2, a_3) = -1$, $q(3, a_3) = -2$, $q(4, a_3) = 3$. Robot se nalazi na ćeliji 3 i poduzima akciju a_1 . **Koja će biti nova vrijednost za $q(3, a_1)$?**

☐ A 0 ☐ B 2.75 ☐ C 1.25 ☐ D -1

6. Prirodom inspirirani optimizacijski algoritmi (3 pitanja)

- 18 (R) Genetskim algoritmom traži se maksimum funkcije $f(x, y) = 3x^2 + 5y^2$. Svaka varijabla kodira se s 4 bita, pri čemu se x pretražuje iz intervala $[-4, 11]$, a y iz intervala $[-6, 9]$. U kromosomu najprije dolaze bitovi koji kodiraju x pa potom bitovi koji kodiraju y . Za odabir roditelja koristi se selekcija proporcionalna dobroti (engl. *roulette wheel selection*). Populacija se sastoji od sljedeće četiri jedinke:

$$K_1 = 10110011, \quad K_2 = 01010101, \quad K_3 = 00101010, \quad K_4 = 00010001$$

Koristi se križanje s jednom točkom prijeloma (točno na polovici kromosoma). Pretpostavite da se u trenutačnoj iteraciji algoritma slučajno odaberu baš dvije najmanje vjerojatne jedinke te da pri izgradnji djece mutacija igrom slučaja ne promijeni niti jedan bit. Označimo s F zbroj vjerojatnosti odabira odabranih jedinki, a s G dobrotu djeteta koje će biti umetnuto u novu populaciju (ako križanjem/mutacijom nastaje više djece, samo najbolje dijete ide dalje). Što od sljedećega vrijedi za F i G ?

- ☐ A $F > 0.1, G > 90$ ☐ B $F > 0.1, 5 < G < 50$ ☐ C $F < 0.3, 60 < G < 100$ ☐ D $F < 0.3, G < 70$

- 19 (P) Genetičkim algoritmom traži se minimum funkcije $f(x) = (x - \frac{15}{8})^2$ na intervalu $[-2, 4.3]$. Pri tome se koriste 6-bitovni kromosomi. Označimo s \hat{x} najbolje rješenje koje GA može pronaći, a s x^* vrijednost za koju funkcija poprima minimum. Koliko iznosi $|x^* - \hat{x}|$?

- ☐ A 0.125 ☐ B 0.75 ☐ C 1.0 ☐ D 0.025

- 20 (R) Mravlji algoritmom uz parametre $\alpha = \beta = 1$ traži se put (hamiltonov ciklus) kroz graf koji se sastoji od 5 čvorova: A do E . Za bridove su poznate vrijednosti heurističkih informacija: $\eta_{AB} = 7, \eta_{AC} = 1, \eta_{BC} = 2, \eta_{BD} = 3, \eta_{CD} = 2, \eta_{DE} = 4$ i $\eta_{CE} = 3$. U nekoj iteraciji algoritma vrijednosti feromonskih tragova su: $\tau_{AB} = 1, \tau_{AC} = 3, \tau_{BC} = 2, \tau_{BD} = 2, \tau_{CD} = 1, \tau_{DE} = 2$ i $\tau_{CE} = 3$. Sve vrijednosti η i τ su ujedno i simetrične. Mrav konstrukciju puta započinje iz čvora A . Pretpostavite da će prilikom slučajnog odabira sljedećeg čvora mrav svaki puta odabrati onaj koji je najvjerojatniji (i u koji je dozvoljeno odnosno moguće prijeći). Odredite put koji će mrav konstruirati. Koji je od sljedećih sljedova dio tog puta?

- ☐ A $D \rightarrow C \rightarrow E$ ☐ B $B \rightarrow C \rightarrow E$ ☐ C $B \rightarrow D \rightarrow C$ ☐ D $E \rightarrow C \rightarrow A$

Uvod u umjetnu inteligenciju – pismeni ispit (2022./2023.) – 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 (4 pitanja)

- 1** (P) Rješavamo problem pretraživanja prostora stanja za koji vrijedi $m = 10$, $d = 7$ i $b = 4$. Koristimo pretraživanje u širinu (BFS) i pretraživanje u dubinu (DFS). Za oba algoritma razmotrite najgori slučaj vremenske složenosti. **Koliko će u takvom slučaju algoritam DFS generirati više čvorova od algoritma BFS?**

☐ A 1310716 ☐ B 1376176 ☐ C 1310640 ☐ D 1376256

- 2** (R) Neka su definirani skup stanja $S = \{a, b, c, d, e, f\}$ i funkcija sljedbenika $\text{succ}(a) = \{(b, 1), (c, 2)\}$, $\text{succ}(b) = \text{succ}(f) = \emptyset$, $\text{succ}(c) = \{(d, 3), (e, 5)\}$, $\text{succ}(d) = \{(b, 8), (f, 5)\}$ te $\text{succ}(e) = \{(d, 1), (f, 2)\}$. Heurističke vrijednosti čvorova neka su $h(a) = 7$, $h(b) = 4$, $h(c) = 6$, $h(d) = h(e) = 2$, $h(f) = 0$. Početno stanje neka je a , a ciljno f . Izvršite pretraživanje algoritmom A^* , bilježeći pritom sadržaje liste otvorenih čvorova O i zatvorenih čvorova C u svakom koraku algoritma (u nultom koraku $O = [(a, 0)]$ i $C = \emptyset$). **Koji su sadržaji listi O i C nakon četvrtog koraka izvođenja algoritma A^* ?**

☐ A $O = [(e, 7), (f, 10)]$, $C = \{(a, 0), (c, 2), (d, 5)\}$
☐ B $O = [(e, 7), (f, 10)]$, $C = \{(a, 0), (b, 1), (c, 2), (d, 5)\}$
☐ C Algoritam ne dostiže četvrti korak
☐ D $O = [(f, 0), (e, 2), (b, 4)]$, $C = \{(a, 0), (b, 1), (c, 2), (d, 5)\}$

- 3** (T) Algoritam minimax s dubinskim ograničenjem uobičajeno je izveden pomoću dvije, međusobno rekurzivne funkcije. Neka su to funkcije $\text{maxValue}(s, d)$ i $\text{minValue}(s, d)$, gdje je s stanje igre, a d je brojač razine u stablu igre. Početni poziv algoritma je $\text{maxValue}(s_0, k)$, gdje je s_0 trenutno stanje igre, a k je dubinsko ograničenje. Funkcija $\text{maxValue}(s, d)$ koristi petlju s iteratorom $t \in \text{succ}(s)$ kako bi izračunala minimax-vrijednost m , koja je izvan petlje inicijalizirana na $-\infty$. **Kako glasi tijelo petlje u funkciji maxValue ?**

☐ A $m \leftarrow \min(m, \text{minValue}(t, d + 1))$ ☐ C $m \leftarrow \max(m, \text{maxValue}(t, d + 1))$
☐ B $m \leftarrow \max(m, \text{minValue}(t, d - 1))$ ☐ D $m \leftarrow \min(m, \text{maxValue}(t, d - 1))$

- 4** (P) Razmatramo dva igrača algoritma: jedan s heuristikom h_1 i dubinom pretraživanja d_1 te drugi s heuristikom h_2 i dubinom pretraživanja d_2 . Oba algoritma koriste strategiju minimax. **Kada će prvi algoritam češće pobjeđivati drugi algoritam?**

☐ A Ako $h_1 > h_2$ ☐ B Ako $h_1 = h_2$ i $d_1 < d_2$ ☐ C Ako $d_1 > d_2$ ☐ D Ako $h_1 = h_2$ i $d_1 > d_2$

2. Prikazivanje znanja i automatsko zaključivanje (4 pitanja)

- 5** (R) Dani su FOL atomi $P(f(x), y, g(w, f(z)))$ i $P(y, f(g(a, z)), x)$, gdje je P predikat, a je konstanta, a f i g su funkcije. **Što je najopćenitiji zajednički unifikator ovih dvaju atoma?**

☐ A Atome nije moguće unificirati ☐ C $P(f(f(g(a, b))), f(g(a, b)), f(g(a, b)))$
☐ B $P(f(g(a, b)), f(g(a, b)), g(a, b))$ ☐ D $P(f(x), f(g(x, y)), g(a, b))$

