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

- A Algoritam ne dostiže šesti korak
- B $O = [(c, 2), (f, 20)]$, $C = \{(a, 0), (b, 2), (d, 4), (e, 6)\}$
- C $O = [(e, 2), (f, 0)]$, $C = \{(a, 16), (b, 6), (c, 14), (d, 3)\}$
- D $O = [(e, 5), (f, 20)]$, $C = \{(a, 0), (b, 2), (c, 2), (d, 3)\}$

2 (P) Alfa-beta podrezivanje može značajno smanjiti broj generiranih čvorova kod algoritma minimax. Ušteda ovisi o redoslijedu kojim se generiraju čvorovi. Optimalan redoslijed je onaj koji djecu čvora MAX generira silazno po minimax vrijednostima, a djecu čvora MIN uzlazno po minimax vrijednostima. Razmotite igru s dva moguća poteza u svakome stanju. Stablo se pretražuje do dubine 3. Vrijednosti heuristike igrača MAX za stanja u listovima stabla igre, kada su čvorovi-djeca generirani nasumičnim redoslijedom, su sljedeća (slijeva nadesno): $0, 3, -3, 4, 4, -1, 0, 2$. Na potezu je igrač MAX. Odredite broj čvorova koji će biti uklonjeni podrezivanjem alfa-beta. Zatim ponovite izračun, ali s optimalnim redoslijedom generiranja čvorova. **Koliko više čvorova će biti podrezano s optimalnim redoslijedom u odnosu na slučajni redoslijed generiranja čvorova?**

A 2 B 4 C 1 D 3

3 (T) Ako je za neki problem moguće konstruirati optimističnu heurističku funkciju, usmjereno pretraživanje općenito će biti bolje od slijepog pretraživanja. **Zašto je usmjereno pretraživanje s optimističnom heuristikom općenito (u prosjeku nad svim problemima) bolje od slijepog pretraživanja?**

- A Ima manju apriornu vremensku složenost
- B Nalazi kraći put do rješenja
- C Brže nalazi optimalno rješenje
- D Ne može zaglaviti u beskonačnoj petlji

4 (P) Poželjno svojstvo aloritma pretraživanja jest da je optimalan. Međutim, algoritmi "najbolji prvi" i "uspon na vrh" nisu optimalni. Oba su ova algoritma "pohlepna", ali je algoritam "najbolji prvi" manje pohlepan od algoritma "uspon na vrh", pa u praksi postoji veća vjerojatnost da će algoritam "najboji prvi" pronaći optimalan put. Skicirajte prostor stanja $S = \{a, b, c, d, e, f\}$ s prijelazima $succ(a) = \{b, c\}$, $succ(b) = \{d, e\}$, $succ(c) = succ(d) = succ(e) = \{f\}$. Cijene svih prijelaza su jednake i iznose 1. Heuristika neka je $h(c) = 2$, $h(d) = h(e) = 3$ i $h(f) = 0$. Stanje a je početno stanje, a stanje f je ciljno stanje. **Za koju vrijednost heuristike za stanje b algoritam "uspona na vrh" *ne* pronalazi optimalno rješenje, ali ga algoritam "najbolji prvi" pronalazi?**

A $h(b) = 5$ B $h(b) = 3$ C $h(b) = 4$ D $h(b) = 1$

2. Prikazivanje znanja i automatsko zaključivanje (4 pitanja)

5 (T) Pravila zaključivanja u idealnom su slučaju i ispravna i potpuna. **Kada je pravilo zaključivanja $F \vDash G$ ispravno?**

A Akko je svaki model od F ujedno i model od G

C Akko formula $F \to G$ ima barem jedan model

B Akko je $F \to \neg G$ kontradikcija

D Akko je G deduktivna posljedica od F

6 (R) Zadane su sljedeće tvrdnje: *Jednorog je besmrtan (B), ako je mitsko biće (A). Ako je jednorog besmrtan ili je obična smrtna životinja (C), onda je i rogat (D). Čim je jednorog rogat, onda je i magičan (E). No, ako nije besmrtan, onda je obična smrtna životinja.* **Koja se od sljedećih tvrdnji *ne* može dokazati rezolucijom opovrgavanjem?**

A *Ako je jednorog obična smrtna životinja, onda nije besmrtan.*

B *Ako je jednorog mitsko biće, onda je rogat.*

C *Jednorog je obična smrtna životinja ili rogat.*

D *Jednorog nije mitsko biće, ako nije rogat.*

7 (T) Formalni sustavi za prikazivanje znanja razlikuju se po svojim ontološkim i epistemološkim pretpostavkama. **U čemu se očituje razlika u epistemološkim pretpostavkama između PL i FOL?**

A Dokazivanje logičke posljedice je odlučivo u PL, ali neodlučivo u FOL

B FOL pretpostavlja postojanje objekata i relacija među njima

C PL ne koristi kvantifikatore, ali ima konačan broj interpretacija

D Nema razlike u epistemološkim pretpostavkama između PL i FOL

8 (P) Neka $R(x)$ označava "x je riječ", $U(x)$ označava "x je ugodan" te $I(x)$ označava "x je istinit". Koja od sljedećih formula predikatne logike predstavlja ispravnu formalizaciju rečenice *"Istinite riječi ne moraju biti ugodne"*?

