

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) Stanje s slagalice 3×3 neka je $[[1, 5, 2], [4, \square, 3], [7, 8, 6]]$, dok je ciljno stanje $[[1, 2, 3], [4, 5, 6], [7, 8, \square]]$. Neka je h optimistična heuristika. Među ponuđenim vrijednostima, koju vrijednost $h(s)$ može poprimiti, a da je najviše obaviještena?

☐ A 8 ☐ B 0 ☐ C 2 ☐ D 3

- 2** (R) Na skupu stanja $S = \{a, b, c, d, e, f\}$ definirana je funkcija sljedbenika $\text{succ}(a) = \{(b, 5), (c, 2)\}$, $\text{succ}(b) = \{(e, 5)\}$, $\text{succ}(c) = \{(a, 2), (b, 1), (d, 3), (f, 11)\}$, $\text{succ}(d) = \{(e, 4)\}$, $\text{succ}(e) = \{(f, 2)\}$, $\text{succ}(f) = \emptyset$. Početno stanje je a , a ciljno f . Heurističke vrijednosti stanja neka su $h(a) = 9, h(b) = 6, h(c) = 5, h(d) = 6, h(e) = 2, h(f) = 0$. **Koliko koraka treba algoritmu A^* da pronade optimalan put u odnosu na broj koraka koji algoritmu pretraživanja u širinu treba da pronade bilo kakav put?** Napomene: 1 korak = 1 ispitivanje čvora (uzmite u obzir da broj proširivanja čvorova ne mora odgovarati ovako definiranom broju koraka). Pretpostavite leksikografski poredak između čvorova.

☐ A Algoritmu A^* treba 2 koraka manje ☐ C Algoritmu A^* treba 3 koraka manje
☐ B Algoritmu A^* treba 2 koraka više ☐ D Algoritmu A^* treba 3 koraka više

- 3** (R) Razmatramo igru dvaju minimax-igrača. Stablo igre definirano je prijelazima $\text{succ}(A) = \{B, C, D\}$, $\text{succ}(B) = \{E, F\}$, $\text{succ}(C) = \{G, H\}$, $\text{succ}(D) = \{I, J\}$, $\text{succ}(F) = \text{succ}(G) = \{K, L\}$, $\text{succ}(E) = \text{succ}(H) = \text{succ}(I) = \{M, N\}$ te $\text{succ}(J) = \{O, P\}$. Igrač MAX koristi heuristiku h_1 , a igrač MIN heuristiku h_2 . Vrijednosti heuristika su sljedeće (obje heuristike definiraju dobit dotičnog igrača):

	E	F	G	H	I	J	K	L	M	N	O	P
h_1	1	-2	3	2	3	5	-1	0	-2	-1	0	2
h_2	0	5	2	4	-1	2	3	2	-3	-2	1	-4

Oba igrača pretražuju najviše dva poteza unaprijed (jedan svoj i jedan suparnički). Igru započinje igrač MAX u stanju A . **Kroz koja stanja će se razvijati igra?**

☐ A A, D, I, \dots ☐ B A, C, H, \dots ☐ C A, D, J, \dots ☐ D A, C, G, \dots

- 4** (T) Algoritam A^* oslanja se na heurističku funkciju. Svojstva heurističke funkcije izravno utječu na svojstva tog algoritma. **Što je od sljedećeg točno?**

- ☐ A Ako heuristika nije konzistentna, algoritam A^* ne nalazi optimalno rješenje
☐ B Ako je heuristika konzistentna, algoritam A^* ne otvara jednom zatvorene čvorove
☐ C Ako je heuristika optimistična, algoritam A^* ima linearnu prostornu složenost
☐ D Ako heuristika nije optimistična, algoritam A^* nije potpun

2. Prikazivanje znanja i automatsko zaključivanje (4 pitanja)

- 5** (T) Razmotrite logičku formulu $\forall x (Paran(x) \rightarrow Neparan(add(x, 1)))$. **Što je semantičko svojstvo ove formule, i zašto?**

- ☐ A Formula je zadovoljiva, jer za nju postoji model
☐ B Formula je neistinita, jer x ne mora nužno biti broj
☐ C Formula je valjana, jer je sljedbenik parnog broja neparan broj
☐ D Formula je istinita, jer je sljedbenik parnog broja neparan broj

6 (T) Rezolucijsko zaključivanje u FOL oslanja se na algoritam najopćenitijeg zajedničkog unifikatora (MGU). U kojem se trenutku primjenjuje algoritam MGU?

- ☐ A Nakon skolemizacije, a prije standardizacije ☐ C Prije uvođenja negiranog cilja u skup klauzula
☐ B Nakon razrješavanja klauzula, a prije faktORIZACIJE ☐ D Prije svake primjene rezolucijskog pravila

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

- ☐ A 74 ☐ B 119 ☐ C 104 ☐ D 60

8 (R) Želimo dokazati sljedeću relaciju deduktivne posljedice:

$$\forall x \left(P(x, a) \rightarrow \neg \forall y (Q(y) \rightarrow \exists z R(z, y)) \right), \forall x \exists y \left(Q(y) \rightarrow R(y, x) \right) \vdash \exists x \neg P(c, x).$$

Pretvorite premise i negaciju ciljne formule u klauzalni oblik te primijenite rezoluciju opovrgavanjem uz strategiju skupa potpore (SoS). **Koliko klauzula ulazi u postupak te je li cilj dokaziv i u koliko rezolucijskih koraka?**

- ☐ A 4 klauzule; nije dokazivo ☐ C 5 klauzula; dokazivo u 3 koraka
☐ B 4 klauzule; dokazivo u 3 koraka ☐ D 4 klauzule; dokazivo u 4 koraka

3. Logičko programiranje i ekspertni sustavi (2 pitanja)

9 (T) Kao i ekspertni sustavi, Prolog također zaključuje na temelju pravila. Međutim, u tom pogledu između ekspertnih sustava i Prologa postoji razlika. **Po čemu se ekspertni sustavi i Prolog razlikuju?**

- ☐ A Prolog u pravilima može koristiti varijable, dok ekspertni sustavi varijable ne koriste
☐ B Neki ekspertni sustavi mogu koristiti negaciju uvjeta u pravilima, dok Prolog ne koristi negaciju
☐ C Prolog koristi prioritete pravila, dok ekspertni sustavi pravila aktiviraju redoslijedom kojim su navedena
☐ D Neki ekspertni sustavi zaključuju ulančavanjem unaprijed, dok Prolog uvijek koristi ulančavanje unazad

10 (R) Razmatramo sljedeći program u Prologu:

```
roditelj(kronos, zeus).  
roditelj(zeus, minos).  
roditelj(europa, minos).  
potomak(X, Y) :- roditelj(Y, X).  
potomak(X, Y) :- roditelj(Y, Z), potomak(X, Z).  
bog(kronos).  
bog(X) :- potomak(X, Y), bog(Y).  
polubog(X) :- potomak(X, Y), not(bog(Y)).
```

Nad ovako definiranom bazom izvodimo upit `polubog(minos)`. Nacrtajte Prologovo stablo dokaza za ovaj upit do trenutka drugog povratka u pretraživanju. Pritom za dokazivanje negiranog cilja koristite dva čvora (jedan za `not(P(X))` i drugi za `P(X)`), dok prazni stog ne brojite kao čvor. **Koliko čvorova ima stablo dokaza neposredno prije drugog povratka u pretraživanju?**

- ☐ A 12 ☐ B 10 ☐ C 8 ☐ D 6

4. Modeliranje neizvjesnosti (2 pitanja)

- 11** (T) Generalizirani modus ponens (GMP) koristi se za zaključivanje u neizrazitoj logici. Kao i modus ponens, GMP se sastoji od premise, antedecenta, konzekvensa, implikacije između antedecenta i konzekvensa, te zaključka. **Na koji su način modelirane komponente GMP-a?**

- ☐ A Premisa je apriorna vjerojatnost, a implikacija i zaključak su uvjetne vjerojatnosti
☐ B Premisa je neizraziti skup, zaključak je neizrazita relacija, a implikacija je max-min kompozicija
☐ C Implikacija je neizraziti skup, a konzekvens i zaključak su max-min kompozicija
☐ D Premisa, antedecent, konzekvens i zaključak su neizraziti skupovi, a implikacija je neizrazita relacija

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

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

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	Evaluacija	Izvođenje	Paradigma	Provjera tipova	y
1	lijena	kompajler	imperativna	statička	0
2	striktna	interpreter	deklarativna	dinamička	0
3	lijena	kompajler	imperativna	dinamička	0
4	lijena	interpreter	hibridna	statička	1
5	striktna	interpreter	imperativna	statička	1
6	lijena	kompajler	hibridna	dinamička	1
7	striktna	kompajler	hibridna	dinamička	1

Ovog ljeta Mali Ivica želi puno jesti i spavati te opet naučiti novi programski jezik. U užem je izboru jezik x sa sljedećim karakteristikama: $x = (\text{striktna}, \text{interpreter}, \text{hibridna}, \text{dinamička})$. Međutim, ovog puta Mali bi Ivica volio unaprijed znati hoće li mu se dotični programski jezik svidjeti, tako da ne gubi cijelo ljeto bezveze. Pomozite Malom Ivici te na gornji skup primjera primjenite naivan Bayesov klasifikator s Laplaceovim zaglađivanjem "dodaj jedan". **Koliko iznosi vjerojatnost da bi se Malom Ivici programski jezik x svidio?**

- ☐ A 0.694 ☐ B 0.856 ☐ C 0.799 ☐ D 0.431

- 14** (P) Radimo svoju implementaciju algoritma ID3. Kako bismo spriječili da se model prenauci, implementirali smo i podrezivanje stabla na dubini k . Nažalost, kod implementacije funkcije informacijske dobiti (IG) potkrala nam se mala pogreška: zaboravili smo negirati vrijednost pri izračunu entropije $E(D)$ skupa primjera D . Dakle, umjesto da izračunava $E(D)$, naša implementacija izračunava $-E(D)$. Neka je M_1^k stablo odluke koje dobivamo učenjem takvim pogrešno implementiranim algoritmom ID3, podržano na neku konačnu dubinu k . Neka je M_2^k stablo koje bismo dobili da smo algoritam ID3 implementirali ispravno i naučeno stablo podrezali na neku konačnu dubinu k . Ako $k = \infty$, onda to znači da stablo ne podrezujemo. Neka je $E_u(M)$ pogreška učenja modela M , a $E_p(M)$ pogreška modela M na skupu za provjeru. **Što od sljedećeg možemo očekivati da vrijedi?**

- ☐ A $E_p(M_2^k) = E_u(M_2^k)$ ☐ B $E_p(M_1^\infty) < E_u(M_1^\infty)$ ☐ C $E_u(M_2^k) < E_u(M_2^\infty)$ ☐ D $E_p(M_1^k) > E_p(M_2^k)$