6 (T) Relacija logičke posljedice između formula F i G definirana je na temelju vrijednosti istinitosti formula F i G kroz sve moguće interpretacije tih formula. Dokazivanje logičke posljedice od praktičnog je značaja za umjetnu inteligenciju. **Na koji način dokazujemo logičku posljedicu u FOL?**

- ☐ A Pretvorbom u klauzalni oblik i skolemizacijom
- ☐ B Pomoću ispravnih i potpunih pravila zaključivanja
- ☐ C Konstrukcijom tablice istinitosti i provjerom modela
- ☐ D Primjenom izravne rezolucije i faktORIZACIJE

7 (R) Zadane su sljedeće dvije premise:

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

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

- ☐ A $\exists x P(x) \wedge P(b)$
- ☐ B $\exists x \neg P(x)$
- ☐ C $\exists x P(x) \vee P(b)$
- ☐ D $\neg \exists x P(x)$

8 (P) Ispravnost je važno svojstvo pravila zaključivanja. **Koje je od sljedećih pravila zaključivanja ispravno?**

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

3. Logičko programiranje i ekspertni sustavi (2 pitanja)

9 (T) Kao i sustavi formalne logike, ekspertni se sustavi temelje na formalnom prikazu znanja i izvođenju novog znanja primjenom pravila zaključivanja. Međutim, između tih sustava postoje razlike. **Po čemu se ekspertni sustavi razlikuju od općenitih sustava formalne logike?**

- ☐ A Znanje je prikazano u obliku činjenica i pravila, a ne proizvoljnih logičkih formula
- ☐ B Znanje je prikazano u obliku ako-onda pravila, a ne Hornovih ili definitnih klauzula
- ☐ C Za dokazivanje se koriste logička pravila koja su potpuna, a ne samo ispravna
- ☐ D Za dokazivanje se koristi rezolucija opovrgavanjem, a ne izravna rezolucija

10 (T) Izvođenje logičkih programa u Prologu odgovara dokazivanju teorema rezolucijom opovrgavanjem. Stanje izvođenja u Prologu definirano je stogom, koji početno sadrži atom upita. **Što sadrži Prologov stog u nastavku izvođenja programa?**

- ☐ A Atome iz tijela logičkih pravila
- ☐ B Činjenice iz logičkog programa
- ☐ C Atome iz glave logičkih pravila
- ☐ D Atome dviju roditeljskih klauzula

4. Modeliranje neizvjesnosti (2 pitanja)

11 Nad univerzalnim skupom $\{a, b, c, d\}$ zadani su neizraziti skupovi $X = \{0.1/a + 0/b + 0.3/c + 1/d\}$, $Y = \{0.5/a + 0.4/b + 0/c + 0.2/d\}$ i $Z = \{1/a + 0/b + 0/c + 0.5/d\}$. Koristeći Zadehove operatore odredite neizraziti skup koji odgovara jezičnome izrazu “Ne Z i (X ili Y)”.

- ☐ A $\{0/a + 0.4/b + 0.3/c + 0.5/d\}$
- ☐ B $\{0.9/a + 0.4/b + 0/c + 0/d\}$
- ☐ C $\{0.1/a + 1/b + 1/c + 0.5/d\}$
- ☐ D $\{1/a + 1/b + 0.7/c + 0.5/d\}$

12 (T) Bayesovo pravilo možemo koristiti za zaključivanje abdukcijom. **Što to znači?**

- ☐ A Na temelju opažene hipoteze zaključujemo o dokazu
- ☐ B Na temelju ostvarenog uzroka zaključujemo o hipotezi
- ☐ C Na temelju ostvarenog uzroka zaključujemo o dokazima
- ☐ D Na temelju opaženih dokaza zaključujemo o uzroku

5. Strojno učenje, umjetne neuronske mreže i podržano učenje (5 pitanja)

- 13** (P) Gradimo naivan Bayesov klasifikator za klasifikaciju poruka Twittera (tvitova) prema sentimentu. Svaki tvit želimo klasificirati u jednu od tri klase: pozitivan, negativan ili neutralan. Svaki tvit sastoji se od najviše 280 riječi i prikazujemo ga kao jedan binaran vektor značajki. Vektor se sastoji od 5000 značajki, i svaka značajka odgovara jednoj od 5000 riječi iz našeg vokabulara. Ako je značajka postavljena na 1, to znači da se dotična riječ pojavila u tvitu, inače je značajka postavljena na 0. Npr., ako je $x_{42} = 1$, onda to znači da se 42. riječ iz našeg vokabulara pojavila u tvitu. Učenje ovog modela svodi se na procjenu apriornih vjerojatnosti i izglednosti klase na temelju označenog skupa podataka. **Koliko ćemo ukupno vjerojatnosti trebati procijeniti za ovaj model?**

☐ A 569 ☐ B 31683 ☐ C 1683 ☐ D 30003

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

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

Tim 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 $[w_2, w_1, w_0] = [8, 4, -12]$. **Koliko će se puta provesti korigiranje vektora težina i koja je njegova konačna vrijednost?**

☐ A 3 puta, $[8, -1, -11]$ ☐ B 3 puta, $[7, 2, -12]$ ☐ C 2 puta, $[10, 4, -28]$ ☐ D 2 puta, $[5, 2, -14]$

- 15** (R) Mali je Ivica svakog svakog ljeta u zadnjih sedam godina naučio jedan novi programski jezik. Svoja vrijedna iskustva sazeo 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	Eval (E)	Izvođenje (I)	Paradigma (P)	Tipovi (T)	y	i	Eval (E)	Izvođenje (I)	Paradigma (P)	Tipovi (T)	y
1	lijena	kompajler	imperativna	statička	0	5	striktna	interpreter	imperativna	statička	1
2	striktna	interpreter	deklarativna	dinamička	0	6	lijena	kompajler	hibridna	dinamička	1
3	lijena	kompajler	imperativna	dinamička	0	7	striktna	kompajler	hibridna	dinamička	1
4	lijena	interpreter	hibridna	statička	0						

Ovog ljeta Mali Ivica opet naučiti novi programski jezik, pa je odlučio izgraditi stablo odluke za klasifikaciju programskih jezika koje još nije naučio. Pomozite Malom Ivici te na gornjem skupu primjera primijenite 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?** (Korijski čvor je čvor na prvoj razini.)

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

- 16** (T) Model Bayesovog klasifikatora učimo tako da procijenimo njegove parametre. Za to koristimo procjenu navjeće izglednosti s Laplaceovim zaglađivanjem. **U kojim situacijama je važno koristiti Laplaceovo zaglađivanje?**

☐ A Kada je skup za ispitivanje malen ☐ C Kada je skup za učenje malen
☐ B Kada je skup za provjeru malen ☐ D Kada je skup za učenje velik

- 17** (P) Rešetkasti svijet sastoji se od pet ćelija (numerirane 1 do 5) koje su smještene slijeva udesno (ćelija 1 je prva, ćelija 5 je posljednja). Ćelija 5 je završna. Na ćelijama 1 do 4 robot može poduzeti jednu od tri akcije: $a_1 = \text{lijevo}$, $a_2 = \text{desno}$, $a_3 = \text{sagni-se-i-pokupi-bocu}$. Boce se nalaze na ćelijama 2 i 4. Okolina robotu dodjeljuje nagrade kako slijedi. Za prelazak na ćeliju 5 robot dobiva 10 bodova. Za saginjanje i skupljanje boce robot dobiva 5 bodova, no ako se sagne i pokuša pokupiti bocu na ćeliji koja nema bocu, robot dobiva -5 bodova. Za sve ostale akcije isporučuje se nagrada življenja od -1 bod. U ovom svijetu robot uči optimalnu politiku algoritmom q-učenja. Neka su $\gamma = 1$ i $\alpha = 0.25$. U nekom trenutku naučene q-vrijednosti su sljedeće: $q(1, a_1) = 0$, $q(2, a_1) = 1$, $q(3, a_1) = 0$, $q(4, a_1) = 1$, $q(1, a_2) = 1$, $q(2, a_2) = 1$, $q(3, a_2) = 1$, $q(4, a_2) = 2$, $q(1, a_3) = 2$, $q(2, a_3) = -1$, $q(3, a_3) = -2$, $q(4, a_3) = 3$. Robot se nalazi na ćeliji 4 i poduzima akciju a_3 . **Koja će biti nova vrijednost za $q(3, a_3)$?**