A $\neg\forall x\Big((R(x) \land U(x)) \to I(x)\Big)$

C $\exists x\neg\Big((R(x) \land U(x)) \to I(x)\Big)$

B $\exists x\neg\Big((R(x) \land I(x)) \to U(x)\Big)$

D $\exists x\Big(R(x) \land U(x) \land \neg I(x)\Big)$

3. Logičko programiranje i ekspertni sustavi (2 pitanja)

9 (R) Baza znanja ekspertnog sustava sadrži sljedeća pravila:

(1) AKO $(A = a_2) \land (D = d_2)$ ONDA $C = c_1$

(2) AKO $(F = f_3) \lor (B = b_3)$ ONDA $C = c_2$

(3) AKO $(E = e_1) \lor (B = b_1)$ ONDA $(A = a_1) \land (D = d_2)$

(4) AKO $F = f_1$ ONDA $D = d_1$

(5) AKO $F = f_2$ ONDA $E = e_2$

(6) AKO $(B = b_3) \lor (D = d_1)$ ONDA $(E = e_1) \land (A = a_2)$

Sustav koristimo za izvođenje vrijednosti varijable D ulančavanjem unazad. Prednost imaju pravila s manjim rednim brojem. Pravila koja su jednom palila više ne mogu paliti. Jednom izvedeni međuciljevi brišu se sa stoga, neovisno o njihovoj poziciji na stogu. Na moguće upite od strane sustava, korisnik odgovara sa $B = b_3$ i $F = f_1$. **Što radi ekspertni sustav pri izvođenju vrijednosti varijable D?**

A Završava s tri činjenice u radnoj memoriji

C Pali tri pravila i izvodi $D = d_1$

B Pali dva pravila i izvodi $D = d_2$

D Odbacuje pravilo 3 te kasnije pali pravilo 5

10 (T) Programski jezik Prolog omogućava uporabu negacije. Međutim, za razliku od FOL, uporaba negacije u Prologu je ograničena. **Kako je ograničena uporaba negacije u Prologu?**

A Negacija se ne može koristiti u rekurzivnim definicijama pravila

B Hornove klauzule uopće ne dopuštaju negaciju atoma

C Negirati se može najviše jedan atom na desnoj strani pravila

D Negirati se mogu samo atomi u tijelu pravila (u antecedentu implikacije)

4. Modeliranje neizvjesnosti (2 pitanja)

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

A 0.714 B 0.785 C 0.897 D 0.853

12 (T) Teorija vjerojatnosti i neizrazita logika na različit način modeliraju neizvjesnost znanja. Razmotrite sljedeću situaciju. Jako ste žedni i naišli ste na dvije boce, A i B. Na prvoj boci stoji oznaka $P(A = otrov) = 0.1$, a na drugoj $\mu_{otrov}(B) = 0.1$. Pretpostavite da je otrov smrtonosan samo ako ga se konzumira u punoj koncentraciji. Očito, u ovom slučaju vrijedi $P(A) = \mu(B)$, no boce A i B različitog su sadržaja. **Sadržaj koje boce biste popili, i zašto?**

A A, jer samo jedna od 10 takvih boca sadrži otrov
C A, jer boca sadrži tek blagu količinu otrova
B B, jer samo jedna od 10 takvih boca sadrži otrov
D B, jer boca sadrži tek blagu količinu otrova

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

13 (P) Algoritam ID3 koristi kriterij informacijske dobiti za odabir varijable u svakom čvoru stabla odluke. Razmotrite klasifikaciju skupa primjera D u tri klase, $y \in \{1, 2, 3\}$. Kandidati za korijen stabla su varijable x_1 i x_2. Varijabla x_1 ima moguće vrijednosti a i b, a varijabla x_2 ima moguće vrijednosti c i d. Za varijablu x_1, entropija podskupova $D_{x_1=a}$ i $D_{x_1=b}$ je minimalna moguća. Za varijablu x_2, u skupu D postoje tri primjera sa $x_2 = c$ te šest primjera sa $x_2 = d$. Entropija podskupova $D_{x_2=c}$ i $D_{x_2=d}$ je maksimalna moguća. **Koliko je informacijska dobit varijable x_1 veća od informacijske dobiti varijable x_2?**

A $\log_2 \frac{1}{3}$ B $\log_2 3$ C $\frac{2}{3} \log_2 \frac{1}{3}$ D $\frac{1}{3} \log_2 3$