- 15** (T) Umjetni neuron TLU (kakav koristi perceptron) za svaku ulaznu značajku x_i ima jednu težinu w_i , ali ima i jednu dodatnu težinu, w_0 . **Koja je uloga težine w_0 ?**

- ☐ A Ako je umnožak značajki i pripadnih težina veći od $-w_0$, neuron na izlazu daje +1
☐ B Ako je težina w_0 pozitivna, korekcija Δw_0 je negativna, inače je pozitivna
☐ C Ako je težina w_0 negativna, a sve ostale težine su različite od nule, izlaz neurona je -1
☐ D Ako je zbroj svih težina veći od w_0 , neuron na izlazu daje pozitivnu vrijednost

- 16 (R) Umjetnu neuronsku mrežu arhitekture $1 \times 2 \times 2$ učimo algoritmom propagacije pogreške unatrag. U nekom koraku algoritam ažurira težine na temelju primjera $(1, (1, 0))$. Vrijednosti svih težina u tom koraku su:

$$w_{01}^{(1)} = 0.1, w_{11}^{(1)} = 0.2, w_{02}^{(1)} = 0.5, w_{12}^{(1)} = 0.6, w_{01}^{(2)} = 0.1, w_{11}^{(2)} = 0.2, w_{21}^{(2)} = 0.3, w_{02}^{(2)} = 0.2, w_{12}^{(2)} = 0.3, w_{22}^{(2)} = 0.4$$

Svi neuroni koriste sigmoidnu prijenosnu funkciju, a stopa učenja je 10. **Koliko iznosi korekcija koja će biti pridodana težini $w_{12}^{(1)}$?**

- ☐ A +0.065732 ☐ B +0.035152 ☐ C -0.147551 ☐ D -0.058547

- 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 1.5 ☐ B 1.25 ☐ C 3 ☐ D 2.75

6. Prirodom inspirirani optimizacijski algoritmi (3 pitanja)

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

- ☐ A 2 ☐ B 15 ☐ C 6 ☐ D 13

- 19 (P) Ako kod algoritma Ant System postavimo τ_0 na vrijednost koja je puno manja od količine feromonskih tragova koju deponira jedan mrav, **što će biti posljedica?**

- ☐ A Algoritam će dugo vremena istraživati nasumične puteve, prije no što se mravi fokusiraju na neke puteve
☐ B Algoritam će rapidno konvergirati prema globalnom optimumu
☐ C Isparavanje feromonskih tragova će biti preveliko
☐ D Algoritam će izgubiti mogućnost istraživanja kvalitetnih rješenja

- 20 (T) Različite instance problema trgovačkog putnika rješavamo genetičkim algoritmom (GA) i mravljim algoritmom (ACO). Na ove je optimizacijske algoritme primjenjiv "teorem o nebesplatnome ručku" (engl. *No free lunch theorem*). **Što to konkretno znači?**

- ☐ A Ovisno o instanci problema, jedan od algoritama češće će nalaziti bolje rješenje
☐ B Ako postoji globalno optimalno rješenje, niti GA niti ACO ga ne pronalaze
☐ C Vremenska složenost oba algoritma u prosjeku je eksponencijalna
☐ D U prosjeku jedan od ova dva algoritma češće zapinje u lokalnom 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** (P) Stanje s slagalice 3×3 neka je $[[1, 2, \square], [4, 5, 3], [7, 8, 6]]$, dok je ciljno stanje $[[1, 2, 3], [4, 5, 6], [7, 8, \square]]$. Neka je h optimistična heuristika. Među ponuđenim vrijednostima, koju vrijednost $h(s)$ može poprimiti, a da je najviše obaviještena?

☐ A 1 ☐ B 4 ☐ C 0 ☐ D 8

- 2** (T) Algoritam A^* oslanja se na heurističku funkciju. Svojstva heurističke funkcije izravno utječu na svojstva tog algoritma. Što je od sljedećeg točno?

- ☐ A Ako heuristika nije optimistična, algoritam A^* nije potpun
☐ B Ako je heuristika konzistentna, algoritam A^* ne otvara jednom zatvorene čvorove
☐ C Ako je heuristika optimistična, algoritam A^* ima linearnu prostornu složenost
☐ D Ako heuristika nije konzistentna, algoritam A^* ne nalazi optimalno rješenje

- 3** (R) Na skupu stanja $S = \{a, b, c, d, e, f\}$ definirana je funkcija sljedbenika $\text{succ}(a) = \{(b, 5), (c, 2)\}$, $\text{succ}(b) = \{(e, 5)\}$, $\text{succ}(c) = \{(a, 2), (b, 1), (d, 3), (f, 11)\}$, $\text{succ}(d) = \{(e, 4)\}$, $\text{succ}(e) = \{(f, 2)\}$, $\text{succ}(f) = \emptyset$. Početno stanje je a , a ciljno f . Heurističke vrijednosti stanja neka su $h(a) = 9, h(b) = 6, h(c) = 5, h(d) = 6, h(e) = 2, h(f) = 0$. **Koliko koraka treba algoritmu A^* da pronađe optimalan put u odnosu na broj koraka koji algoritmu pretraživanja u širinu treba da pronađe bilo kakav put?** Napomene: 1 korak = 1 ispitivanje čvora (uzmite u obzir da broj proširivanja čvorova ne mora odgovarati ovako definiranom broju koraka). Pretpostavite leksikografski poredak između čvorova.

- ☐ A Algoritmu A^* treba 2 koraka manje ☐ C Algoritmu A^* treba 5 koraka manje
☐ B Algoritmu A^* treba 3 koraka manje ☐ D Oba algoritma trebaju isti broj koraka

- 4** (R) Razmatramo igru dvaju minimax-igrača. Stablo igre definirano je prijelazima $\text{succ}(A) = \{B, C, D\}$, $\text{succ}(B) = \{E, F\}$, $\text{succ}(C) = \{G, H\}$, $\text{succ}(D) = \{I, J\}$, $\text{succ}(F) = \text{succ}(G) = \{K, L\}$, $\text{succ}(E) = \text{succ}(H) = \text{succ}(I) = \{M, N\}$ te $\text{succ}(J) = \{O, P\}$. Igrač MAX koristi heuristiku h_1 , a igrač MIN heuristiku h_2 . Vrijednosti heuristika su sljedeće (obje heuristike definiraju dobit dotičnog igrača):

	E	F	G	H	I	J	K	L	M	N	O	P
h_1	1	-2	3	2	3	5	-1	0	-2	-1	0	2
h_2	0	5	2	4	-1	2	3	2	1	-4	-3	-2

Oba igrača pretražuju najviše dva poteza unaprijed (jedan svoj i jedan suparnički). Igru započinje igrač MAX u stanju A . **Kroz koja stanja će se razvijati igra?**

- ☐ A A, C, H, \dots ☐ B A, D, I, \dots ☐ C A, C, G, \dots ☐ D A, D, J, \dots

2. Prikazivanje znanja i automatsko zaključivanje (4 pitanja)

- 5** (T) Razmotrite logičku formulu $\forall x (Paran(x) \rightarrow Neparan(add(x, 1)))$. Što je semantičko svojstvo ove formule, i zašto?

- ☐ A Formula je istinita, jer je sljedbenik parnog broja neparan broj
☐ B Formula je neistinita, jer x ne mora nužno biti broj
☐ C Formula je valjana, jer je sljedbenik parnog broja neparan broj
☐ D Formula je zadovoljiva, jer za nju postoji model

- 6 (R) Želimo dokazati sljedeću relaciju deduktivne posljedice:

$$\forall x \left(P(x, a) \rightarrow \neg \forall y (Q(y) \rightarrow \exists z R(z, y)) \right), \forall x \exists y (Q(y) \rightarrow R(y, x)) \vdash \exists x \neg (P(x, a) \wedge R(x, c)).$$

Pretvorite premise i negaciju ciljne formule u klauzalni oblik te primijenite rezoluciju opovrgavanjem uz strategiju skupa potpore (SoS). **Koliko klauzula ulazi u postupak te je li cilj dokaziv i u koliko rezolucijskih koraka?**

- ☐ A 3 klauzule; dokazivo u 2 koraka ☐ C 5 klauzula; dokazivo u 5 koraka
☐ B 5 klauzula; nije dokazivo ☐ D 5 klauzula; dokazivo u 3 koraka

- 7 (T) Rezolucijsko zaključivanje u FOL oslanja se na algoritam najopćenitijeg zajedničkog unifikatora (MGU). U kojem se trenutku primjenjuje algoritam MGU?

- ☐ A Prije uvođenja negiranog cilja u skup klauzula ☐ C Prije svake primjene rezolucijskog pravila
☐ B Nakon razrješavanja klauzula, a prije faktorizacije ☐ D Nakon skolemizacije, a prije standardizacije

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

- ☐ A 104 ☐ B 60 ☐ C 74 ☐ D 119

3. Logičko programiranje i ekspertni sustavi (2 pitanja)

- 9 (T) Kao i ekspertni sustavi, Prolog također zaključuje na temelju pravila. Međutim, u tom pogledu između ekspertnih sustava i Prologa postoji razlika. **Po čemu se ekspertni sustavi i Prolog razlikuju?**

- ☐ A Prolog koristi prioritete pravila, dok ekspertni sustavi pravila aktiviraju redoslijedom kojim su navedena
☐ B Prolog u pravilima može koristiti varijable, dok ekspertni sustavi varijable ne koriste
☐ C Neki ekspertni sustavi mogu koristiti negaciju uvjeta u pravilima, dok Prolog ne koristi negaciju
☐ D Neki ekspertni sustavi zaključuju ulančavanjem unaprijed, dok Prolog uvijek koristi ulančavanje unazad

- 10 (R) Razmatramo sljedeći program u Prologu:

```
roditelj(kronos, zeus).  
roditelj(zeus, minos).  
roditelj(europa, minos).  
potomak(X, Y) :- roditelj(Y, X).  
potomak(X, Y) :- roditelj(Y, Z), potomak(X, Z).  
bog(kronos).  
bog(X) :- potomak(X, Y), bog(Y).  
polubog(X) :- potomak(X, Y), not(bog(Y)).
```

Nad ovako definiranom bazom izvodimo upit `polubog(minos)`. Nacrtajte Prologovo stablo dokaza za ovaj upit do trenutka drugog povratka u pretraživanju. Pritom za dokazivanje negiranog cilja koristite dva čvora (jedan za `not(P(X))` i drugi za `P(X)`), dok prazni stog ne brojite kao čvor. **Koliko čvorova ima stablo dokaza neposredno prije drugog povratka u pretraživanju?**