☐ A 2.25 ☐ B 8 ☐ C 4.25 ☐ D 3

6. Prirodom inspirirani optimizacijski algoritmi (3 pitanja)

- 18 (R) Mravlji algoritmom uz parametre $\alpha = \beta = 1$ traži se put (hamiltonov ciklus) kroz graf koji se sastoji od 5 čvorova: A do E . Za bridove su poznate vrijednosti heurističkih informacija: $\eta_{AB} = 7$, $\eta_{AC} = 1$, $\eta_{BC} = 2$, $\eta_{BD} = 3$, $\eta_{CD} = 2$, $\eta_{DE} = 4$ i $\eta_{CE} = 3$. U nekoj iteraciji algoritma vrijednosti feromonskih tragova su: $\tau_{AB} = 1$, $\tau_{AC} = 3$, $\tau_{BC} = 2$, $\tau_{BD} = 2$, $\tau_{CD} = 1$, $\tau_{DE} = 2$ i $\tau_{CE} = 3$. Sve vrijednosti η i τ su ujedno i simetrične. Mrav konstrukciju puta započinje iz čvora A . Pretpostavite da će prilikom slučajnog odabira sljedećeg čvora mrav svaki puta odabrati onaj koji je najvjerojatniji (i u koji je dozvoljeno odnosno moguće prijeći). Odredite put koji će mrav konstruirati. **Koji je od sljedećih sljedova dio tog puta?**

☐ A $B \rightarrow C \rightarrow E$ ☐ B $B \rightarrow D \rightarrow C$ ☐ C $E \rightarrow C \rightarrow A$ ☐ D $D \rightarrow C \rightarrow E$

- 19 (P) Genetičkim algoritmom traži se minimum funkcije $f(x) = (x - \frac{7}{8})^2$ na intervalu $[-3, 3.3]$. Pri tome se koriste 6-bitovni kromosomi. Označimo s \hat{x} najbolje rješenje koje GA može pronaći, a s x^* vrijednost za koju funkcija poprima minimum. **Koliko iznosi $|x^* - \hat{x}|$?**

☐ A 0.3 ☐ B 0.025 ☐ C 0.125 ☐ D 0.625

- 20 (T) Genetski algoritmi koriste operatore selekcije, križanja i mutacije te mehanizam elitizma kako bi pretražili prostor rješenja. **Što bi se dogodilo kada ne bismo koristili mehanizam elitizma?**

- ☐ A Algoritam bi generirao rješenja koja su neispravna (izvan intervala pretrage)
☐ B Povećala bi se prosječna vrijednost funkcije dobrote u populaciji
☐ C Algoritam bi mogao vratiti rješenje koje je lošije od nekog ranije generiranog rješenja
☐ D Porasla bi vjerojatnost da pretraga zapne u lokalnome optimumu

Uvod u umjetnu inteligenciju – pismeni ispit (2022./2023.) – 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 (4 pitanja)

- 1** (T) Algoritam minimax s dubinskim ograničenjem uobičajeno je izveden pomoću dvije, međusobno rekurzivne funkcije. Neka su to funkcije $\text{maxValue}(s, d)$ i $\text{minValue}(s, d)$, gdje je s stanje igre, a d je brojač razine u stablu igre. Početni poziv algoritma je $\text{maxValue}(s_0, k)$, gdje je s_0 trenutno stanje igre, a k je dubinsko ograničenje. Funkcija $\text{minValue}(s, d)$ koristi petlju s iteratorom $t \in \text{succ}(s)$ kako bi izračunala minimax-vrijednost m , koja je izvan petlje inicijalizirana na $+\infty$. **Kako glasi tijelo petlje u funkciji minValue ?**

- ☐ A $m \leftarrow \min(m, \text{minValue}(t, d + 1))$ ☐ C $m \leftarrow \max(m, \text{minValue}(t, d - 1))$
☐ B $m \leftarrow \min(m, \text{maxValue}(t, d - 1))$ ☐ D $m \leftarrow \max(m, \text{maxValue}(t, d + 1))$

- 2** (P) Razmatramo dva igrača algoritma: jedan s heuristikom h_1 i dubinom pretraživanja d_1 te drugi s heuristikom h_2 i dubinom pretraživanja d_2 . Oba algoritma koriste strategiju minimax. **Kada će prvi algoritam češće pobjeđivati drugi algoritam?**

- ☐ A Ako $h_1 = h_2$ i $d_1 > d_2$ ☐ B Ako $h_2 > h_1$ ☐ C Ako $h_1 > h_2$ ☐ D Ako $h_1 = h_2$ i $d_1 < d_2$

- 3** (P) Rješavamo problem pretraživanja prostora stanja za koji vrijedi $m = 9$, $d = 6$ i $b = 3$. Koristimo pretraživanje u širinu (BFS) i pretraživanje u dubinu (DFS). Za oba algoritma razmotrite najgori slučaj vremenske složenosti. **Koliko će u takvom slučaju algoritam DFS generirati više čvorova od algoritma BFS?**

- ☐ A 28395 ☐ B 26241 ☐ C 26208 ☐ D 28431

- 4** (R) Neka su definirani skup stanja $S = \{a, b, c, d, e, f\}$ i funkcija sljedbenika $\text{succ}(a) = \{(b, 1), (c, 2)\}$, $\text{succ}(b) = \text{succ}(f) = \emptyset$, $\text{succ}(c) = \{(d, 3), (e, 5)\}$, $\text{succ}(d) = \{(b, 8), (f, 5)\}$ te $\text{succ}(e) = \{(d, 1), (f, 2)\}$. Heurističke vrijednosti čvorova neka su $h(a) = 7$, $h(b) = 4$, $h(c) = 6$, $h(d) = h(e) = 2$, $h(f) = 0$. Početno stanje neka je a , a ciljno f . Izvršite pretraživanje algoritmom A^* , bilježeći pritom sadržaje liste otvorenih čvorova O i zatvorenih čvorova C u svakom koraku algoritma (u nultom koraku $O = [(a, 0)]$ i $C = \emptyset$). **Koji su sadržaji listi O i C nakon petog koraka izvođenja algoritma A^* ?**

- ☐ A $O = [(f, 9)]$, $C = \{(a, 0), (b, 1), (c, 2), (d, 5), (e, 7)\}$
☐ B $O = [(f, 10)]$, $C = \{(a, 0), (b, 1), (c, 2), (d, 5), (e, 7)\}$
☐ C $O = [(e, 7), (f, 10), (b, 13)]$, $C = \{(a, 0), (b, 1), (c, 2), (d, 5)\}$
☐ D Algoritam ne dostiže peti korak

2. Prikazivanje znanja i automatsko zaključivanje (4 pitanja)

- 5** (R) Zadane su sljedeće dvije premise:

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

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

- ☐ A $\exists x \neg P(x)$ ☐ B $\exists x P(x) \vee P(b)$ ☐ C $\neg \exists x P(x)$ ☐ D $\exists x R(x)$

- 6** (P) Ispravnost je važno svojstvo pravila zaključivanja. **Koje je od sljedećih pravila zaključivanja ispravno?**

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

7 (R) Dani su FOL atomi $P(f(x), y, g(w, b))$ i $P(y, f(g(a, z)), x)$, gdje je P predikat, a i b su konstante, a f i g su funkcije. Što je najopćenitiji zajednički unifikator ovih dvaju atoma?

- ☐ A $P(f(y), f(g(a, z)), g(w, b))$ ☐ C $P(f(g(a, b)), f(g(a, z)), g(w, b))$
☐ B Atome nije moguće unificirati ☐ D $P(f(g(a, b)), f(g(a, b)), g(a, b))$

8 (T) Relacija logičke posljedice između formula F i G definirana je na temelju vrijednosti istinitosti formula F i G kroz sve moguće interpretacije tih formula. Dokazivanje logičke posljedice od praktičnog je značaja za umjetnu inteligenciju. Na koji način dokazujemo logičku posljedicu u FOL?