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 $\mathbf{x}$ 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 x svidio?**

A 0.431 B 0.694 C 0.856 D 0.799

15 (T) Učenje težina umjetne neuronske mreže na odabranom aproksimacijskom problemu formuliramo kao optimizacijski problem koji potom rješavamo nekim optimizacijskim algoritmom. Pri tome učimo aproksimirati ponašanje nekog sustava za koji nemamo analitički opis, ali smo uspjeli mjerenjima prikupiti 1000 primjera. Naučenu mrežu potom želimo koristiti za predviđanje rada sustava. **Što od sljedećega vrijedi?**

A Najbolje je učenje provoditi što duže možemo, kako bi pogreška nad svim primjerima kojima raspolažemo pala na najmanju moguću vrijednost

B Učenje možemo provoditi postupkom propagacije pogreške unatrag i to samo unatražnim prolazima; unaprijedne prolaze možemo preskočiti

C Zahvaljujući teoremu "No free lunch" znamo da su metode koje se temelje na uporabi derivacija (poput algoritma propagacije pogreške unatrag) najbolji optimizacijski algoritmi za ovaj problem

D Prilikom učenja težina ne smijemo koristiti sve primjere za učenje, već dio primjera trebamo koristiti za odluku u kojem trenutku treba prekinuti učenje

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 $(1, (1, 1))$. 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 -0.192488 B -0.024799 C -0.109142 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: $a_1 = \textit{lijevo}$, $a_2 = \textit{desno}$, $a_3 = \textit{sagni-se-i-pokupi-bocu}$. 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** $q(3, a_2)$?

A 0 B 2.75 C 1.25 D -1

6. Prirodom inspirirani optimizacijski algoritmi (3 pitanja)

18 (R) Uporabom mravlje kolonije traži se ciklus kroz graf. Poznati su sljedeći podatci: $\tau_{1,2} = 0.5$, $\tau_{1,5} = \frac{1}{\sqrt{3}}$, $\tau_{1,6} = 2$, $\tau_{2,3} = 10\sqrt{2}$, $\tau_{2,4} = 3$, $\tau_{2,6} = 0.5$, $\tau_{3,4} = 10$, $\tau_{3,5} = 2$, $\tau_{3,7} = \sqrt{3}$, $\tau_{4,7} = 1$, $\tau_{5,6} = \frac{1}{2}$, $\tau_{5,7} = \frac{1}{3}$. $\eta_{1,2} = 2$, $\eta_{1,5} = 3$, $\eta_{1,6} = \frac{1}{\sqrt{2}}$, $\eta_{2,3} = 0.1$, $\eta_{2,4} = 3\sqrt{3}$, $\eta_{2,6} = 0.5$, $\eta_{3,4} = \frac{1}{2\sqrt{5}}$, $\eta_{3,5} = 0.5$, $\eta_{3,7} = 1$, $\eta_{4,7} = \sqrt{2}$, $\eta_{5,6} = 2\sqrt{2}$, $\eta_{5,7} = 3\sqrt{3}$. Također, sve dane vrijednosti su simetrične, tj. $\tau_{i,j} = \tau_{j,i}$ i $\eta_{i,j} = \eta_{j,i}$. Dodatno, $\alpha = 2$ i $\beta = 2$. Prvi mrav kreće iz čvora 1. Kada mrav treba donijeti vjerojatnosnu odluku, pretpostavite da će ishod slučajnog odabira odgovarati najvjerojatnijem. Ako iz nekog čvora mrav ne može dalje, konstrukcija ciklusa se prekida. Uz te pretpostavke odredite ciklus koji će taj mrav konstruirati. **Slijed od koja tri čvora je dio tog ciklusa?**

A 5, 3, 7 B 2, 6, 1 C 5, 6, 2 D Mrav neće uspjeti konstruirati ciklus

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

A Algoritam bi mogao vratiti rješenje koje je lošije od nekog ranije generiranog rješenja

B Porasla bi 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)

20 (P) Genetskim algoritmom tražimo maksimum funkcije $f(x, y, z)$, koristeći binarnu reprezentaciju rješenja. Vrijednost svake od varijabli pretražuje se u intervalu $[10, 90]$, pri čemu je potrebno osigurati da se to pretraživanje provodi barem s preciznošću 0.001. **Od koliko se *minimalno* bitova treba sastojati kromosom?**