- ☐ A 6 ☐ B 8 ☐ C 10 ☐ D 12

4. Modeliranje neizvjesnosti (2 pitanja)

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

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

- 12** (T) Generalizirani modus ponens (GMP) koristi se za zaključivanje u neizrazitoj logici. Kao i modus ponens, GMP se sastoji od premise, antecedenta, konzekvensa, implikacije između antecedenta i konzekvensa, te zaključka. **Na koji su način modelirane komponente GMP-a?**

- ☐ A Premisa je neizraziti skup, zaključak je neizrazita relacija, a implikacija je max-min kompozicija
☐ B Implikacija je neizraziti skup, a konzekvens i zaključak su max-min kompozicija
☐ C Premisa, antecedent, konzekvens i zaključak su neizraziti skupovi, a implikacija je neizrazita relacija
☐ D Premisa je apriorna vjerojatnost, a implikacija i zaključak su uvjetne vjerojatnosti

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	Evaluacija	Izvođenje	Paradigma	Provjera tipova	y
1	lijena	kompajler	imperativna	statička	0
2	striktna	interpreter	deklarativna	dinamička	0
3	lijena	kompajler	imperativna	dinamička	0
4	lijena	interpreter	hibridna	statička	0
5	striktna	interpreter	imperativna	statička	1
6	lijena	kompajler	hibridna	dinamička	1
7	striktna	kompajler	hibridna	dinamička	1

Ovog ljeta Mali Ivica želi puno jesti i spavati te opet naučiti novi programski jezik. U užem je izboru jezik x sa sljedećim karakteristikama: $x = (\text{striktna}, \text{interpreter}, \text{hibridna}, \text{dinamička})$. Međutim, ovog puta Mali bi Ivica volio unaprijed znati hoće li mu se dotični programski jezik svidjeti, tako da ne gubi cijelo ljeto bezveze. Pomozite Malom Ivici te na gornji skup primjera primijenite naivan Bayesov klasifikator s Laplaceovim zaglađivanjem "dodaj jedan". **Koliko iznosi vjerojatnost da bi se Malom Ivici programski jezik x svidio?**

- ☐ A 0.856 ☐ B 0.694 ☐ C 0.431 ☐ D 0.799

- 14** (P) Radimo svoju implementaciju algoritma ID3. Kako bismo spriječili da se model prenauci, implementirali smo i podrezivanje stabla na dubini k . Nažalost, kod implementacije funkcije informacijske dobiti (IG) potkrala nam se mala pogreška: zaboravili smo negirati vrijednost pri izračunu entropije $E(D)$ skupa primjera D . Dakle, umjesto da izračunava $E(D)$, naša implementacija izračunava $-E(D)$. Neka je M_1^k stablo odluke koje dobivamo učenjem takvim pogrešno implementiranim algoritmom ID3, podrezano na neku konačnu dubinu k . Neka je M_2^k stablo koje bismo dobili da smo algoritam ID3 implementirali ispravno i naučeno stablo podrezali na neku konačnu dubinu k . Ako $k = \infty$, onda to znači da stablo ne podrezujemo. Neka je $E_u(M)$ pogreška učenja modela M , a $E_p(M)$ pogreška modela M na skupu za provjeru. **Što od sljedećeg možemo očekivati da vrijedi?**

- ☐ A $E_u(M_2^k) = E_u(M_2^\infty)$ ☐ B $E_p(M_2^k) = E_u(M_2^k)$ ☐ C $E_p(M_1^\infty) < E_p(M_2^\infty)$ ☐ D $E_u(M_1^\infty) = E_u(M_2^\infty)$

- 15** (T) Umjetni neuron TLU (kakav koristi perceptron) za svaku ulaznu značajku x_i ima jednu težinu w_i , ali ima i jednu dodatnu težinu, w_0 . **Koja je uloga težine w_0 ?**

- ☐ A Ako je težina w_0 negativna, a sve ostale težine su različite od nule, izlaz neurona je -1
☐ B Ako je umnožak značajki i pripadnih težina veći od $-w_0$, neuron na izlazu daje $+1$
☐ C Ako je zbroj svih težina veći od w_0 , neuron na izlazu daje pozitivnu vrijednost
☐ D Ako je težina w_0 pozitivna, korekcija Δw_0 je negativna, inače je pozitivna

- 16 (R) Umjetnu neuronsku mrežu arhitekture $1 \times 2 \times 2$ učimo algoritmom propagacije pogreške unatrag. U nekom koraku algoritam ažurira težine na temelju primjera $(1, (1, 0))$. Vrijednosti svih težina u tom koraku su:

$$w_{01}^{(1)} = 0.1, w_{11}^{(1)} = 0.2, w_{02}^{(1)} = 0.5, w_{12}^{(1)} = 0.6, w_{01}^{(2)} = 0.1, w_{11}^{(2)} = 0.2, w_{21}^{(2)} = 0.3, w_{02}^{(2)} = 0.2, w_{12}^{(2)} = 0.3, w_{22}^{(2)} = 0.4$$

Svi neuroni koriste sigmoidnu prijenosnu funkciju, a stopa učenja je 10. **Koliko iznosi korekcija koja će biti pridodana težini $w_{12}^{(1)}$?**

- ☐ A +0.065732 ☐ B +0.035152 ☐ C -0.058547 ☐ D -0.147551

- 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 3 ☐ C 1.25 ☐ D 1.5

6. Prirodom inspirirani optimizacijski algoritmi (3 pitanja)

- 18 (P) Ako kod algoritma Ant System postavimo τ_0 na vrijednost koja je puno veća od količine feromonskih tragova koju deponira jedan mrav, **što će biti posljedica?**

- ☐ A Algoritam će izgubiti mogućnost istraživanja kvalitetnih rješenja
☐ B Algoritam će rapidno konvergirati prema globalnom optimumu
☐ C Isparavanje feromonskih tragova će biti preveliko
☐ D Algoritam će dugo vremena istraživati nasumične puteve, prije no što se mravi fokusiraju na neke puteve

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

- ☐ A 20 ☐ B 11 ☐ C 7 ☐ D 18

- 20 (T) Različite instance problema trgovačkog putnika rješavamo genetičkim algoritmom (GA) i mravljim algoritmom (ACO). Na ove je optimizacijske algoritme primjenjiv "teorem o nebesplatnome ručku" (engl. *No free lunch theorem*). **Što to konkretno znači?**

- ☐ A Vremenska složenost oba algoritma u prosjeku je eksponencijalna
☐ B Ovisno o instanci problema, jedan od algoritama češće će nalaziti bolje rješenje
☐ C Ako postoji globalno optimalno rješenje, niti GA niti ACO ga ne pronalaze
☐ D U prosjeku jedan od ova dva algoritma češće zapinje u lokalnom 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 A^* oslanja se na heurističku funkciju. Svojstva heurističke funkcije izravno utječu na svojstva tog algoritma. Što je od sljedećeg točno?

- ☐ A Ako je heuristika konzistentna, algoritam A^* ne otvara jednom zatvorene čvorove
☐ B Ako heuristika nije optimistična, algoritam A^* nije potpun
☐ C Ako je heuristika optimistična, algoritam A^* ima linearnu prostornu složenost
☐ D Ako heuristika nije konzistentna, algoritam A^* ne nalazi optimalno rješenje

- 2** (R) Razmatramo igru dvaju minimax-igrača. Stablo igre definirano je prijelazima $\text{succ}(A) = \{B, C, D\}$, $\text{succ}(B) = \{E, F\}$, $\text{succ}(C) = \{G, H\}$, $\text{succ}(D) = \{I, J\}$, $\text{succ}(F) = \text{succ}(G) = \{K, L\}$, $\text{succ}(E) = \text{succ}(H) = \text{succ}(I) = \{M, N\}$ te $\text{succ}(J) = \{O, P\}$. Igrač MAX koristi heuristiku h_1 , a igrač MIN heuristiku h_2 . Vrijednosti heuristika su sljedeće (obje heuristike definiraju dobit dotičnog igrača):

	E	F	G	H	I	J	K	L	M	N	O	P
h_1	1	-2	3	2	3	5	-1	0	-2	-1	0	2
h_2	0	5	2	4	-1	2	3	2	1	-4	-3	-2

Oba igrača pretražuju najviše dva poteza unaprijed (jedan svoj i jedan suparnički). Igru započinje igrač MAX u stanju A . **Kroz koja stanja će se razvijati igra?**

- ☐ A A, C, H, \dots ☐ B A, D, J, \dots ☐ C A, C, G, \dots ☐ D A, D, I, \dots

- 3** (R) Na skupu stanja $S = \{a, b, c, d, e, f\}$ definirana je funkcija sljedbenika $\text{succ}(a) = \{(b, 5), (c, 2)\}$, $\text{succ}(b) = \{(e, 5)\}$, $\text{succ}(c) = \{(a, 2), (b, 1), (d, 3), (f, 11)\}$, $\text{succ}(d) = \{(e, 4)\}$, $\text{succ}(e) = \{(f, 2)\}$, $\text{succ}(f) = \emptyset$. Početno stanje je a , a ciljno f . Heurističke vrijednosti stanja neka su $h(a) = 9, h(b) = 6, h(c) = 5, h(d) = 6, h(e) = 2, h(f) = 0$. **Koliko koraka treba algoritmu A^* da pronađe optimalan put u odnosu na broj koraka koji algoritmu iterativnog pretraživanja u dubinu treba da pronađe bilo kakav put?** Napomene: 1 korak = 1 ispitivanje čvora (uzmite u obzir da broj proširivanja čvorova ne mora odgovarati ovako definiranom broju koraka). Pretpostavite leksikografski poredak između čvorova.

- ☐ A Algoritmu A^* treba 5 koraka manje ☐ C Algoritmu A^* treba 1 korak više
☐ B Algoritmu A^* treba 4 koraka više ☐ D Algoritmu A^* treba 7 koraka manje

- 4** (P) Stanje s slagalice 3×3 neka je $[[1, 5, 2], [4, \square, 3], [7, 8, 6]]$, dok je ciljno stanje $[[1, 2, 3], [4, 5, 6], [7, 8, \square]]$. Neka je h optimistična heuristika. **Među ponuđenim vrijednostima, koju vrijednost $h(s)$ može poprimiti, a da je najviše obaviještena?**

- ☐ A 8 ☐ B 7 ☐ C 0 ☐ D 3

2. Prikazivanje znanja i automatsko zaključivanje (4 pitanja)

- 5** (R) Želimo dokazati sljedeću relaciju deduktivne posljedice:

$$\forall x \left(P(x, a) \rightarrow \neg \forall y (Q(y) \rightarrow \exists z R(z, y)) \right), \forall x \exists y (Q(y) \rightarrow R(y, x)) \vdash \exists x \neg P(c, x).$$

Pretvorite premise i negaciju ciljne formule u klauzalni oblik te primijenite rezoluciju opovrgavanjem uz strategiju skupa potpore (SoS). **Koliko klauzula ulazi u postupak te je li cilj dokaziv i u koliko rezolucijskih koraka?**

- ☐ A 4 klauzule; nije dokazivo ☐ C 4 klauzule; dokazivo u 3 koraka
☐ B 3 klauzule; dokazivo u 3 koraka ☐ D 3 klauzule; dokazivo u 2 koraka

6 (T) Razmotrite logičku formulu $\forall x (Paran(x) \rightarrow Neparan(add(x, 1)))$. Što je semantičko svojstvo ove formule, i zašto?