- ☐ A Primjenom izravne rezolucije i faktorizacije
☐ B Pretvorbom u klauzalni oblik i skolemizacijom
☐ C Pomoću ispravnih i potpunih pravila zaključivanja
☐ D Konstrukcijom tablice istinitosti i provjerom modela

3. Logičko programiranje i ekspertni sustavi (2 pitanja)

9 (T) Izvođenje logičkih programa u Prologu odgovara dokazivanju teorema rezolucijom opovrgavanjem. Stanje izvođenja u Prologu definirano je stogom, koji početno sadrži atom upita. Što sadrži Prologov stog u nastavku izvođenja programa?

- ☐ A Atome iz glave logičkih pravila ☐ C Atome dviju roditeljskih klauzula
☐ B Činjenice iz logičkog programa ☐ D Atome iz tijela logičkih pravila

10 (T) Kao i sustavi formalne logike, ekspertni se sustavi temelje na formalnom prikazu znanja i izvođenju novog znanja primjenom pravila zaključivanja. Međutim, između tih sustava postoje razlike. Po čemu se ekspertni sustavi razlikuju od općenitih sustava formalne logike?

- ☐ A Za dokazivanje se koristi rezolucija opovrgavanjem, a ne izravna rezolucija
☐ B Znanje je prikazano u obliku ako-onda pravila, a ne Hornovih ili definitnih klauzula
☐ C Znanje je prikazano u obliku činjenica i pravila, a ne proizvoljnih logičkih formula
☐ D Za dokazivanje se koriste logička pravila koja su potpuna, a ne samo ispravna

4. Modeliranje neizvjesnosti (2 pitanja)

11 (T) Bayesovo pravilo možemo koristiti za zaključivanje abdukcijom. Što to znači?

- ☐ A Na temelju opažene hipoteze zaključujemo o dokazu
☐ B Na temelju ostvarenog uzroka zaključujemo o dokazima
☐ C Na temelju ostvarenog uzroka zaključujemo o hipotezi
☐ D Na temelju opaženih dokaza zaključujemo o uzroku

12 (R) Nad univerzalnim skupom $\{a, b, c, d\}$ zadani su neizraziti skupovi $X = \{0.1/a + 0/b + 0.3/c + 1/d\}$, $Y = \{0.5/a + 0.4/b + 0/c + 0.2/d\}$ i $Z = \{1/a + 0/b + 0/c + 0.5/d\}$. Koristeći Zadehove operatore izvodimo neizraziti skup X tako da on odgovara jezičnome izrazu "Ne X ili (Y i Z)". Kako glasi neizraziti skup X ?

- ☐ A $\{0.5/a + 0/b + 0/c + 0/d\}$ ☐ C $\{0.9/a + 0.4/b + 0/c + 0/d\}$
☐ B $\{1/a + 1/b + 0.7/c + 0.5/d\}$ ☐ D $\{0.9/a + 1.0/b + 0.7/c + 0.2/d\}$

5. Strojno učenje, umjetne neuronske mreže i podržano učenje (5 pitanja)

- 13** (T) Model Bayesovog klasifikatora učimo tako da procijenimo njegove parametre. Za to koristimo procjenu navjeće izglednosti s Laplaceovim zaglađivanjem. **U kojim situacijama je važno koristiti Laplaceovo zaglađivanje?**

☐ A Kada je skup za učenje velik ☐ C Kada je skup za provjeru malen
☐ B Kada je skup za učenje malen ☐ D Kada je skup za ispitivanje malen

- 14** (R) Mali je Ivica svakog svakog ljeta u zadnjih sedam godina naučio jedan novi programski jezik. Svoja vrijedna iskustva sazeo 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	Eval (E)	Izvođenje (I)	Paradigma (P)	Tipovi (T)	y	i	Eval (E)	Izvođenje (I)	Paradigma (P)	Tipovi (T)	y
1	lijena	kompajler	imperativna	statička	0	5	striktna	interpreter	imperativna	statička	1
2	striktna	interpreter	deklarativna	dinamička	0	6	lijena	kompajler	hibridna	dinamička	1
3	lijena	kompajler	imperativna	dinamička	0	7	striktna	kompajler	hibridna	dinamička	1
4	lijena	interpreter	hibridna	statička	0						

Ovog ljeta Mali Ivica opet naučiti novi programski jezik, 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 (ljeviya značajka). **Koje se značajke nalaze u drugoj razini stabla odluke dobivenog algoritmom ID3?** (Korijski čvor je čvor na prvoj razini.)

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

- 15** (P) Gradimo naivan Bayesov klasifikator za klasifikaciju poruka Twittera (tvitova) prema sentimentu. Svaki tvit želimo klasificirati u jednu od tri klase: pozitivan, negativan ili neutralan. Svaki tvit sastoji se od najviše 280 riječi i prikazujemo ga kao jedan binaran vektor značajki. Vektor se sastoji od 5000 značajki, i svaka značajka odgovara jednoj od 5000 riječi iz našeg vokabulara. Ako je značajka postavljena na 1, to znači da se dotična riječ pojavila u tvitu, inače je značajka postavljena na 0. Npr., ako je $x_{42} = 1$, onda to znači da se 42. riječ iz našeg vokabulara pojavila u tvitu. Učenje ovog modela svodi se na procjenu apriornih vjerojatnosti i izglednosti klase na temelju označenog skupa podataka. **Koliko ćemo ukupno vjerojatnosti trebati procijeniti za ovaj model?**

☐ A 569 ☐ B 30003 ☐ C 1683 ☐ D 31683

- 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. Neke 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, $[7, 2, -12]$ ☐ C 1 puta, $[5, 2, -13]$ ☐ D 2 puta, $[8, -1, -11]$

- 17** (P) Rešetkasti svijet sastoji se od pet ćelija (numerirane 1 do 5) koje su smještene slijeva udesno (ćelija 1 je prva, ćelija 5 je posljednja). Ćelija 5 je završna. Na ćelijama 1 do 4 robot može poduzeti jednu od tri akcije: $a_1 = \text{lijevo}$, $a_2 = \text{desno}$, $a_3 = \text{sagni-se-i-pokupi-bocu}$. Boce se nalaze na ćelijama 2 i 4. Okolina robotu dodjeljuje nagrade kako slijedi. Za prelazak na ćeliju 5 robot dobiva 10 bodova. Za saginjanje i skupljanje boce robot dobiva 5 bodova, no ako se sagne i pokuša pokupiti bocu na ćeliji koja nema bocu, robot dobiva -5 bodova. Za sve ostale akcije isporučuje se nagrada življenja od -1 bod. U ovom svijetu robot uči optimalnu politiku algoritmom q-učenja. Neke su $\gamma = 1$ i $\alpha = 0.25$. U nekom trenutku naučene q-vrijednosti su sljedeće: $q(1, a_1) = 0$, $q(2, a_1) = 1$, $q(3, a_1) = 0$, $q(4, a_1) = 1$, $q(1, a_2) = 1$, $q(2, a_2) = 1$, $q(3, a_2) = 1$, $q(4, a_2) = 2$, $q(1, a_3) = 2$, $q(2, a_3) = -1$, $q(3, a_3) = -2$, $q(4, a_3) = 3$. Robot se nalazi na ćeliji 4 i poduzima akciju a_3 . **Koja će biti nova vrijednost za $q(3, a_3)$?**

☐ A 3 ☐ B 4.25 ☐ C 2.25 ☐ D 8

6. Prirodom inspirirani optimizacijski algoritmi (3 pitanja)

- 18 (R) Genetskim algoritmom traži se maksimum funkcije $f(x, y) = 3x^2 + 5y^2$. Svaka varijabla kodira se s 4 bita, pri čemu se x pretražuje iz intervala $[-4, 11]$, a y iz intervala $[-6, 9]$. U kromosomu najprije dolaze bitovi koji kodiraju x pa potom bitovi koji kodiraju y . Za odabir roditelja koristi se selekcija proporcionalna dobroti (engl. *roulette wheel selection*). Populacija se sastoji od sljedeće četiri jedinke:

$$K_1 = 10110011, \quad K_2 = 01010101, \quad K_3 = 00101010, \quad K_4 = 00010001$$