A 51 B 48 C 16 D 49

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) Alfa-beta podrezivanje može značajno smanjiti broj generiranih čvorova kod algoritma minimax. Ušteda ovisi o redoslijedu kojim se generiraju čvorovi. Optimalan redoslijed je onaj koji djecu čvora MAX generira silazno po minimax vrijednostima, a djecu čvora MIN uzlazno po minimax vrijednostima. Razmotite igru s dva moguća poteza u svakome stanju. Stablo se pretražuje do dubine 3. Vrijednosti heuristike igrača MAX za stanja u listovima stabla igre, kada su čvorovi-djeca generirani nasumičnim redoslijedom, su sljedeća (slijeva nadesno): $0, 3, -3, 4, 4, -1, 0, 2$. Na potezu je igrač MAX. Odredite broj čvorova koji će biti uklonjeni podrezivanjem alfa-beta. Zatim ponovite izračun, ali s optimalnim redoslijedom generiranja čvorova. **Koliko više čvorova će biti podrezano s optimalnim redoslijedom u odnosu na slučajni redoslijed generiranja čvorova?**

A 2 B 4 C 3 D 1

2 (P) Poželjno svojstvo aloritma pretraživanja jest da je optimalan. Međutim, algoritmi “najbolji prvi” i “uspon na vrh” nisu optimalni. Oba su ova algoritma “pohlepna”, ali je algoritam “najbolji prvi” manje pohlepan od algoritma “uspon na vrh”, pa u praksi postoji veća vjerojatnost da će algoritam “najboji prvi” pronaći optimalan put. Skicirajte prostor stanja $S = \{a, b, c, d, e, f\}$ s prijelazima $succ(a) = \{b, c\}$, $succ(b) = \{d, e\}$, $succ(c) = succ(d) = succ(e) = \{f\}$. Cijene svih prijelaza su jednake i iznose 1. Heuristika neka je $h(b) = 1$, $h(d) = h(e) = 3$ i $h(f) = 0$. Stanje a je početno stanje, a stanje f je ciljno stanje. **Za koju vrijednost heuristike za stanje c algoritam “uspona na vrh” *ne* pronalazi optimalno rješenje, ali ga algoritam ”najbolji prvi” pronalazi?**

A $h(c) = 5$ B $h(c) = 0$ C $h(c) = 4$ D $h(c) = 2$

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

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

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

C Algoritam ne dostiže peti korak

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

4 (T) Ako je za neki problem moguće konstruirati optimističnu heurističku funkciju, usmjereno pretraživanje općenito će biti bolje od slijepog pretraživanja. **Zašto je usmjereno pretraživanje s optimističnom heuristikom općenito (u prosjeku nad svim problemima) bolje od slijepog pretraživanja?**

A Ne može zaglaviti u beskonačnoj petlji

B Ima manju apriornu vremensku složenost

C Brže nalazi optimalno rješenje

D Nalazi kraći put do rješenja

2. Prikazivanje znanja i automatsko zaključivanje (4 pitanja)

5 (R) Zadane su sljedeće tvrdnje: *Jednorog je besmrtan (B), ako je mitsko biće (A). Ako je jednorog besmrtan ili je obična smrtna životinja (C), onda je i rogat (D). Čim je jednorog rogat, onda je i magičan (E). No, ako nije* *besmrtan, onda je obična smrtna životinja.* **Koja se od sljedećih tvrdnji *ne* može dokazati rezolucijom opovrgavanjem?**

A *Ako je jednorog rogat, onda je jednorog rogat.*

B *Ako je jednorog obična smrtna životinja, onda nije besmrtan.*

C *Ako jednorog nije obična smrtna životinja, onda je besmrtan.*

D *Jednorog je magičan.*

6 (P) Neka $R(x)$ označava "x je riječ", $U(x)$ označava "x je ugodan" te $I(x)$ označava "x je istinit". Koja od sljedećih formula predikatne logike predstavlja ispravnu formalizaciju rečenice *"Istinite riječi ne moraju biti ugodne"*?

A $\forall x\Big((R(x) \wedge \neg I(x)) \rightarrow \neg U(x)\Big)$

B $\exists x\Big(R(x) \wedge U(x) \wedge \neg I(x)\Big)$

C $\exists x\Big(\neg(R(x) \wedge I(x)) \vee \neg U(x)\Big)$

D $\exists x \neg\Big((R(x) \wedge I(x)) \rightarrow U(x)\Big)$

7 (T) Pravila zaključivanja u idealnom su slučaju i ispravna i potpuna. **Kada je pravilo zaključivanja $F \vDash G$ ispravno?**

A Akko je G deduktivna posljedica od F

B Akko formula $F \rightarrow G$ ima barem jedan model

C Akko je svaki model od F ujedno i model od G

D Akko je $F \rightarrow \neg G$ kontradikcija

8 (T) Formalni sustavi za prikazivanje znanja razlikuju se po svojim ontološkim i epistemološkim pretpostavkama. **U čemu se očituje razlika u epistemološkim pretpostavkama između PL i FOL?**

A PL ne koristi kvantifikatore, ali ima konačan broj interpretacija

B Nema razlike u epistemološkim pretpostavkama između PL i FOL

C Dokazivanje logičke posljedice je odlučivo u PL, ali neodlučivo u FOL

D FOL pretpostavlja postojanje objekata i relacija među njima

3. Logičko programiranje i ekspertni sustavi (2 pitanja)

9 (R) Baza znanja ekspertnog sustava sadrži sljedeća pravila:

(1) AKO $(A = a_2) \wedge (D = d_2)$ ONDA $C = c_1$
(2) AKO $(F = f_3) \vee (B = b_3)$ ONDA $C = c_2$
(3) AKO $(E = e_1) \vee (B = b_1)$ ONDA $(A = a_1) \wedge (D = d_2)$
(4) AKO $F = f_1$ ONDA $D = d_1$
(5) AKO $F = f_2$ ONDA $E = e_2$
(6) AKO $(B = b_3) \vee (D = d_1)$ ONDA $(E = e_1) \wedge (A = a_2)$

Sustav koristimo za izvođenje vrijednosti varijable D ulančavanjem unazad. Prednost imaju pravila s manjim rednim brojem. Pravila koja su jednom palila više ne mogu paliti. Jednom izvedeni međuciljevi brišu se sa stoga, neovisno o njihovoj poziciji na stogu. Na možebitne upite od strane sustava, korisnik odgovara sa $B = b_3$ i $F = f_1$. **Što radi ekspertni sustav pri izvođenju vrijednosti varijable D?**

A Završava s tri činjenice u radnoj memoriji

B Pali tri pravila i izvodi $D = d_1$

C Izvodi $E = e_2$ te kasnije $C = c_2$

D Pali dva pravila i izvodi $D = d_2$

10 (T) Programski jezik Prolog omogućava uporabu negacije. Međutim, za razliku od FOL, uporaba negacije u Prologu je ograničena. **Kako je ograničena uporaba negacije u Prologu?**

A Negirati se mogu samo atomi u tijelu pravila (u antecedentu implikacije)

B Negacija se ne može koristiti u rekurzivnim definicijama pravila

C Hornove klauzule uopće ne dopuštaju negaciju atoma

D Negirati se može najviše jedan atom na desnoj strani pravila

4. Modeliranje neizvjesnosti (2 pitanja)

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

A 0.897 B 0.714 C 0.785 D 0.853

12 (T) Teorija vjerojatnosti i neizrazita logika na različit način modeliraju neizvjesnost znanja. Razmotrite sljedeću situaciju. Jako ste žedni i naišli ste na dvije boce, A i B. Na prvoj boci stoji oznaka $P(A = otrov) = 0.1$, a na drugoj $\mu_{otrov}(B) = 0.1$. Pretpostavite da je otrov smrtonosan samo ako ga se konzumira u punoj koncentraciji. Očito, u ovom slučaju vrijedi $P(A) = \mu(B)$, no boce A i B različitog su sadržaja. **Sadržaj koje boce biste popili, i zašto?**

A A, jer boca sadrži tek blagu količinu otrova
C B, jer samo jedna od 10 takvih boca sadrži otrov
B A, jer samo jedna od 10 takvih boca sadrži otrov
D B, jer boca sadrži tek blagu količinu otrova

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 $(1, (0, 1))$. 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 -0.192488 B -0.109142 C -0.058547 D -0.024799

14 (T) Učenje težina umjetne neuronske mreže na odabranom aproksimacijskom problemu formuliramo kao optimizacijski problem koji potom rješavamo nekim optimizacijskim algoritmom. Pri tome učimo aproksimirati ponašanje nekog sustava za koji nemamo analitički opis, ali smo uspjeli mjerenjima prikupiti 1000 primjera. Naučenu mrežu potom želimo koristiti za predviđanje rada sustava. **Što od sljedećega vrijedi?**

A Najbolje je učenje provoditi što duže možemo, kako bi pogreška nad svim primjerima kojima raspolažemo pala na najmanju moguću vrijednost

B Zahvaljujući teoremu “No free lunch” znamo da su metode koje se temelje na uporabi derivacija (poput algoritma propagacije pogreške unatrag) najbolji optimizacijski algoritmi za ovaj problem

C Prilikom učenja težina ne smijemo koristiti sve primjere za učenje, već dio primjera trebamo koristiti za odluku u kojem trenutku treba prekinuti učenje

D Učenje možemo provoditi postupkom propagacije pogreške unatrag i to samo unatražnim prolazima; unaprijedne prolaze možemo preskočiti

15 (P) Algoritam ID3 koristi kriterij informacijske dobiti za odabir varijable u svakom čvoru stabla odluke. Razmotrite klasifikaciju skupa primjera D u tri klase, $y \in \{1, 2, 3\}$. Kandidati za korijen stabla su varijable x_1 i x_2. Varijabla x_1 ima moguće vrijednosti a i b, a varijabla x_2 ima moguće vrijednosti c i d. Za varijablu x_1, entropija podskupova $D_{x_1=a}$ i $D_{x_1=b}$ je minimalna moguća. Za varijablu x_2, u skupu D postoje tri primjera sa $x_2 = c$ te šest primjera sa $x_2 = d$. Entropija podskupova $D_{x_2=c}$ i $D_{x_2=d}$ je maksimalna moguća. **Koliko je informacijska dobit varijable x_1 veća od informacijske dobiti varijable x_2?**