- ☐ A Formula je zadovoljiva, jer za nju postoji model
☐ B Formula je istinita, jer je sljedbenik parnog broja neparan broj
☐ C Formula je neistinita, jer x ne mora nužno biti broj
☐ D Formula je valjana, jer je sljedbenik parnog broja neparan broj

7 (T) Rezolucijsko zaključivanje u FOL oslanja se na algoritam najopćenitijeg zajedničkog unifikatora (MGU). U kojem se trenutku primjenjuje algoritam MGU?

- ☐ A Prije svake primjene rezolucijskog pravila
☐ B Prije uvođenja negiranog cilja u skup klauzula
☐ C Nakon skolemizacije, a prije standardizacije
☐ D Nakon razrješavanja klauzula, a prije faktorizacije

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

- ☐ A 74 ☐ B 60 ☐ C 104 ☐ D 119

3. Logičko programiranje i ekspertni sustavi (2 pitanja)

9 (R) Razmatramo sljedeći program u Prologu:

```
roditelj(kronos, zeus).  
roditelj(zeus, minos).  
roditelj(europa, minos).  
potomak(X, Y) :- roditelj(Y, X).  
potomak(X, Y) :- roditelj(Y, Z), potomak(X, Z).  
bog(kronos).  
bog(X) :- potomak(X, Y), bog(Y).  
polubog(X) :- potomak(X, Y), not(bog(Y)).
```

Nad ovako definiranom bazom izvodimo upit `polubog(minos)`. Nacrtajte Prologovo stablo dokaza za ovaj upit do trenutka drugog povratka u pretraživanju. Pritom za dokazivanje negiranog cilja koristite dva čvora (jedan za `not(P(X))` i drugi za `P(X)`), dok prazni stog ne brojite kao čvor. **Koliko čvorova ima stablo dokaza neposredno prije drugog povratka u pretraživanju?**

- ☐ A 8 ☐ B 12 ☐ C 10 ☐ D 6

10 (T) Kao i ekspertni sustavi, Prolog također zaključuje na temelju pravila. Međutim, u tom pogledu između ekspertnih sustava i Prologa postoji razlika. **Po čemu se ekspertni sustavi i Prolog razlikuju?**

- ☐ A Neki ekspertni sustavi mogu koristiti negaciju uvjeta u pravilima, dok Prolog ne koristi negaciju
☐ B Neki ekspertni sustavi zaključuju ulančavanjem unaprijed, dok Prolog uvijek koristi ulančavanje unazad
☐ C Prolog koristi prioritete pravila, dok ekspertni sustavi pravila aktiviraju redoslijedom kojim su navedena
☐ D Prolog u pravilima može koristiti varijable, dok ekspertni sustavi varijable ne koriste

4. Modeliranje neizvjesnosti (2 pitanja)

- 11** (T) Generalizirani modus ponens (GMP) koristi se za zaključivanje u neizrazitoj logici. Kao i modus ponens, GMP se sastoji od premise, antedecenta, konzekvensa, implikacije između antedecenta i konzekvensa, te zaključka. **Na koji su način modelirane komponente GMP-a?**

- ☐ A Premisa je apriorna vjerojatnost, a implikacija i zaključak su uvjetne vjerojatnosti
☐ B Premisa je neizraziti skup, zaključak je neizrazita relacija, a implikacija je max-min kompozicija
☐ C Premisa, antedecent, konzekvens i zaključak su neizraziti skupovi, a implikacija je neizrazita relacija
☐ D Implikacija je neizraziti skup, a konzekvens i zaključak su max-min kompozicija

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

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

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 sažeo je u listu "*Programski jezik koji mi se sviđa*", gdje je svaki jezik opisao četirima značajkama, te je naznačio je li mu se dotični jezik svidio ($y = 1$) ili nije ($y = 0$). Ta lista izgleda ovako:

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

Ovog ljeta Mali Ivica želi puno jesti i spavati te opet naučiti novi programski jezik. U užem je izboru jezik \mathbf{x} sa sljedećim karakteristikama: $\mathbf{x} = (\text{lijena}, \text{interpreter}, \text{hibridna}, \text{dinamička})$. Međutim, ovog puta Mali bi Ivica volio unaprijed znati hoće li mu se dotični programski jezik svidjeti, tako da ne gubi cijelo ljeto bezveze. Pomozite Malom Ivici te na gornji skup primjera primjenite naivan Bayesov klasifikator s Laplaceovim zaglađivanjem "dodaj jedan". **Koliko iznosi vjerojatnost da bi se Malom Ivici programski jezik \mathbf{x} svidio?**

- ☐ A 0.856 ☐ B 0.431 ☐ C 0.799 ☐ D 0.694

- 14** (R) Umjetnu neuronsku mrežu arhitekture $1 \times 2 \times 2$ učimo algoritmom propagacije pogreške unatrag. U nekom koraku algoritam ažurira težine na temelju primjera $(1, (1, 0))$. Vrijednosti svih težina u tom koraku su:

$$w_{01}^{(1)} = 0.1, w_{11}^{(1)} = 0.2, w_{02}^{(1)} = 0.5, w_{12}^{(1)} = 0.6, w_{01}^{(2)} = 0.1, w_{11}^{(2)} = 0.2, w_{21}^{(2)} = 0.3, w_{02}^{(2)} = 0.2, w_{12}^{(2)} = 0.3, w_{22}^{(2)} = 0.4$$

Svi neuroni koriste sigmoidnu prijenosnu funkciju, a stopa učenja je 10. **Koliko iznosi korekcija koja će biti pridodana težini $w_{12}^{(1)}$?**

- ☐ A +0.035152 ☐ B -0.147551 ☐ C -0.058547 ☐ D +0.065732

- 15** (T) Umjetni neuron TLU (kakav koristi perceptron) za svaku ulaznu značajku x_i ima jednu težinu w_i , ali ima i jednu dodatnu težinu, w_0 . **Koja je uloga težine w_0 ?**

- ☐ A Ako je umnožak značajki i pripadnih težina veći od $-w_0$, neuron na izlazu daje +1
☐ B Ako je zbroj svih težina veći od w_0 , neuron na izlazu daje pozitivnu vrijednost
☐ C Ako je težina w_0 negativna, a sve ostale težine su različite od nule, izlaz neurona je -1
☐ D Ako je težina w_0 pozitivna, korekcija Δw_0 je negativna, inače je pozitivna

- 16** (P) Radimo svoju implementaciju algoritma ID3. Kako bismo spriječili da se model prenauči, implementirali smo i podrezivanje stabla na dubini k . Nažalost, kod implementacije funkcije informacijske dobiti (IG) potkrala nam se mala pogreška: zaboravili smo negirati vrijednost pri izračunu entropije $E(D)$ skupa primjera D . Dakle, umjesto da izračunava $E(D)$, naša implementacija izračunava $-E(D)$. Neka je M_1^k stablo odluke koje dobivamo učenjem takvim pogrešno implementiranim algoritmom ID3, podrezano na neku konačnu dubinu k . Neka je M_2^k stablo koje bismo dobili da smo algoritam ID3 implementirali ispravno i naučeno stablo podrezali na neku konačnu dubinu k . Ako $k = \infty$, onda to znači da stablo ne podrezujemo. Neka je $E_u(M)$ pogreška učenja modela M , a $E_p(M)$ pogreška modela M na skupu za provjeru. **Što od sljedećeg možemo očekivati da vrijedi?**

☐ A $E_p(M_2^\infty) = E_u(M_2^\infty)$ ☐ B $E_p(M_1^\infty) < E_u(M_1^\infty)$ ☐ C $E_p(M_1^k) > E_p(M_2^k)$ ☐ D $E_p(M_1^k) = E_u(M_1^k)$

- 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 3 ☐ B 2.75 ☐ C 1.5 ☐ D 1.25

6. Prirodom inspirirani optimizacijski algoritmi (3 pitanja)

- 18** (P) Ako kod algoritma Ant System postavimo τ_0 na vrijednost koja je puno manja od količine feromonskih tragova koju deponira jedan mrav, **što će biti posljedica?**

☐ A Algoritam će dugo vremena istraživati nasumične puteve, prije no što se mravi fokusiraju na neke puteve
☐ B Algoritam će izgubiti mogućnost istraživanja kvalitetnih rješenja
☐ C Algoritam će rapidno konvergirati prema globalnom optimumu
☐ D Isparavanje feromonskih tragova će biti preveliko

- 19** (T) Različite instance problema trgovačkog putnika rješavamo genetičkim algoritmom (GA) i mravljim algoritmom (ACO). Na ove je optimizacijske algoritme primjenjiv "teorem o nebesplatnome ručku" (engl. *No free lunch theorem*). **Što to konkretno znači?**

☐ A Ako postoji globalno optimalno rješenje, niti GA niti ACO ga ne pronalaze
☐ B Ovisno o instanci problema, jedan od algoritama češće će nalaziti bolje rješenje
☐ C Vremenska složenost oba algoritma u prosjeku je eksponencijalna
☐ D U prosjeku jedan od ova dva algoritma češće zapinje u lokalnom optimumu

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

☐ A 16 ☐ B 18 ☐ C 9 ☐ D 5

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) Razmatramo igru dvaju minimax-igrača. Stablo igre definirano je prijelazima $\text{succ}(A) = \{B, C, D\}$, $\text{succ}(B) = \{E, F\}$, $\text{succ}(C) = \{G, H\}$, $\text{succ}(D) = \{I, J\}$, $\text{succ}(F) = \text{succ}(G) = \{K, L\}$, $\text{succ}(E) = \text{succ}(H) = \text{succ}(I) = \{M, N\}$ te $\text{succ}(J) = \{O, P\}$. Igrač MAX koristi heuristiku h_1 , a igrač MIN heuristiku h_2 . Vrijednosti heuristika su sljedeće (obje heuristike definiraju dobit dotičnog igrača):

	E	F	G	H	I	J	K	L	M	N	O	P
h_1	1	-2	3	2	3	5	-1	0	-2	-1	0	2
h_2	0	5	2	4	-1	2	3	2	-3	-2	1	-4

Oba igrača pretražuju najviše dva poteza unaprijed (jedan svoj i jedan suparnički). Igru započinje igrač MAX u stanju A. **Kroz koja stanja će se razvijati igra?**

- ☐ A A, C, G, ... ☐ B A, C, H, ... ☐ C A, D, J, ... ☐ D A, D, I, ...