Koristi se križanje s jednom točkom prijeloma (točno na polovici kromosoma). Pretpostavite da se u trenutačnoj iteraciji algoritma slučajno odaberu baš dvije najmanje vjerojatne jedinke te da pri izgradnji djece mutacija igrom slučaja ne promijeni niti jedan bit. Označimo s F zbroj vjerojatnosti odabira odabranih jedinki, a s G dobrotu djeteta koje će biti umetnuto u novu populaciju (ako križanjem/mutacijom nastaje više djece, samo najbolje dijete ide dalje). **Što od sljedećega vrijedi za F i G ?**

- ☐ A $F < 0.3, G < 70$ ☐ B $F > 0.1, G > 90$ ☐ C $F < 0.3, 60 < G < 100$ ☐ D $F > 0.1, 5 < G < 50$

- 19 (P) Genetičkim algoritmom traži se minimum funkcije $f(x) = (x - \frac{15}{8})^2$ na intervalu $[-2, 4.3]$. Pri tome se koriste 6-bitovni kromosomi. Označimo s \hat{x} najbolje rješenje koje GA može pronaći, a s x^* vrijednost za koju funkcija poprima minimum. **Koliko iznosi $|x^* - \hat{x}|$?**

- ☐ A 0.025 ☐ B 0.3 ☐ C 0.625 ☐ D 0.875

- 20 (T) Genetski algoritmi koriste operatore selekcije, križanja i mutacije te mehanizam elitizma kako bi pretražili prostor rješenja. **Što bi se dogodilo kada ne bismo koristili mehanizam elitizma?**

- ☐ A Porasla bi vjerojatnost da pretraga zapne u lokalnome optimumu
☐ B Algoritam bi generirao rješenja koja su neispravna (izvan intervala pretrage)
☐ C Algoritam bi mogao vratiti rješenje koje je lošije od nekog ranije generiranog rješenja
☐ D Povećala bi se prosječna vrijednost funkcije dobrote u populaciji

Uvod u umjetnu inteligenciju – pismeni ispit (2022./2023.) – 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 (4 pitanja)

- 1** (T) Algoritam minimax s dubinskim ograničenjem uobičajeno je izveden pomoću dvije, međusobno rekurzivne funkcije. Neka su to funkcije $\text{maxValue}(s, d)$ i $\text{minValue}(s, d)$, gdje je s stanje igre, a d je brojač razine u stablu igre. Početni poziv algoritma je $\text{maxValue}(s_0, k)$, gdje je s_0 trenutno stanje igre, a k je dubinsko ograničenje. Funkcija $\text{maxValue}(s, d)$ koristi petlju s iteratorom $t \in \text{succ}(s)$ kako bi izračunala minimax-vrijednost m , koja je izvan petlje inicijalizirana na $-\infty$. **Kako glasi tijelo petlje u funkciji maxValue ?**

- ☐ A $m \leftarrow \max(m, \text{maxValue}(t, d + 1))$ ☐ C $m \leftarrow \max(m, \text{minValue}(t, d - 1))$
☐ B $m \leftarrow \min(m, \text{minValue}(t, d + 1))$ ☐ D $m \leftarrow \min(m, \text{maxValue}(t, d - 1))$

- 2** (P) Rješavamo problem pretraživanja prostora stanja za koji vrijedi $m = 9$, $d = 6$ i $b = 3$. Koristimo pretraživanje u širinu (BFS) i pretraživanje u dubinu (DFS). Za oba algoritma razmotrite najgori slučaj vremenske složenosti. **Koliko će u takvom slučaju algoritam DFS generirati više čvorova od algoritma BFS?**

- ☐ A 28395 ☐ B 26241 ☐ C 28431 ☐ D 26208

- 3** (R) Neka su definirani skup stanja $S = \{a, b, c, d, e, f\}$ i funkcija sljedbenika $\text{succ}(a) = \{(b, 1), (c, 2)\}$, $\text{succ}(b) = \text{succ}(f) = \emptyset$, $\text{succ}(c) = \{(d, 3), (e, 5)\}$, $\text{succ}(d) = \{(b, 8), (f, 5)\}$ te $\text{succ}(e) = \{(d, 1), (f, 2)\}$. Heurističke vrijednosti čvorova neka su $h(a) = 7$, $h(b) = 4$, $h(c) = 6$, $h(d) = h(e) = 2$, $h(f) = 0$. Početno stanje neka je a , a ciljno f . Izvršite pretraživanje algoritmom A^* , bilježeći pritom sadržaje liste otvorenih čvorova O i zatvorenih čvorova C u svakom koraku algoritma (u nultom koraku $O = [(a, 0)]$ i $C = \emptyset$). **Koji su sadržaji listi O i C nakon petog koraka izvođenja algoritma A^* ?**

- ☐ A $O = [(f, 9)]$, $C = \{(a, 0), (b, 1), (c, 2), (d, 5), (e, 7)\}$
☐ B Algoritam ne dostiže peti korak
☐ C $O = [(e, 7), (f, 10), (b, 13)]$, $C = \{(a, 0), (b, 1), (c, 2), (d, 5)\}$
☐ D $O = [(e, 7), (f, 10)]$, $C = \{(a, 0), (c, 2), (d, 5)\}$

- 4** (P) Razmatramo dva igrača algoritma: jedan s heuristikom h_1 i dubinom pretraživanja d_1 te drugi s heuristikom h_2 i dubinom pretraživanja d_2 . Oba algoritma koriste strategiju minimax. **Kada će prvi algoritam češće pobjeđivati drugi algoritam?**

- ☐ A Ako $d_1 > d_2$ ☐ B Ako $h_1 > h_2$ ☐ C Ako $d_2 < d_1$ ☐ D Ako $h_1 = h_2$ i $d_1 > d_2$

2. Prikazivanje znanja i automatsko zaključivanje (4 pitanja)

- 5** (P) Ispravnost je važno svojstvo pravila zaključivanja. **Koje je od sljedećih pravila zaključivanja ispravno?**

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

- 6** (T) Relacija logičke posljedice između formula F i G definirana je na temelju vrijednosti istinitosti formula F i G kroz sve moguće interpretacije tih formula. Dokazivanje logičke posljedice od praktičnog je značaja za umjetnu inteligenciju. **Na koji način dokazujemo logičku posljedicu u FOL?**

- ☐ A Primjenom izravne rezolucije i faktORIZACIJE
☐ B Pretvorbom u klauzalni oblik i skolemizacijom
☐ C Pomoću ispravnih i potpunih pravila zaključivanja
☐ D Konstrukcijom tablice istinitosti i provjerom modela

- 7 (R) Zadane su sljedeće dvije premise:

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

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

- ☐ A $\forall x Q(a, x)$ ☐ B $\forall x P(x)$ ☐ C $\exists x P(x)$ ☐ D $\exists x R(x)$

- 8 (R) Dani su FOL atomi $P(f(x), y, g(w, f(z)))$ i $P(y, f(g(a, z)), x)$, gdje je P predikat, a je konstanta, a f i g su funkcije. Što je najopćenitiji zajednički unifikator ovih dvaju atoma?

- ☐ A $P(f(g(a, b)), f(g(a, b)), g(a, b))$ ☐ C $P(f(g(a, b)), f(g(a, z)), g(w, b))$
☐ B Atome nije moguće unificirati ☐ D $P(f(x), f(g(x, y)), g(a, b))$

3. Logičko programiranje i ekspertni sustavi (2 pitanja)

- 9 (T) Kao i sustavi formalne logike, ekspertni se sustavi temelje na formalnom prikazu znanja i izvođenju novog znanja primjenom pravila zaključivanja. Međutim, između tih sustava postoje razlike. Po čemu se ekspertni sustavi razlikuju od općenitih sustava formalne logike?

- ☐ A Znanje je prikazano u obliku činjenica i pravila, a ne proizvoljnih logičkih formula
☐ B Za dokazivanje se koriste logička pravila koja su potpuna, a ne samo ispravna
☐ C Za dokazivanje se koristi rezolucija opovrgavanjem, a ne izravna rezolucija
☐ D Znanje je prikazano u obliku ako-onda pravila, a ne Hornovih ili definitnih klauzula

- 10 (T) Izvođenje logičkih programa u Prologu odgovara dokazivanju teorema rezolucijom opovrgavanjem. Stanje izvođenja u Prologu definirano je stogom, koji početno sadrži atom upita. Što sadrži Prologov stog u nastavku izvođenja programa?