A $\log_2 3$ B $\log_2 \frac{1}{3}$ C $\frac{1}{3} \log_2 3$ D $\frac{2}{3} \log_2 \frac{1}{3}$

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

i	Evaluacija	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 x svidio?**

A 0.799 B 0.694 C 0.431 D 0.856

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: $a_1 = \textit{lijevo}$, $a_2 = \textit{desno}$, $a_3 = \textit{sagni-se-i-pokupi-bocu}$. 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** $q(3, a_2)$?

A 0 B -1 C 1.25 D 2.75

6. Prirodom inspirirani optimizacijski algoritmi (3 pitanja)

18 (P) Genetskim algoritmom tražimo maksimum funkcije $f(x, y, z)$, koristeći binarnu reprezentaciju rješenja. Vrijednost svake od varijabli pretražuje se u intervalu $[-20, 10]$, pri čemu je potrebno osigurati da se to pretraživanje provodi barem s preciznošću 0.05. **Od koliko se *minimalno* bitova treba sastojati kromosom?**

A 28 B 27 C 30 D 9

19 (R) Uporabom mravlje kolonije traži se ciklus kroz graf. Poznati su sljedeći podatci: $\tau_{1,2} = 0.5$, $\tau_{1,5} = \frac{1}{\sqrt{3}}$, $\tau_{1,6} = 2$, $\tau_{2,3} = 10\sqrt{2}$, $\tau_{2,4} = 3$, $\tau_{2,6} = 0.5$, $\tau_{3,4} = 10$, $\tau_{3,5} = 2$, $\tau_{3,7} = \sqrt{3}$, $\tau_{4,7} = 1$, $\tau_{5,6} = \frac{1}{2}$, $\tau_{5,7} = \frac{1}{3}$. $\eta_{1,2} = 2$, $\eta_{1,5} = 3$, $\eta_{1,6} = \frac{1}{\sqrt{2}}$, $\eta_{2,3} = 0.1$, $\eta_{2,4} = 3\sqrt{3}$, $\eta_{2,6} = 0.5$, $\eta_{3,4} = \frac{1}{2\sqrt{5}}$, $\eta_{3,5} = 0.5$, $\eta_{3,7} = 1$, $\eta_{4,7} = \sqrt{2}$, $\eta_{5,6} = 2\sqrt{2}$, $\eta_{5,7} = 3\sqrt{3}$. Također, sve dane vrijednosti su simetrične, tj. $\tau_{i,j} = \tau_{j,i}$ i $\eta_{i,j} = \eta_{j,i}$. Dodatno, $\alpha = 2$ i $\beta = 2$. Prvi mrav kreće iz čvora 1. Kada mrav treba donijeti vjerojatnosnu odluku, pretpostavite da će ishod slučajnog odabira odgovarati najvjerojatnijem. Ako iz nekog čvora mrav ne može dalje, konstrukcija ciklusa se prekida. Uz te pretpostavke odredite ciklus koji će taj mrav konstruirati. **Slijed od koja tri čvora je dio tog ciklusa?**

A Mrav neće uspjeti konstruirati ciklus B 7, 3, 4 C 5, 6, 2 D 1, 5, 6

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

A Porasla bi 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

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

- A Algoritam ne dostiže peti korak
- B $O = [(f, 19)]$, $C = \{(a, 0), (b, 2), (c, 2), (d, 3), (e, 5)\}$
- C $O = [(d, 3), (f, 20)]$, $C = \{(a, 0), (b, 2), (c, 2), (e, 6)\}$
- D $O = [(d, 4), (f, 20)]$, $C = \{(a, 16), (b, 6), (c, 14), (e, 2)\}$

2 (P) Alfa-beta podrezivanje može značajno smanjiti broj generiranih čvorova kod algoritma minimax. Ušteda ovisi o redoslijedu kojim se generiraju čvorovi. Optimalan redoslijed je onaj koji djecu čvora MAX generira silazno po minimax vrijednostima, a djecu čvora MIN uzlazno po minimax vrijednostima. Razmotite igru s dva moguća poteza u svakome stanju. Stablo se pretražuje do dubine 3. Vrijednosti heuristike igrača MAX za stanja u listovima stabla igre, kada su čvorovi-djeca generirani nasumičnim redoslijedom, su sljedeća (slijeva nadesno): $0, 3, -3, 4, 4, -1, 0, 2$. Na potezu je igrač MAX. Odredite broj čvorova koji će biti uklonjeni podrezivanjem alfa-beta. Zatim ponovite izračun, ali s optimalnim redoslijedom generiranja čvorova. **Koliko više čvorova će biti podrezano s optimalnim redoslijedom u odnosu na slučajni redoslijed generiranja čvorova?**