- 2** (R) Na skupu stanja $S = \{a, b, c, d, e, f\}$ definirana je funkcija sljedbenika $\text{succ}(a) = \{(b, 5), (c, 2)\}$, $\text{succ}(b) = \{(e, 5)\}$, $\text{succ}(c) = \{(a, 2), (b, 1), (d, 3), (f, 11)\}$, $\text{succ}(d) = \{(e, 4)\}$, $\text{succ}(e) = \{(f, 2)\}$, $\text{succ}(f) = \emptyset$. Početno stanje je a, a ciljno f. Heurističke vrijednosti stanja neka su $h(a) = 9, h(b) = 6, h(c) = 5, h(d) = 6, h(e) = 2, h(f) = 0$. **Koliko koraka treba algoritmu A^* da pronade optimalan put u odnosu na broj koraka koji algoritmu pretraživanja u širinu treba da pronade bilo kakav put?** Napomene: 1 korak = 1 ispitivanje čvora (uzmite u obzir da broj proširivanja čvorova ne mora odgovarati ovako definiranom broju koraka). Pretpostavite leksikografski poredak između čvorova.

- ☐ A Oba algoritma trebaju isti broj koraka ☐ C Algoritmu A^* treba 5 koraka manje
☐ B Algoritmu A^* treba 3 koraka manje ☐ D Algoritmu A^* treba 2 koraka manje

- 3** (P) Stanje s slagalice 3×3 neka je $[[1, 5, 2], [4, \square, 3], [7, 8, 6]]$, dok je ciljno stanje $[[1, 2, 3], [4, 5, 6], [7, 8, \square]]$. Neka je h optimistična heuristika. **Među ponuđenim vrijednostima, koju vrijednost $h(s)$ može poprimiti, a da je najviše obaviještena?**

- ☐ A 1 ☐ B 3 ☐ C 0 ☐ D 2

- 4** (T) Algoritam A^* oslanja se na heurističku funkciju. Svojstva heurističke funkcije izravno utječu na svojstva tog algoritma. **Što je od sljedećeg točno?**

- ☐ A Ako heuristika nije optimistična, algoritam A^* nije potpun
☐ B Ako je heuristika optimistična, algoritam A^* ima linearnu prostornu složenost
☐ C Ako je heuristika konzistentna, algoritam A^* ne otvara jednom zatvorene čvorove
☐ D Ako heuristika nije konzistentna, algoritam A^* ne nalazi optimalno rješenje

2. Prikazivanje znanja i automatsko zaključivanje (4 pitanja)

- 5** (T) Razmotrite logičku formulu $\forall x (Paran(x) \rightarrow Neparan(add(x, 1)))$. **Što je semantičko svojstvo ove formule, i zašto?**

- ☐ A Formula je valjana, jer je sljedbenik parnog broja neparan broj
☐ B Formula je zadovoljiva, jer za nju postoji model
☐ C Formula je neistinita, jer x ne mora nužno biti broj
☐ D Formula je istinita, jer je sljedbenik parnog broja neparan broj

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

☐ A 74 ☐ B 119 ☐ C 60 ☐ D 104

- 7 (R) Želimo dokazati sljedeću relaciju deduktivne posljedice:

$$\forall x \left(P(x, a) \rightarrow \neg \forall y (Q(y) \rightarrow \exists z R(z, y)) \right), \forall x \exists y \left(Q(y) \rightarrow R(y, x) \right) \vdash \exists x \neg P(c, x).$$

Pretvorite premise i negaciju ciljne formule u klauzalni oblik te primijenite rezoluciju opovrgavanjem uz strategiju skupa potpore (SoS). **Koliko klauzula ulazi u postupak te je li cilj dokaziv i u koliko rezolucijskih koraka?**

☐ A 4 klauzule; nije dokazivo ☐ C 4 klauzule; dokazivo u 4 koraka
☐ B 5 klauzula; nije dokazivo ☐ D 4 klauzule; dokazivo u 3 koraka

- 8 (T) Rezolucijsko zaključivanje u FOL oslanja se na algoritam najopćenitijeg zajedničkog unifikatora (MGU). **U kojem se trenutku primjenjuje algoritam MGU?**

☐ A Nakon skolemizacije, a prije standardizacije
☐ B Nakon razrješavanja klauzula, a prije faktorizacije
☐ C Prije svake primjene rezolucijskog pravila
☐ D Prije uvođenja negiranog cilja u skup klauzula

3. Logičko programiranje i ekspertni sustavi (2 pitanja)

- 9 (T) Kao i ekspertni sustavi, Prolog također zaključuje na temelju pravila. Međutim, u tom pogledu između ekspertnih sustava i Prologa postoji razlika. **Po čemu se ekspertni sustavi i Prolog razlikuju?**

☐ A Prolog u pravilima može koristiti varijable, dok ekspertni sustavi varijable ne koriste
☐ B Neki ekspertni sustavi zaključuju ulančavanjem unaprijed, dok Prolog uvijek koristi ulančavanje unazad
☐ C Prolog koristi prioritete pravila, dok ekspertni sustavi pravila aktiviraju redoslijedom kojim su navedena
☐ D Neki ekspertni sustavi mogu koristiti negaciju uvjeta u pravilima, dok Prolog ne koristi negaciju

- 10 (R) Razmatramo sljedeći program u Prologu:

```
roditelj(kronos, zeus).
roditelj(zeus, minos).
roditelj(europa, minos).
potomak(X, Y) :- roditelj(Y, X).
potomak(X, Y) :- roditelj(Y, Z), potomak(X, Z).
bog(kronos).
bog(X) :- potomak(X, Y), bog(Y).
polubog(X) :- potomak(X, Y), not(bog(Y)).
```

Nad ovako definiranom bazom izvodimo upit `polubog(minos)`. Nacrtajte Prologovo stablo dokaza za ovaj upit do trenutka drugog povratka u pretraživanju. Pritom za dokazivanje negiranog cilja koristite dva čvora (jedan za `not(P(X))` i drugi za `P(X)`), dok prazni stog ne brojite kao čvor. **Koliko čvorova ima stablo dokaza neposredno prije drugog povratka u pretraživanju?**

☐ A 10 ☐ B 6 ☐ C 12 ☐ D 8

4. Modeliranje neizvjesnosti (2 pitanja)

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

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

- 12** (T) Generalizirani modus ponens (GMP) koristi se za zaključivanje u neizrazitoj logici. Kao i modus ponens, GMP se sastoji od premise, antecedenta, konzekvensa, implikacije između antecedenta i konzekvensa, te zaključka. **Na koji su način modelirane komponente GMP-a?**

- ☐ A Premisa je neizraziti skup, zaključak je neizrazita relacija, a implikacija je max-min kompozicija
☐ B Implikacija je neizraziti skup, a konzekvens i zaključak su max-min kompozicija
☐ C Premisa, antecedent, konzekvens i zaključak su neizraziti skupovi, a implikacija je neizrazita relacija
☐ D Premisa je apriorna vjerojatnost, a implikacija i zaključak su uvjetne vjerojatnosti

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

- 13** (R) Umjetnu neuronsku mrežu arhitekture $1 \times 2 \times 2$ učimo algoritmom propagacije pogreške unatrag. U nekom koraku algoritam ažurira težine na temelju primjera $(1, (1, 0))$. Vrijednosti svih težina u tom koraku su:

$$w_{01}^{(1)} = 0.1, w_{11}^{(1)} = 0.2, w_{02}^{(1)} = 0.5, w_{12}^{(1)} = 0.6, w_{01}^{(2)} = 0.1, w_{11}^{(2)} = 0.2, w_{21}^{(2)} = 0.3, w_{02}^{(2)} = 0.2, w_{12}^{(2)} = 0.3, w_{22}^{(2)} = 0.4$$

Svi neuroni koriste sigmoidnu prijenosnu funkciju, a stopa učenja je 10. **Koliko iznosi korekcija koja će biti pridodana težini $w_{12}^{(1)}$?**

- ☐ A -0.147551 ☐ B -0.058547 ☐ C $+0.065732$ ☐ D $+0.035152$

- 14** (T) Umjetni neuron TLU (kakav koristi perceptron) za svaku ulaznu značajku x_i ima jednu težinu w_i , ali ima i jednu dodatnu težinu, w_0 . **Koja je uloga težine w_0 ?**

- ☐ A Ako je zbroj svih težina veći od w_0 , neuron na izlazu daje pozitivnu vrijednost
☐ B Ako je težina w_0 negativna, a sve ostale težine su različite od nule, izlaz neurona je -1
☐ C Ako je umnožak značajki i pripadnih težina veći od $-w_0$, neuron na izlazu daje $+1$
☐ D Ako je težina w_0 pozitivna, korekcija Δw_0 je negativna, inače je pozitivna

- 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	Evaluacija	Izvođenje	Paradigma	Provjera tipova	y
1	lijena	kompajler	imperativna	statička	0
2	striktna	interpreter	deklarativna	dinamička	0
3	lijena	kompajler	imperativna	dinamička	0
4	lijena	interpreter	hibridna	statička	0
5	striktna	interpreter	imperativna	statička	1
6	lijena	kompajler	hibridna	dinamička	1
7	striktna	kompajler	hibridna	dinamička	1

Ovog ljeta Mali Ivica želi puno jesti i spavati te opet naučiti novi programski jezik. U užem izboru jezik x sa sljedećim karakteristikama: $x = (\text{lijena}, \text{interpreter}, \text{hibridna}, \text{dinamička})$. Međutim, ovog puta Mali bi Ivica volio unaprijed znati hoće li mu se dotični programski jezik svidjeti, tako da ne gubi cijelo ljeto bezveze. Pomozite Malom Ivici te na gornji skup primjera primijenite naivan Bayesov klasifikator s Laplaceovim zaglađivanjem "dodaj jedan". **Koliko iznosi vjerojatnost da bi se Malom Ivici programski jezik x svidio?**

- ☐ A 0.856 ☐ B 0.431 ☐ C 0.799 ☐ D 0.694

- 16** (P) Radimo svoju implementaciju algoritma ID3. Kako bismo spriječili da se model prenauči, implementirali smo i podrezivanje stabla na dubini k . Nažalost, kod implementacije funkcije informacijske dobiti (IG) potkrala nam se mala pogreška: zaboravili smo negirati vrijednost pri izračunu entropije $E(D)$ skupa primjera D . Dakle, umjesto da izračunava $E(D)$, naša implementacija izračunava $-E(D)$. Neka je M_1^k stablo odluke koje dobivamo učenjem takvim pogrešno implementiranim algoritmom ID3, podrezano na neku konačnu dubinu k . Neka je M_2^k stablo koje bismo dobili da smo algoritam ID3 implementirali ispravno i naučeno stablo podrezali na neku konačnu dubinu k . Ako $k = \infty$, onda to znači da stablo ne podrezujemo. Neka je $E_u(M)$ pogreška učenja modela M , a $E_p(M)$ pogreška modela M na skupu za provjeru. **Što od sljedećeg možemo očekivati da vrijedi?**