- ☐ A Atome dviju roditeljskih klauzula ☐ C Činjenice iz logičkog programa
☐ B Atome iz tijela logičkih pravila ☐ D Atome iz glave logičkih pravila

4. Modeliranje neizvjesnosti (2 pitanja)

- 11 Nad univerzalnim skupom $\{a, b, c, d\}$ zadani su neizraziti skupovi $X = \{0.1/a + 0/b + 0.3/c + 1/d\}$, $Y = \{0.5/a + 0.4/b + 0/c + 0.2/d\}$ i $Z = \{1/a + 0/b + 0/c + 0.5/d\}$. Koristeći Zadehove operatore odredite neizraziti skup koji odgovara jezičnome izrazu "Ne Z i (X ili Y)".

- ☐ A $\{0.1/a + 1/b + 1/c + 0.5/d\}$ ☐ C $\{1/a + 1/b + 0.7/c + 0.5/d\}$
☐ B $\{0/a + 0.4/b + 0.3/c + 0.5/d\}$ ☐ D $\{0.9/a + 0.4/b + 0/c + 0/d\}$

- 12 (T) Bayesovo pravilo možemo koristiti za zaključivanje abdukcijom. Što to znači?

- ☐ A Na temelju ostvarenog uzroka zaključujemo o dokazima
☐ B Na temelju opaženih dokaza zaključujemo o uzroku
☐ C Na temelju opažene hipoteze zaključujemo o dokazu
☐ D Na temelju ostvarenog uzroka zaključujemo o hipotezi

5. Strojno učenje, umjetne neuronske mreže i podržano učenje (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 $y_i = f(x_{2,i}, x_{1,i})$. 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 3 puta, $[14, -2, -28]$ ☐ B 2 puta, $[7, 2, -12]$ ☐ C 3 puta, $[7, -1, -14]$ ☐ D 2 puta, $[6, 3, -15]$

- 14 (R) Mali je Ivica svakog svakog ljeta u zadnjih sedam godina naučio jedan novi programski jezik. Svoja vrijedna iskustva sazeo 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	Eval (E)	Izvođenje (I)	Paradigma (P)	Tipovi (T)	y	i	Eval (E)	Izvođenje (I)	Paradigma (P)	Tipovi (T)	y
1	lijena	kompajler	imperativna	statička	0	5	striktna	interpreter	imperativna	statička	1
2	striktna	interpreter	deklarativna	dinamička	0	6	lijena	kompajler	hibridna	dinamička	1
3	lijena	kompajler	imperativna	dinamička	0	7	striktna	kompajler	hibridna	dinamička	1
4	lijena	interpreter	hibridna	statička	0						

Ovog ljeta Mali Ivica opet naučiti novi programski jezik, 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 (ljeviya značajka). **Koje se značajke nalaze u drugoj razini stabla odluke dobivenog algoritmom ID3?** (Korijsenski čvor je čvor na prvoj razini.)

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

- 15 (P) Gradimo naivan Bayesov klasifikator za klasifikaciju poruka Twittera (tvitova) prema sentimentu. Svaki tvit želimo klasificirati u jednu od tri klase: pozitivan, negativan ili neutralan. Svaki tvit sastoji se od najviše 280 riječi i prikazujemo ga kao jedan binaran vektor značajki. Vektor se sastoji od 5000 značajki, i svaka značajka odgovara jednoj od 5000 riječi iz našeg vokabulara. Ako je značajka postavljena na 1, to znači da se dotična riječ pojavila u tvitu, inače je značajka postavljena na 0. Npr., ako je $x_{42} = 1$, onda to znači da se 42. riječ iz našeg vokabulara pojavila u tvitu. Učenje ovog modela svodi se na procjenu apriornih vjerojatnosti i izglednosti klase na temelju označenog skupa podataka. **Koliko ćemo ukupno vjerojatnosti trebati procijeniti za ovaj model?**

- ☐ A 569 ☐ B 31683 ☐ C 1683 ☐ D 30003

- 16 (T) Model Bayesovog klasifikatora učimo tako da procijenimo njegove parametre. Za to koristimo procjenu navjeće izglednosti s Laplaceovim zaglađivanjem. **U kojim situacijama je važno koristiti Laplaceovo zaglađivanje?**

- ☐ A Kada je skup za provjeru malen ☐ C Kada je skup za učenje velik
☐ B Kada je skup za učenje malen ☐ D Kada je skup za ispitivanje malen

- 17 (P) Rešetkasti svijet sastoji se od pet ćelija (numerirane 1 do 5) koje su smještene slijeva udesno (ćelija 1 je prva, ćelija 5 je posljednja). Ćelija 5 je završna. Na ćelijama 1 do 4 robot može poduzeti jednu od tri akcije: $a_1 = \text{lijevo}$, $a_2 = \text{desno}$, $a_3 = \text{sagni-se-i-pokupi-bocu}$. Boce se nalaze na ćelijama 2 i 4. Okolina robotu dodjeljuje nagrade kako slijedi. Za prelazak na ćeliju 5 robot dobiva 10 bodova. Za saginjanje i skupljanje boce robot dobiva 5 bodova, no ako se sagne i pokuša pokupiti bocu na ćeliji koja nema bocu, robot dobiva -5 bodova. Za sve ostale akcije isporučuje se nagrada življenja od -1 bod. U ovom svijetu robot uči optimalnu politiku algoritmom q-učenja. Neka su $\gamma = 1$ i $\alpha = 0.25$. U nekom trenutku naučene q-vrijednosti su sljedeće: $q(1, a_1) = 0$, $q(2, a_1) = 1$, $q(3, a_1) = 0$, $q(4, a_1) = 1$, $q(1, a_2) = 1$, $q(2, a_2) = 1$, $q(3, a_2) = 1$, $q(4, a_2) = 2$, $q(1, a_3) = 2$, $q(2, a_3) = -1$, $q(3, a_3) = -2$, $q(4, a_3) = 3$. Robot se nalazi na ćeliji 3 i poduzima akciju a_3 . **Koja će biti nova vrijednost za $q(3, a_3)$?**

- ☐ A -1 ☐ B 2 ☐ C -2.5 ☐ D 1.25

6. Prirodom inspirirani optimizacijski algoritmi (3 pitanja)

- 18 (T) Genetski algoritmi koriste operatore selekcije, križanja i mutacije te mehanizam elitizma kako bi pretražili prostor rješenja. **Što bi se dogodilo kada ne bismo koristili mehanizam elitizma?**

- ☐ A Povećala bi se prosječna vrijednost funkcije dobrote u populaciji
☐ B Porasla bi vjerojatnost da pretraga zapne u lokalnome optimumu
☐ C Algoritam bi generirao rješenja koja su neispravna (izvan intervala pretrage)
☐ D Algoritam bi mogao vratiti rješenje koje je lošije od nekog ranije generiranog rješenja

- 19 (R) Mravlji algoritmom uz parametre $\alpha = \beta = 1$ traži se put (hamiltonov ciklus) kroz graf koji se sastoji od 5 čvorova: A do E . Za bridove su poznate vrijednosti heurističkih informacija: $\eta_{AB} = 7$, $\eta_{AC} = 1$, $\eta_{BC} = 2$, $\eta_{BD} = 3$, $\eta_{CD} = 2$, $\eta_{DE} = 4$ i $\eta_{CE} = 3$. U nekoj iteraciji algoritma vrijednosti feromonskih tragova su: $\tau_{AB} = 1$, $\tau_{AC} = 3$, $\tau_{BC} = 2$, $\tau_{BD} = 2$, $\tau_{CD} = 1$, $\tau_{DE} = 2$ i $\tau_{CE} = 3$. Sve vrijednosti η i τ su ujedno i simetrične. Mrav konstrukciju puta započinje iz čvora A . Pretpostavite da će prilikom slučajnog odabira sljedećeg čvora mrav svaki puta odabrati onaj koji je najvjerojatniji (i u koji je dozvoljeno odnosno moguće prijeći). Odredite put koji će mrav konstruirati. **Koji je od sljedećih sljedova dio tog puta?**

☐ A $E \rightarrow C \rightarrow A$ ☐ B $B \rightarrow D \rightarrow C$ ☐ C $B \rightarrow C \rightarrow E$ ☐ D $D \rightarrow C \rightarrow E$