A 2 B 3 C 1 D 4

3 (T) Ako je za neki problem moguće konstruirati optimističnu heurističku funkciju, usmjereno pretraživanje općenito će biti bolje od slijepog pretraživanja. **Zašto je usmjereno pretraživanje s optimističnom heuristikom općenito (u prosjeku nad svim problemima) bolje od slijepog pretraživanja?**

- A Brže nalazi optimalno rješenje
- B Ne može zaglaviti u beskonačnoj petlji
- C Ima manju apriornu vremensku složenost
- D Nalazi kraći put do rješenja

4 (P) Poželjno svojstvo aloritma pretraživanja jest da je optimalan. Međutim, algoritmi "najbolji prvi" i "uspon na vrh" nisu optimalni. Oba su ova algoritma "pohlepna", ali je algoritam "najbolji prvi" manje pohlepan od algoritma "uspon na vrh", pa u praksi postoji veća vjerojatnost da će algoritam "najboji prvi" pronaći optimalan put. Skicirajte prostor stanja $S = \{a, b, c, d, e, f\}$ s prijelazima $succ(a) = \{b, c\}$, $succ(b) = \{d, e\}$, $succ(c) = succ(d) = succ(e) = \{f\}$. Cijene svih prijelaza su jednake i iznose 1. Heuristika neka je $h(b) = 1$, $h(d) = h(e) = 3$ i $h(f) = 0$. Stanje a je početno stanje, a stanje f je ciljno stanje. **Za koju vrijednost heuristike za stanje c algoritam "uspona na vrh" *ne* pronalazi optimalno rješenje, ali ga algoritam "najbolji prvi" pronalazi?**

A $h(c) = 5$ B $h(c) = 0$ C $h(c) = 4$ D $h(c) = 2$

2. Prikazivanje znanja i automatsko zaključivanje (4 pitanja)

5 (P) Neka $R(x)$ označava "x je riječ", $U(x)$ označava "x je ugodan" te $I(x)$ označava "x je istinit". Koja od sljedećih formula predikatne logike predstavlja ispravnu formalizaciju rečenice *"Istinite riječi ne moraju biti ugodne"*?

A $\exists x\Big(R(x) \land U(x) \land \neg I(x)\Big)$ C $\exists x\neg\Big(R(x) \land I(x) \land U(x)\Big)$

B $\exists x\Big(R(x) \land I(x) \land U(x)\Big)$ D $\exists x\neg\Big((R(x) \land I(x)) \rightarrow U(x)\Big)$

6 (T) Formalni sustavi za prikazivanje znanja razlikuju se po svojim ontološkim i epistemološkim pretpostavkama. **U čemu se očituje razlika u epistemološkim pretpostavkama između PL i FOL?**

A Dokazivanje logičke posljedice je odlučivo u PL, ali neodlučivo u FOL

B Nema razlike u epistemološkim pretpostavkama između PL i FOL

C PL ne koristi kvantifikatore, ali ima konačan broj interpretacija

D FOL pretpostavlja postojanje objekata i relacija među njima

7 (T) Pravila zaključivanja u idealnom su slučaju i ispravna i potpuna. **Kada je pravilo zaključivanja $F \vDash G$ ispravno?**

A Akko je G deduktivna posljedica od F C Akko je svaki model od F ujedno i model od G

B Akko formula $F \rightarrow G$ ima barem jedan model D Akko je $F \rightarrow \neg G$ kontradikcija

8 (R) Zadane su sljedeće tvrdnje: *Jednorog je besmrtan (B), ako je mitsko biće (A). Ako je jednorog besmrtan ili je obična smrtna životinja (C), onda je i rogat (D). Čim je jednorog rogat, onda je i magičan (E). No, ako nije besmrtan, onda je obična smrtna životinja.* **Koja se od sljedećih tvrdnji *ne* može dokazati rezolucijom opovrgavanjem?**

A *Ako je jednorog mitsko biće, onda je rogat.* C *Jednorog je mitsko biće.*

B *Ako je jednorog rogat, onda je jednorog rogat.* D *Jednorog je obična smrtna životinja ili rogat.*

3. Logičko programiranje i ekspertni sustavi (2 pitanja)

9 (R) Baza znanja ekspertnog sustava sadrži sljedeća pravila:

(1) AKO $(A = a_2) \land (D = d_2)$ ONDA $C = c_1$
(2) AKO $(F = f_3) \lor (B = b_3)$ ONDA $C = c_2$
(3) AKO $(E = e_1) \lor (B = b_1)$ ONDA $(A = a_1) \land (D = d_2)$
(4) AKO $F = f_1$ ONDA $D = d_1$
(5) AKO $F = f_2$ ONDA $E = e_2$
(6) AKO $(B = b_3) \lor (D = d_1)$ ONDA $(E = e_1) \land (A = a_2)$