☐ A $E_p(M_1^\infty) > E_p(M_2^\infty)$ ☐ B $E_u(M_2^k) < E_u(M_2^\infty)$ ☐ C $E_u(M_1^k) = E_u(M_1^\infty)$ ☐ D $E_p(M_2^\infty) < E_u(M_2^\infty)$

- 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 1.25 ☐ B 3 ☐ C 2.75 ☐ D 1.5

6. Prirodom inspirirani optimizacijski algoritmi (3 pitanja)

- 18** (P) Ako kod algoritma Ant System postavimo τ_0 na vrijednost koja je puno veća od količine feromonskih tragova koju deponira jedan mrav, **što će biti posljedica?**

☐ A Algoritam će dugo vremena istraživati nasumične puteve, prije no što se mravi fokusiraju na neke puteve
☐ B Algoritam će izgubiti mogućnost istraživanja kvalitetnih rješenja
☐ C Algoritam će rapidno konvergirati prema globalnom optimumu
☐ D Isparavanje feromonskih tragova će biti preveliko

- 19** (T) Različite instance problema trgovačkog putnika rješavamo genetičkim algoritmom (GA) i mravljim algoritmom (ACO). Na ove je optimizacijske algoritme primjenjiv “teorem o nebesplatnome ručku” (engl. *No free lunch theorem*). **Što to konkretno znači?**

☐ A U prosjeku jedan od ova dva algoritma češće zapinje u lokalnom optimumu
☐ B Ako postoji globalno optimalno rješenje, niti GA niti ACO ga ne pronalaze
☐ C Vremenska složenost oba algoritma u prosjeku je eksponencijalna
☐ D Ovisno o instanci problema, jedan od algoritama češće će nalaziti bolje rješenje

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

☐ A 9 ☐ B 5 ☐ C 18 ☐ D 16

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) Razmatramo igru dvaju minimax-igrača. Stablo igre definirano je prijelazima $\text{succ}(A) = \{B, C, D\}$, $\text{succ}(B) = \{E, F\}$, $\text{succ}(C) = \{G, H\}$, $\text{succ}(D) = \{I, J\}$, $\text{succ}(F) = \text{succ}(G) = \{K, L\}$, $\text{succ}(E) = \text{succ}(H) = \text{succ}(I) = \{M, N\}$ te $\text{succ}(J) = \{O, P\}$. Igrač MAX koristi heuristiku h_1 , a igrač MIN heuristiku h_2 . Vrijednosti heuristika su sljedeće (obje heuristike definiraju dobit dotičnog igrača):

	E	F	G	H	I	J	K	L	M	N	O	P
h_1	1	-2	3	2	3	5	-1	0	-2	-1	0	2
h_2	0	5	2	4	-1	2	3	2	-3	-2	1	-4

Oba igrača pretražuju najviše dva poteza unaprijed (jedan svoj i jedan suparnički). Igru započinje igrač MAX u stanju A. **Kroz koja stanja će se razvijati igra?**

- ☐ A A, C, H, ... ☐ B A, C, G, ... ☐ C A, D, I, ... ☐ D A, D, J, ...

- 2** (T) Algoritam A^* oslanja se na heurističku funkciju. Svojstva heurističke funkcije izravno utječu na svojstva tog algoritma. **Što je od sljedećeg točno?**

- ☐ A Ako je heuristika konzistentna, algoritam A^* ne otvara jednom zatvorene čvorove
☐ B Ako heuristika nije optimistična, algoritam A^* nije potpun
☐ C Ako je heuristika optimistična, algoritam A^* ima linearnu prostornu složenost
☐ D Ako heuristika nije konzistentna, algoritam A^* ne nalazi optimalno rješenje

- 3** (R) Na skupu stanja $S = \{a, b, c, d, e, f\}$ definirana je funkcija sljedbenika $\text{succ}(a) = \{(b, 5), (c, 2)\}$, $\text{succ}(b) = \{(e, 5)\}$, $\text{succ}(c) = \{(a, 2), (b, 1), (d, 3), (f, 11)\}$, $\text{succ}(d) = \{(e, 4)\}$, $\text{succ}(e) = \{(f, 2)\}$, $\text{succ}(f) = \emptyset$. Početno stanje je a, a ciljno f. Heurističke vrijednosti stanja neka su $h(a) = 9$, $h(b) = 6$, $h(c) = 5$, $h(d) = 6$, $h(e) = 2$, $h(f) = 0$. **Koliko koraka treba algoritmu A^* da pronade optimalan put u odnosu na broj koraka koji algoritmu "najbolji prvi" treba da pronade bilo kakav put?** *Napomene:* 1 korak = 1 ispitivanje čvora (uzmite u obzir da broj proširivanja čvorova ne mora odgovarati ovako definiranom broju koraka). Pretpostavite leksikografski poredak između čvorova.

- ☐ A Algoritmu A^* treba 2 koraka više ☐ C Algoritmu A^* treba 1 korak više
☐ B Algoritmu A^* treba 4 koraka više ☐ D Algoritmu A^* treba 1 korak manje

- 4** (P) Stanje s slagalice 3×3 neka je $[[1, \square, 2], [4, 5, 3], [7, 8, 6]]$, dok je ciljno stanje $[[1, 2, 3], [4, 5, 6], [7, 8, \square]]$. Neka je h optimistična heuristika. **Među ponuđenim vrijednostima, koju vrijednost $h(s)$ može poprimiti, a da je najviše obaviještena?**

- ☐ A 4 ☐ B 8 ☐ C 1 ☐ D 2

2. Prikazivanje znanja i automatsko zaključivanje (4 pitanja)

- 5** (T) Razmotrite logičku formulu $\forall x (Paran(x) \rightarrow Neparan(add(x, 1)))$. **Što je semantičko svojstvo ove formule, i zašto?**

- ☐ A Formula je valjana, jer je sljedbenik parnog broja neparan broj
☐ B Formula je neistinita, jer x ne mora nužno biti broj
☐ C Formula je zadovoljljiva, jer za nju postoji model
☐ D Formula je istinita, jer je sljedbenik parnog broja neparan broj

- 6 (R) Želimo dokazati sljedeću relaciju deduktivne posljedice:

$$\forall x \left(P(x, a) \rightarrow \neg \forall y (Q(y) \rightarrow \exists z R(z, y)) \right), \forall x \exists y (Q(y) \rightarrow R(y, x)) \vdash \exists x \neg (P(c, x) \wedge R(x, c)).$$

Pretvorite premise i negaciju ciljne formule u klauzalni oblik te primijenite rezoluciju opovrgavanjem uz strategiju skupa potpore (SoS). **Koliko klauzula ulazi u postupak te je li cilj dokaziv i u koliko rezolucijskih koraka?**

- ☐ A 4 klauzule; dokazivo u 2 koraka ☐ C 5 klauzula; nije dokazivo
☐ B 4 klauzule; dokazivo u 4 koraka ☐ D 5 klauzula; dokazivo u 4 koraka

- 7 (T) Rezolucijsko zaključivanje u FOL oslanja se na algoritam najopćenitijeg zajedničkog unifikatora (MGU). U kojem se trenutku primjenjuje algoritam MGU?

- ☐ A Nakon razrješavanja klauzula, a prije faktorizacije ☐ C Nakon skolemizacije, a prije standardizacije
☐ B Prije uvođenja negiranog cilja u skup klauzula ☐ D Prije svake primjene rezolucijskog pravila

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

- ☐ A 104 ☐ B 60 ☐ C 74 ☐ D 119

3. Logičko programiranje i ekspertni sustavi (2 pitanja)

- 9 (R) Razmatramo sljedeći program u Prologu:

```
roditelj(kronos, zeus).  
roditelj(zeus, minos).  
roditelj(europa, minos).  
potomak(X, Y) :- roditelj(Y, X).  
potomak(X, Y) :- roditelj(Y, Z), potomak(X, Z).  
bog(kronos).  
bog(X) :- potomak(X, Y), bog(Y).  
polubog(X) :- potomak(X, Y), not(bog(Y)).
```

Nad ovako definiranom bazom izvodimo upit `polubog(minos)`. Nacrtajte Prologovo stablo dokaza za ovaj upit do trenutka drugog povratka u pretraživanju. Pritom za dokazivanje negiranog cilja koristite dva čvora (jedan za `not(P(X))` i drugi za `P(X)`), dok prazni stog ne brojite kao čvor. **Koliko čvorova ima stablo dokaza neposredno prije drugog povratka u pretraživanju?**

- ☐ A 8 ☐ B 12 ☐ C 6 ☐ D 10

- 10 (T) Kao i ekspertni sustavi, Prolog također zaključuje na temelju pravila. Međutim, u tom pogledu između ekspertnih sustava i Prologa postoji razlika. **Po čemu se ekspertni sustavi i Prolog razlikuju?**

- ☐ A Neki ekspertni sustavi zaključuju ulančavanjem unaprijed, dok Prolog uvijek koristi ulančavanje unazad
☐ B Prolog u pravilima može koristiti varijable, dok ekspertni sustavi varijable ne koriste
☐ C Neki ekspertni sustavi mogu koristiti negaciju uvjeta u pravilima, dok Prolog ne koristi negaciju
☐ D Prolog koristi prioritete pravila, dok ekspertni sustavi pravila aktiviraju redoslijedom kojim su navedena

4. Modeliranje neizvjesnosti (2 pitanja)

- 11 (P) Razmotrimo jezičnu varijablu *starost kanarinca*, definiranu s izrazima *mlad* i *star*. Značenje tih izraza modeliramo neirazitim skupovima M odnosno S , definiranim nad univerzalnim skupom \mathbb{R}^+ , koji odgovara godinama. Funkcije pripadnosti μ_M i μ_S definiramo kao po dijelovima linearne funkcije. Funkcija $\mu_M(x)$ ima vrijednost 1 za

$0 \leq x \leq 5$, vrijednost 0 za $x \geq 9$, te linearno pada za $5 < x < 9$. Funkcija $\mu_S(x)$ ima vrijednost 0 za $0 \leq x \leq 6$, vrijednost 1 za $x \geq 12$, te linearno raste za $6 < x < 12$. Uporabom Zadehových operatora definiramo dva neizrazita skupa: skup X_1 sa značenjem *mlad ili star kanarinac* te skup X_2 sa značenjem *mlad ili ne mlad kanarinac*. **Za koje se sve elemente $x \in \mathbb{R}^+$ pripadnost ovim dvama neizrazitim skupovima razlikuje?**