- 20 (P) Genetičkim algoritmom traži se minimum funkcije $f(x) = (x + \frac{1}{8})^2$ na intervalu $[-4, 2.3]$. Pri tome se koriste 6-bitovni kromosomi. Označimo s \hat{x} najbolje rješenje koje GA može pronaći, a s x^* vrijednost za koju funkcija poprima minimum. **Koliko iznosi $|x^* - \hat{x}|$?**

☐ A 0.3 ☐ B 0.025 ☐ C 0.625 ☐ D 0.25

Uvod u umjetnu inteligenciju – pismeni ispit (2022./2023.) – 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 (4 pitanja)

- 1** (P) Razmatramo dva igrača algoritma: jedan s heuristikom h_1 i dubinom pretraživanja d_1 te drugi s heuristikom h_2 i dubinom pretraživanja d_2 . Oba algoritma koriste strategiju minimax. **Kada će prvi algoritam češće pobjeđivati drugi algoritam?**

☐ A Ako $h_2 > h_1$ ☐ B Ako $h_1 = h_2$ i $d_1 > d_2$ ☐ C Ako $d_1 = d_2$ i $h_1 > h_2$ ☐ D Ako $h_1 = h_2$ i $d_1 < d_2$

- 2** (T) Algoritam minimax s dubinskim ograničenjem uobičajeno je izveden pomoću dvije, međusobno rekurzivne funkcije. Neka su to funkcije $\text{maxValue}(s, d)$ i $\text{minValue}(s, d)$, gdje je s stanje igre, a d je brojač razine u stablu igre. Početni poziv algoritma je $\text{maxValue}(s_0, k)$, gdje je s_0 trenutno stanje igre, a k je dubinsko ograničenje. Funkcija $\text{minValue}(s, d)$ koristi petlju s iteratorom $t \in \text{succ}(s)$ kako bi izračunala minimax-vrijednost m , koja je izvan petlje inicijalizirana na $+\infty$. **Kako glasi tijelo petlje u funkciji minValue?**

☐ A $m \leftarrow \min(m, \text{minValue}(t, d + 1))$ ☐ C $m \leftarrow \max(m, \text{minValue}(t, d - 1))$
☐ B $m \leftarrow \min(m, \text{maxValue}(t, d - 1))$ ☐ D $m \leftarrow \max(m, \text{maxValue}(t, d + 1))$

- 3** (P) Rješavamo problem pretraživanja prostora stanja za koji vrijedi $m = 9$, $d = 6$ i $b = 3$. Koristimo pretraživanje u širinu (BFS) i pretraživanje u dubinu (DFS). Za oba algoritma razmotrite najgori slučaj vremenske složenosti. **Koliko će u takvom slučaju algoritam DFS generirati više čvorova od algoritma BFS?**

☐ A 26241 ☐ B 28431 ☐ C 28395 ☐ D 26208

- 4** (R) Neka su definirani skup stanja $S = \{a, b, c, d, e, f\}$ i funkcija sljedbenika $\text{succ}(a) = \{(b, 1), (c, 2)\}$, $\text{succ}(b) = \text{succ}(f) = \emptyset$, $\text{succ}(c) = \{(d, 3), (e, 5)\}$, $\text{succ}(d) = \{(b, 8), (f, 5)\}$ te $\text{succ}(e) = \{(d, 1), (f, 2)\}$. Heurističke vrijednosti čvorova neka su $h(a) = 7$, $h(b) = 4$, $h(c) = 6$, $h(d) = h(e) = 2$, $h(f) = 0$. Početno stanje neka je a , a ciljno f . Izvršite pretraživanje algoritmom A^* , bilježeći pritom sadržaje liste otvorenih čvorova O i zatvorenih čvorova C u svakom koraku algoritma (u nultom koraku $O = [(a, 0)]$ i $C = \emptyset$). **Koji su sadržaji listi O i C nakon petog koraka izvođenja algoritma A^* ?**

☐ A Algoritam ne dostiže peti korak
☐ B $O = [(f, 10)]$, $C = \{(a, 0), (b, 1), (c, 2), (d, 5), (e, 7)\}$
☐ C $O = [(e, 7), (f, 10)]$, $C = \{(a, 0), (c, 2), (d, 5)\}$
☐ D $O = [(f, 9)]$, $C = \{(a, 0), (b, 1), (c, 2), (d, 5), (e, 7)\}$

2. Prikazivanje znanja i automatsko zaključivanje (4 pitanja)

- 5** (T) Relacija logičke posljedice između formula F i G definirana je na temelju vrijednosti istinitosti formula F i G kroz sve moguće interpretacije tih formula. Dokazivanje logičke posljedice od praktičnog je značaja za umjetnu inteligenciju. **Na koji način dokazujemo logičku posledicu u FOL?**

☐ A Pomoću ispravnih i potpunih pravila zaključivanja
☐ B Primjenom izravne rezolucije i faktorizacije
☐ C Pretvorbom u klauzalni oblik i skolemizacijom
☐ D Konstrukcijom tablice istinitosti i provjerom modela

6 (P) Ispravnost je važno svojstvo pravila zaključivanja. **Koje je od sljedećih pravila zaključivanja ispravno?**

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

7 (R) Zadane su sljedeće dvije premise:

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

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

- ☐ A $\exists x P(x) \vee P(b)$ ☐ B $\forall x Q(a, x)$ ☐ C $\exists x P(x) \wedge P(b)$ ☐ D $P(b)$

8 (R) Dani su FOL atomi $P(f(x), y, g(w, f(z)))$ i $P(y, f(g(a, z)), x)$, gdje je P predikat, a je konstanta, a f i g su funkcije. **Što je najopćenitiji zajednički unifikator ovih dvaju atoma?**

- ☐ A $P(f(g(a, b)), f(g(a, z)), g(w, b))$ ☐ C Atome nije moguće unificirati
☐ B $P(f(x), f(g(x, y)), g(a, b))$ ☐ D $P(f(g(a, b)), f(g(a, b)), g(a, b))$

3. Logičko programiranje i ekspertni sustavi (2 pitanja)

9 (T) Kao i sustavi formalne logike, ekspertni se sustavi temelje na formalnom prikazu znanja i izvođenju novog znanja primjenom pravila zaključivanja. Međutim, između tih sustava postoje razlike. **Po čemu se ekspertni sustavi razlikuju od općenitih sustava formalne logike?**

- ☐ A Za dokazivanje se koristi rezolucija opovrgavanjem, a ne izravna rezolucija
☐ B Znanje je prikazano u obliku ako-onda pravila, a ne Hornovih ili definitnih klauzula
☐ C Za dokazivanje se koriste logička pravila koja su potpuna, a ne samo ispravna
☐ D Znanje je prikazano u obliku činjenica i pravila, a ne proizvoljnih logičkih formula

10 (T) Izvođenje logičkih programa u Prologu odgovara dokazivanju teorema rezolucijom opovrgavanjem. Stanje izvođenja u Prologu definirano je stogom, koji početno sadrži atom upita. **Što sadrži Prologov stog u nastavku izvođenja programa?**

- ☐ A Činjenice iz logičkog programa ☐ C Atome iz glave logičkih pravila
☐ B Atome dviju roditeljskih klauzula ☐ D Atome iz tijela logičkih pravila

4. Modeliranje neizvjesnosti (2 pitanja)

11 (T) Bayesovo pravilo možemo koristiti za zaključivanje abdukcijom. **Što to znači?**

- ☐ A Na temelju opažene hipoteze zaključujemo o dokazu
☐ B Na temelju ostvarenog uzroka zaključujemo o hipotezi
☐ C Na temelju opaženih dokaza zaključujemo o uzroku
☐ D Na temelju ostvarenog uzroka zaključujemo o dokazima

12 (R) Nad univerzalnim skupom $\{a, b, c, d\}$ zadani su neizraziti skupovi $X = \{0.1/a + 0/b + 0.3/c + 1/d\}$, $Y = \{0.5/a + 0.4/b + 0/c + 0.2/d\}$ i $Z = \{1/a + 0/b + 0/c + 0.5/d\}$. Koristeći Zadehove operatore izvodimo neizraziti skup X tako da on odgovara jezičnome izrazu "*Ne X ili (Y i Z)*". **Kako glasi neizraziti skup X?**

- ☐ A $\{0.9/a + 0.4/b + 0/c + 0/d\}$ ☐ C $\{0.1/a + 1/b + 1/c + 0.5/d\}$
☐ B $\{0/a + 0.4/b + 0.3/c + 0.5/d\}$ ☐ D $\{0.5/a + 0/b + 0/c + 0/d\}$

5. Strojno učenje, umjetne neuronske mreže i podržano učenje (5 pitanja)

- 13** (R) Mali je Ivica svakog svakog ljeta u zadnjih sedam godina naučio jedan novi programski jezik. Svoja vrijedna iskustva sazeo 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	Eval (E)	Izvođenje (I)	Paradigma (P)	Tipovi (T)	y	i	Eval (E)	Izvođenje (I)	Paradigma (P)	Tipovi (T)	y
1	lijena	kompajler	imperativna	statička	0	5	striktna	interpreter	imperativna	statička	1
2	striktna	interpreter	deklarativna	dinamička	0	6	lijena	kompajler	hibridna	dinamička	1
3	lijena	kompajler	imperativna	dinamička	0	7	striktna	kompajler	hibridna	dinamička	1
4	lijena	interpreter	hibridna	statička	0						

Ovog ljeta Mali Ivica opet naučiti novi programski jezik, 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?** (Korijski čvor je čvor na prvoj razini.)

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

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

- 15** (P) Gradimo naivan Bayesov klasifikator za klasifikaciju poruka Twittera (tvitova) prema sentimentu. Svaki tvit želimo klasificirati u jednu od tri klase: pozitivan, negativan ili neutralan. Svaki tvit sastoji se od najviše 280 riječi i prikazujemo ga kao jedan binarni vektor značajki. Vektor se sastoji od 5000 značajki, i svaka značajka odgovara jednoj od 5000 riječi iz našeg vokabulara. Ako je značajka postavljena na 1, to znači da se dotična riječ pojavila u tvitu, inače je značajka postavljena na 0. Npr., ako je $x_{42} = 1$, onda to znači da se 42. riječ iz našeg vokabulara pojavila u tvitu. Učenje ovog modela svodi se na procjenu apriornih vjerojatnosti i izglednosti klase na temelju označenog skupa podataka. **Koliko ćemo ukupno vjerojatnosti trebati procijeniti za ovaj model?**

- ☐ A 569 ☐ B 1683 ☐ C 31683 ☐ D 30003

- 16** (T) Model Bayesovog klasifikatora učimo tako da procijenimo njegove parametre. Za to koristimo procjenu navjeće izglednosti s Laplaceovim zaglađivanjem. **U kojim situacijama je važno koristiti Laplaceovo zaglađivanje?**

- ☐ A Kada je skup za učenje velik ☐ C Kada je skup za provjeru malen
☐ B Kada je skup za učenje malen ☐ D Kada je skup za ispitivanje malen

- 17** (P) Rešetkasti svijet sastoji se od pet ćelija (numerirane 1 do 5) koje su smještene slijeva udesno (ćelija 1 je prva, ćelija 5 je posljednja). Ćelija 5 je završna. Na ćelijama 1 do 4 robot može poduzeti jednu od tri akcije: $a_1 = \text{lijevo}$, $a_2 = \text{desno}$, $a_3 = \text{sagni-se-i-pokupi-bocu}$. Boce se nalaze na ćelijama 2 i 4. Okolina robotu dodjeljuje nagrade kako slijedi. Za prelazak na ćeliju 5 robot dobiva 10 bodova. Za saginjanje i skupljanje boce robot dobiva 5 bodova, no ako se sagne i pokuša pokupiti bocu na ćeliji koja nema bocu, robot dobiva -5 bodova. Za sve ostale akcije isporučuje se nagrada življenja od -1 bod. U ovom svijetu robot uči optimalnu politiku algoritmom q-učenja. Neka su $\gamma = 1$ i $\alpha = 0.25$. U nekom trenutku naučene q-vrijednosti su sljedeće: $q(1, a_1) = 0$, $q(2, a_1) = 1$, $q(3, a_1) = 0$, $q(4, a_1) = 1$, $q(1, a_2) = 1$, $q(2, a_2) = 1$, $q(3, a_2) = 1$, $q(4, a_2) = 2$, $q(1, a_3) = 2$, $q(2, a_3) = -1$, $q(3, a_3) = -2$, $q(4, a_3) = 3$. Robot se nalazi na ćeliji 3 i poduzima akciju a_1 . **Koja će biti nova vrijednost za $q(3, a_1)$?**

- ☐ A -1 ☐ B 2.75 ☐ C 0 ☐ D 1.25

6. Prirodom inspirirani optimizacijski algoritmi (3 pitanja)

- 18 (R) Genetskim algoritmom traži se maksimum funkcije $f(x, y) = 3x^2 + 5y^2$. Svaka varijabla kodira se s 4 bita, pri čemu se x pretražuje iz intervala $[-4, 11]$, a y iz intervala $[-6, 9]$. U kromosomu najprije dolaze bitovi koji kodiraju x pa potom bitovi koji kodiraju y . Za odabir roditelja koristi se selekcija proporcionalna dobroti (engl. *roulette wheel selection*). Populacija se sastoji od sljedeće četiri jedinke:

$$K_1 = 10110011, \quad K_2 = 01010101, \quad K_3 = 00101010, \quad K_4 = 00010001$$

Koristi se križanje s jednom točkom prijeloma (točno na polovici kromosoma). Pretpostavite da se u trenutačnoj iteraciji algoritma slučajno odaberu baš dvije najmanje vjerojatne jedinke te da pri izgradnji djece mutacija igrom slučaja ne promijeni niti jedan bit. Označimo s F zbroj vjerojatnosti odabira odabranih jedinki, a s G dobrotu djeteta koje će biti umetnuto u novu populaciju (ako križanjem/mutacijom nastaje više djece, samo najbolje dijete ide dalje). **Što od sljedećega vrijedi za F i G ?**

- ☐ A $F > 0.1, 5 < G < 50$ ☐ B $F < 0.3, 60 < G < 100$ ☐ C $F > 0.1, G > 90$ ☐ D $F < 0.3, G < 70$

- 19 (P) Genetičkim algoritmom traži se minimum funkcije $f(x) = (x - \frac{11}{8})^2$ na intervalu $[-2.5, 3.8]$. Pri tome se koriste 6-bitovni kromosomi. Označimo s \hat{x} najbolje rješenje koje GA može pronaći, a s x^* vrijednost za koju funkcija poprima minimum. **Koliko iznosi $|x^* - \hat{x}|$?**

- ☐ A 0.025 ☐ B 0.3 ☐ C 0.125 ☐ D 1.0

- 20 (T) Genetski algoritmi koriste operatore selekcije, križanja i mutacije te mehanizam elitizma kako bi pretražili prostor rješenja. **Što bi se dogodilo kada ne bismo koristili mehanizam elitizma?**

- ☐ A Algoritam bi generirao rješenja koja su neispravna (izvan intervala pretrage)
☐ B Algoritam bi mogao vratiti rješenje koje je lošije od nekog ranije generiranog rješenja
☐ C Porasla bi vjerojatnost da pretraga zapne u lokalnome optimumu
☐ D Povećala bi se prosječna vrijednost funkcije dobrote u populaciji

Grupa																				
	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0
A	C	D	C	C	A	C	A	B	D	A	D	B	B	C	A	C	D	B	D	A
B	C	A	B	C	B	C	B	B	D	A	D	D	B	B	C	B	D	A	B	C
C	D	D	D	A	A	C	C	C	A	B	A	D	A	A	A	C	A	B	A	D
D	A	B	C	D	D	C	A	D	C	B	B	B	A	C	B	A	A	C	D	D
E	C	B	B	D	A	B	C	C	A	A	A	D	D	D	B	C	C	C	B	C
F	B	A	C	A	B	B	D	C	D	C	D	C	B	B	B	C	B	C	A	C
G	C	D	A	D	A	C	C	B	A	B	B	B	C	C	D	B	C	D	A	B
H	B	B	D	D	A	D	A	C	D	D	C	A	C	C	D	B	C	B	A	B