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

- 12** (T) Generalizirani modus ponens (GMP) koristi se za zaključivanje u neizrazitoj logici. Kao i modus ponens, GMP se sastoji od premise, antedecenta, konzekvensa, implikacije između antedecenta i konzekvensa, te zaključka. **Na koji su način modelirane komponente GMP-a?**

- ☐ A Premisa, antedecent, konzekvens i zaključak su neizraziti skupovi, a implikacija je neizrazita relacija
☐ B Premisa je neizraziti skup, zaključak je neizrazita relacija, a implikacija je max-min kompozicija
☐ C Implikacija je neizraziti skup, a konzekvens i zaključak su max-min kompozicija
☐ D Premisa je apriorna vjerojatnost, a implikacija i zaključak su uvjetne vjerojatnosti

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

- 13** (P) Radimo svoju implementaciju algoritma ID3. Kako bismo spriječili da se model prenauči, implementirali smo i podrezivanje stabla na dubini k . Nažalost, kod implementacije funkcije informacijske dobiti (IG) potkrala nam se mala pogreška: zaboravili smo negirati vrijednost pri izračunu entropije $E(D)$ skupa primjera D . Dakle, umjesto da izračunava $E(D)$, naša implementacija izračunava $-E(D)$. Neka je M_1^k stablo odluke koje dobivamo učenjem takvim pogrešno implementiranim algoritmom ID3, podrezano na neku konačnu dubinu k . Neka je M_2^k stablo koje bismo dobili da smo algoritam ID3 implementirali ispravno i naučeno stablo podrezali na neku konačnu dubinu k . Ako $k = \infty$, onda to znači da stablo ne podrezujemo. Neka je $E_u(M)$ pogreška učenja modela M , a $E_p(M)$ pogreška modela M na skupu za provjeru. **Što od sljedećeg možemo očekivati da vrijedi?**

- ☐ A $E_p(M_1^\infty) > E_p(M_2^\infty)$ ☐ B $E_p(M_1^k) = E_u(M_1^k)$ ☐ C $E_p(M_2^k) < E_u(M_2^k)$ ☐ D $E_p(M_1^\infty) = E_u(M_1^\infty)$

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

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

Ovog ljeta Mali Ivica želi puno jesti i spavati te opet naučiti novi programski jezik. U užem je izboru jezik x sa sljedećim karakteristikama: $x = (\text{lijena}, \text{interpreter}, \text{hibridna}, \text{dinamička})$. Međutim, ovog puta Mali bi Ivica volio unaprijed znati hoće li mu se dotični programski jezik svidjeti, tako da ne gubi cijelo ljeto bezveze. Pomozite Malom Ivici te na gornji skup primjera primjenite naivan Bayesov klasifikator s Laplaceovim zaglađivanjem “dodaj jedan”. **Koliko iznosi vjerojatnost da bi se Malom Ivici programski jezik x svidio?**

- ☐ A 0.799 ☐ B 0.694 ☐ C 0.431 ☐ D 0.856

- 15** (R) Umjetnu neuronsku mrežu arhitekture $1 \times 2 \times 2$ učimo algoritmom propagacije pogreške unatrag. U nekom koraku algoritam ažurira težine na temelju primjera $(1, (1, 0))$. Vrijednosti svih težina u tom koraku su:

$$w_{01}^{(1)} = 0.1, w_{11}^{(1)} = 0.2, w_{02}^{(1)} = 0.5, w_{12}^{(1)} = 0.6, w_{01}^{(2)} = 0.1, w_{11}^{(2)} = 0.2, w_{21}^{(2)} = 0.3, w_{02}^{(2)} = 0.2, w_{12}^{(2)} = 0.3, w_{22}^{(2)} = 0.4$$

Svi neuroni koriste sigmoidnu prijenosnu funkciju, a stopa učenja je 10. **Koliko iznosi korekcija koja će biti pridodana težini $w_{12}^{(1)}$?**

- ☐ A -0.058547 ☐ B -0.147551 ☐ C $+0.035152$ ☐ D $+0.065732$

- 16** (T) Umjetni neuron TLU (kakav koristi perceptron) za svaku ulaznu značajku x_i ima jednu težinu w_i , ali ima i jednu dodatnu težinu, w_0 . **Koja je uloga težine w_0 ?**
- ☐ A Ako je težina w_0 negativna, a sve ostale težine su različite od nule, izlaz neurona je -1
- ☐ B Ako je zbroj svih težina veći od w_0 , neuron na izlazu daje pozitivnu vrijednost
- ☐ C Ako je težina w_0 pozitivna, korekcija Δw_0 je negativna, inače je pozitivna
- ☐ D Ako je umnožak značajki i pripadnih težina veći od $-w_0$, neuron na izlazu daje $+1$
- 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 1.25 ☐ B 2.75 ☐ C 1.5 ☐ D 3

6. Prirodom inspirirani optimizacijski algoritmi (3 pitanja)

- 18** (R) Generacijskim genetskim algoritmom tražimo maksimum funkcije $f(x, y) = 13 - (x - 2)^2 - (y - 3)^2$. Kao reprezentaciju rješenja koristimo 6-bitovni kromosom, pri čemu se prva tri bita koriste za kodiranje vrijednosti varijable x , a preostala tri bita za kodiranje vrijednosti varijable y . Domena nad kojom se pretražuju vrijednosti od x je $[0, 7]$ te od y je $[-1, 6]$. Populacija se sastoji od četiri jedinke: J1=000110, J2=011001, J3=100111, J4=001110. Pretpostavite da se roditelji biraju proporcionalnom selekcijom te da su u jednom koraku kao roditelji izvučena dva rješenja koja imaju najmanju vjerojatnost odabira. Provedite nad njima postupak križanja s jednom točkom prijeloma (točka prijeloma je nakon prva dva bita); pretpostavite da u ovom koraku operator mutacije svaki puta djeluje na posljednja dva bita kromosoma. **Odredite iznos funkcije f u rješenju koje odgovara djetetu koje će biti poslano u sljedeću generaciju.** (Ako ih operatori križanja i mutacije generiraju više, u novu generaciju treba prosljediti najbolje.)
- ☐ A 13 ☐ B 4 ☐ C 0 ☐ D 11
- 19** (P) Ako kod algoritma Ant System postavimo τ_0 na vrijednost koja je puno veća od količine feromonskih tragova koju deponira jedan mrav, **što će biti posljedica?**
- ☐ A Algoritam će rapidno konvergirati prema globalnom optimumu
- ☐ B Algoritam će dugo vremena istraživati nasumične puteve, prije no što se mravi fokusiraju na neke puteve
- ☐ C Algoritam će izgubiti mogućnost istraživanja kvalitetnih rješenja
- ☐ D Isparavanje feromonskih tragova će biti preveliko
- 20** (T) Različite instance problema trgovačkog putnika rješavamo genetičkim algoritmom (GA) i mravljim algoritmom (ACO). Na ove je optimizacijske algoritme primjenjiv "teorem o nebesplatnome ručku" (engl. *No free lunch theorem*). **Što to konkretno znači?**
- ☐ A U prosjeku jedan od ova dva algoritma češće zapinje u lokalnom optimumu
- ☐ B Ako postoji globalno optimalno rješenje, niti GA niti ACO ga ne pronalaze
- ☐ C Ovisno o instanci problema, jedan od algoritama češće će nalaziti bolje rješenje
- ☐ D Vremenska složenost oba algoritma u prosjeku je eksponencijalna

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) Stanje s slagalice 3×3 neka je $[[1, 2, \square], [4, 5, 3], [7, 8, 6]]$, dok je ciljno stanje $[[1, 2, 3], [4, 5, 6], [7, 8, \square]]$. Neka je h optimistična heuristika. Među ponuđenim vrijednostima, koju vrijednost $h(s)$ može poprimiti, a da je najviše obaviještena?

☐ A 3 ☐ B 8 ☐ C 5 ☐ D 1

- 2** (T) Algoritam A^* oslanja se na heurističku funkciju. Svojstva heurističke funkcije izravno utječu na svojstva tog algoritma. Što je od sljedećeg točno?

- ☐ A Ako je heuristika optimistična, algoritam A^* ima linearnu prostornu složenost
☐ B Ako heuristika nije konzistentna, algoritam A^* ne nalazi optimalno rješenje
☐ C Ako je heuristika konzistentna, algoritam A^* ne otvara jednom zatvorene čvorove
☐ D Ako heuristika nije optimistična, algoritam A^* nije potpun

- 3** (R) Na skupu stanja $S = \{a, b, c, d, e, f\}$ definirana je funkcija sljedbenika $\text{succ}(a) = \{(b, 5), (c, 2)\}$, $\text{succ}(b) = \{(e, 5)\}$, $\text{succ}(c) = \{(a, 2), (b, 1), (d, 3), (f, 11)\}$, $\text{succ}(d) = \{(e, 4)\}$, $\text{succ}(e) = \{(f, 2)\}$, $\text{succ}(f) = \emptyset$. Početno stanje je a , a ciljno f . Heurističke vrijednosti stanja neka su $h(a) = 9, h(b) = 6, h(c) = 5, h(d) = 6, h(e) = 2, h(f) = 0$. Koliko koraka treba algoritmu A^* da pronade optimalan put u odnosu na broj koraka koji algoritmu “najbolji prvi” treba da pronade bilo kakav put? Napomene: 1 korak = 1 ispitivanje čvora (uzmite u obzir da broj proširivanja čvorova ne mora odgovarati ovako definiranom broju koraka). Pretpostavite leksikografski poredak između čvorova.

- ☐ A Oba algoritma trebaju isti broj koraka ☐ C Algoritmu A^* treba 2 koraka manje
☐ B Algoritmu A^* treba 2 koraka više ☐ D Algoritmu A^* treba 4 koraka više

- 4** (R) Razmatramo igru dvaju minimax-igrača. Stablo igre definirano je prijelazima $\text{succ}(A) = \{B, C, D\}$, $\text{succ}(B) = \{E, F\}$, $\text{succ}(C) = \{G, H\}$, $\text{succ}(D) = \{I, J\}$, $\text{succ}(F) = \text{succ}(G) = \{K, L\}$, $\text{succ}(E) = \text{succ}(H) = \text{succ}(I) = \{M, N\}$ te $\text{succ}(J) = \{O, P\}$. Igrač MAX koristi heuristiku h_1 , a igrač MIN heuristiku h_2 . Vrijednosti heuristika su sljedeće (obje heuristike definiraju dobit dotičnog igrača):

	E	F	G	H	I	J	K	L	M	N	O	P
h_1	1	-2	3	2	3	5	-1	0	-2	-1	0	2
h_2	0	5	2	4	-1	2	3	2	-3	-2	1	-4

Oba igrača pretražuju najviše dva poteza unaprijed (jedan svoj i jedan suparnički). Igru započinje igrač MAX u stanju A . Kroz koja stanja će se razvijati igra?

- ☐ A A, D, I, \dots ☐ B A, C, H, \dots ☐ C A, C, G, \dots ☐ D A, D, J, \dots

2. Prikazivanje znanja i automatsko zaključivanje (4 pitanja)

- 5** (R) Želimo dokazati sljedeću relaciju deduktivne posljedice:

$$\forall x \left(P(x, a) \rightarrow \neg \forall y (Q(y) \rightarrow \exists z R(z, y)) \right), \forall x \exists y \left(Q(y) \rightarrow R(y, x) \right) \vdash \exists x \neg \left(P(x, c) \wedge R(x, c) \right).$$

Pretvorite premise i negaciju ciljne formule u klauzalni oblik te primijenite rezoluciju opovrgavanjem uz strategiju skupa potpore (SoS). **Koliko klauzula ulazi u postupak te je li cilj dokaziv i u koliko rezolucijskih koraka?**

- ☐ A 3 klauzule; dokazivo u 2 koraka ☐ C 5 klauzula; dokazivo u 2 koraka
☐ B 5 klauzula; nije dokazivo ☐ D 5 klauzula; dokazivo u 3 koraka

6 (T) Razmotrite logičku formulu $\forall x (Paran(x) \rightarrow Neparan(add(x, 1)))$. Što je semantičko svojstvo ove formule, i zašto?

- ☐ A Formula je zadovoljiva, jer za nju postoji model
☐ B Formula je neistinita, jer x ne mora nužno biti broj
☐ C Formula je istinita, jer je sljedbenik parnog broja neparan broj
☐ D Formula je valjana, jer je sljedbenik parnog broja neparan broj

7 (T) Rezolucijsko zaključivanje u FOL oslanja se na algoritam najopćenitijeg zajedničkog unifikatora (MGU). U kojem se trenutku primjenjuje algoritam MGU?

- ☐ A Prije svake primjene rezolucijskog pravila
☐ B Nakon razrješavanja klauzula, a prije faktorizacije
☐ C Nakon skolemizacije, a prije standardizacije
☐ D Prije uvođenja negiranog cilja u skup klauzula

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

- ☐ A 60 ☐ B 119 ☐ C 104 ☐ D 74

3. Logičko programiranje i ekspertni sustavi (2 pitanja)

9 (T) Kao i ekspertni sustavi, Prolog također zaključuje na temelju pravila. Međutim, u tom pogledu između ekspertnih sustava i Prologa postoji razlika. **Po čemu se ekspertni sustavi i Prolog razlikuju?**

- ☐ A Neki ekspertni sustavi zaključuju ulančavanjem unaprijed, dok Prolog uvijek koristi ulančavanje unazad
☐ B Prolog koristi prioritete pravila, dok ekspertni sustavi pravila aktiviraju redoslijedom kojim su navedena
☐ C Neki ekspertni sustavi mogu koristiti negaciju uvjeta u pravilima, dok Prolog ne koristi negaciju
☐ D Prolog u pravilima može koristiti varijable, dok ekspertni sustavi varijable ne koriste

10 (R) Razmatramo sljedeći program u Prologu:

```
roditelj(kronos, zeus).
roditelj(zeus, minos).
roditelj(europa, minos).
potomak(X, Y) :- roditelj(Y, X).
potomak(X, Y) :- roditelj(Y, Z), potomak(X, Z).
bog(kronos).
bog(X) :- potomak(X, Y), bog(Y).
polubog(X) :- potomak(X, Y), not(bog(Y)).
```

Nad ovako definiranom bazom izvodimo upit `polubog(minos)`. Nacrtajte Prologovo stablo dokaza za ovaj upit do trenutka drugog povratka u pretraživanju. Pritom za dokazivanje negiranog cilja koristite dva čvora (jedan za `not(P(X))` i drugi za `P(X)`), dok prazni stog ne brojite kao čvor. **Koliko čvorova ima stablo dokaza neposredno prije drugog povratka u pretraživanju?**

- ☐ A 6 ☐ B 10 ☐ C 12 ☐ D 8

4. Modeliranje neizvjesnosti (2 pitanja)

- 11** (T) Generalizirani modus ponens (GMP) koristi se za zaključivanje u neizrazitoj logici. Kao i modus ponens, GMP se sastoji od premise, antedecenta, konzekvensa, implikacije između antedecenta i konzekvensa, te zaključka. **Na koji su način modelirane komponente GMP-a?**

- ☐ A Implikacija je neizraziti skup, a konzekvens i zaključak su max-min kompozicija
- ☐ B Premisa je neizraziti skup, zaključak je neizrazita relacija, a implikacija je max-min kompozicija
- ☐ C Premisa, antedecent, konzekvens i zaključak su neizraziti skupovi, a implikacija je neizrazita relacija
- ☐ D Premisa je apriorna vjerojatnost, a implikacija i zaključak su uvjetne vjerojatnosti

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

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

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 sažeo je u listu "*Programski jezik koji mi se sviđa*", gdje je svaki jezik opisao četirima značajkama, te je naznačio je li mu se dotični jezik svidio ($y = 1$) ili nije ($y = 0$). Ta lista izgleda ovako:

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

Ovog ljeta Mali Ivica želi puno jesti i spavati te opet naučiti novi programski jezik. U užem je izboru jezik x sa sljedećim karakteristikama: $x = (\text{lijena}, \text{interpreter}, \text{hibridna}, \text{dinamička})$. Međutim, ovog puta Mali bi Ivica volio unaprijed znati hoće li mu se dotični programski jezik svidjeti, tako da ne gubi cijelo ljeto bezveze. Pomozite Malom Ivici te na gornji skup primjera primjenite naivan Bayesov klasifikator s Laplaceovim zaglađivanjem "dodaj jedan". **Koliko iznosi vjerojatnost da bi se Malom Ivici programski jezik x svidio?**

- ☐ A 0.431 ☐ B 0.694 ☐ C 0.856 ☐ D 0.799

- 14** (P) Radimo svoju implementaciju algoritma ID3. Kako bismo spriječili da se model prenauci, implementirali smo i podrezivanje stabla na dubini k . Nažalost, kod implementacije funkcije informacijske dobiti (IG) potkrala nam se mala pogreška: zaboravili smo negirati vrijednost pri izračunu entropije $E(D)$ skupa primjera D . Dakle, umjesto da izračunava $E(D)$, naša implementacija izračunava $-E(D)$. Neka je M_1^k stablo odluke koje dobivamo učenjem takvim pogrešno implementiranim algoritmom ID3, podrezano na neku konačnu dubinu k . Neka je M_2^k stablo koje bismo dobili da smo algoritam ID3 implementirali ispravno i naučeno stablo podrezali na neku konačnu dubinu k . Ako $k = \infty$, onda to znači da stablo ne podrezujemo. Neka je $E_u(M)$ pogreška učenja modela M , a $E_p(M)$ pogreška modela M na skupu za provjeru. **Što od sljedećeg možemo očekivati da vrijedi?**

- ☐ A $E_p(M_2^\infty) = E_u(M_2^\infty)$ ☐ B $E_p(M_1^\infty) > E_p(M_2^\infty)$ ☐ C $E_p(M_1^\infty) = E_p(M_2^\infty)$ ☐ D $E_p(M_1^\infty) = E_u(M_1^\infty)$

- 15** (R) Umjetnu neuronsku mrežu arhitekture $1 \times 2 \times 2$ učimo algoritmom propagacije pogreške unatrag. U nekom koraku algoritam ažurira težine na temelju primjera $(1, (1, 0))$. Vrijednosti svih težina u tom koraku su:

$$w_{01}^{(1)} = 0.1, w_{11}^{(1)} = 0.2, w_{02}^{(1)} = 0.5, w_{12}^{(1)} = 0.6, w_{01}^{(2)} = 0.1, w_{11}^{(2)} = 0.2, w_{21}^{(2)} = 0.3, w_{02}^{(2)} = 0.2, w_{12}^{(2)} = 0.3, w_{22}^{(2)} = 0.4$$

Svi neuroni koriste sigmoidnu prijenosnu funkciju, a stopa učenja je 10. **Koliko iznosi korekcija koja će biti pridodana težini $w_{12}^{(1)}$?**

- ☐ A -0.147551 ☐ B -0.058547 ☐ C $+0.065732$ ☐ D $+0.035152$

16 (T) Umjetni neuron TLU (kakav koristi perceptron) za svaku ulaznu značajku x_i ima jednu težinu w_i , ali ima i jednu dodatnu težinu, w_0 . **Koja je uloga težine w_0 ?**

- ☐ A Ako je težina w_0 negativna, a sve ostale težine su različite od nule, izlaz neurona je -1
☐ B Ako je težina w_0 pozitivna, korekcija Δw_0 je negativna, inače je pozitivna
☐ C Ako je umnožak značajki i pripadnih težina veći od $-w_0$, neuron na izlazu daje $+1$
☐ D Ako je zbroj svih težina veći od w_0 , neuron na izlazu daje pozitivnu vrijednost

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 1.25 ☐ B 2.75 ☐ C 1.5 ☐ D 3

6. Prirodom inspirirani optimizacijski algoritmi (3 pitanja)

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

- ☐ A 18 ☐ B 11 ☐ C 7 ☐ D 20

19 (P) Ako kod algoritma Ant System postavimo τ_0 na vrijednost koja je puno manja od količine feromonskih tragova koju deponira jedan mrav, **što će biti posljedica?**

- ☐ A Isparavanje feromonskih tragova će biti preveliko
☐ B Algoritam će izgubiti mogućnost istraživanja kvalitetnih rješenja
☐ C Algoritam će dugo vremena istraživati nasumične puteve, prije no što se mravi fokusiraju na neke puteve
☐ D Algoritam će rapidno konvergirati prema globalnom optimumu

20 (T) Različite instance problema trgovačkog putnika rješavamo genetičkim algoritmom (GA) i mravljim algoritmom (ACO). Na ove je optimizacijske algoritme primjenjiv "teorem o nebesplatnome ručku" (engl. *No free lunch theorem*). **Što to konkretno znači?**

- ☐ A Ako postoji globalno optimalno rješenje, niti GA niti ACO ga ne pronalaze
☐ B Vremenska složenost oba algoritma u prosjeku je eksponencijalna
☐ C U prosjeku jedan od ova dva algoritma češće zapinje u lokalnom optimumu
☐ D Ovisno o instanci problema, jedan od algoritama češće će nalaziti bolje rješenje

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