Sustav koristimo za izvođenje vrijednosti varijable D ulančavanjem unazad. Prednost imaju pravila s manjim rednim brojem. Pravila koja su jednom palila više ne mogu paliti. Jednom izvedeni međuciljevi brišu se sa stoga, neovisno o njihovoj poziciji na stogu. Na možebitne upite od strane sustava, korisnik odgovara sa $B = b_3$ i $F = f_1$. **Što radi ekspertni sustav pri izvođenju vrijednosti varijable D?**

A Odbacuje pravilo 3 te kasnije pali pravilo 5 C Završava s tri činjenice u radnoj memoriji

B Pali dva pravila i izvodi $D = d_2$ D Izvodi $E = e_2$ te kasnije $C = c_2$

10 (T) Programski jezik Prolog omogućava uporabu negacije. Međutim, za razliku od FOL, uporaba negacije u Prologu je ograničena. **Kako je ograničena uporaba negacije u Prologu?**

A Negacija se ne može koristiti u rekurzivnim definicijama pravila

B Negirati se mogu samo atomi u tijelu pravila (u antecedentu implikacije)

C Hornove klauzule uopće ne dopuštaju negaciju atoma

D Negirati se može najviše jedan atom na desnoj strani pravila

4. Modeliranje neizvjesnosti (2 pitanja)

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

A 0.897 B 0.785 C 0.714 D 0.853

12 (T) Teorija vjerojatnosti i neizrazita logika na različit način modeliraju neizvjesnost znanja. Razmotrite sljedeću situaciju. Jako ste žedni i naišli ste na dvije boce, A i B. Na prvoj boci stoji oznaka $P(A = otrov) = 0.1$, a na drugoj $\mu_{otrov}(B) = 0.1$. Pretpostavite da je otrov smrtonosan samo ako ga se konzumira u punoj koncentraciji. Očito, u ovom slučaju vrijedi $P(A) = \mu(B)$, no boce A i B različitog su sadržaja. **Sadržaj koje boce biste popili, i zašto?**

A A, jer boca sadrži tek blagu količinu otrova
B B, jer boca sadrži tek blagu količinu otrova
C A, jer samo jedna od 10 takvih boca sadrži otrov
D B, jer samo jedna od 10 takvih boca sadrži otrov

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

13 (P) Algoritam ID3 koristi kriterij informacijske dobiti za odabir varijable u svakom čvoru stabla odluke. Razmotrite klasifikaciju skupa primjera D u tri klase, $y \in \{1, 2, 3\}$. Kandidati za korijen stabla su varijable x_1 i x_2. Varijabla x_1 ima moguće vrijednosti a i b, a varijabla x_2 ima moguće vrijednosti c i d. Za varijablu x_1, entropija podskupova $D_{x_1=a}$ i $D_{x_1=b}$ je minimalna moguća. Za varijablu x_2, u skupu D postoje tri primjera sa $x_2 = c$ te šest primjera sa $x_2 = d$. Entropija podskupova $D_{x_2=c}$ i $D_{x_2=d}$ je maksimalna moguća. **Koliko je informacijska dobit varijable x_1 veća od informacijske dobiti varijable x_2?**

A $\log_2 \frac{1}{3}$ B $\log_2 3$ C $\frac{2}{3} \log_2 \frac{1}{3}$ D $\frac{1}{3} \log_2 3$

14 (T) Učenje težina umjetne neuronske mreže na odabranom aproksimacijskom problemu formuliramo kao optimizacijski problem koji potom rješavamo nekim optimizacijskim algoritmom. Pri tome učimo aproksimirati ponašanje nekog sustava za koji nemamo analitički opis, ali smo uspjeli mjerenjima prikupiti 1000 primjera. Naučenu mrežu potom želimo koristiti za predviđanje rada sustava. **Što od sljedećega vrijedi?**

A Zahvaljujući teoremu "No free lunch" znamo da su metode koje se temelje na uporabi derivacija (poput algoritma propagacije pogreške unatrag) najbolji optimizacijski algoritmi za ovaj problem

B Prilikom učenja težina ne smijemo koristiti sve primjere za učenje, već dio primjera trebamo koristiti za odluku u kojem trenutku treba prekinuti učenje

C Učenje možemo provoditi postupkom propagacije pogreške unatrag i to samo unatražnim prolazima; unaprijedne prolaze možemo preskočiti

D Najbolje je učenje provoditi što duže možemo, kako bi pogreška nad svim primjerima kojima raspolažemo pala na najmanju moguću vrijednost

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

i	Evaluacija	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 $\mathbf{x}$ sa sljedećim karakteristikama: $\mathbf{x} = (\text{striktna}, \text{interpreter}, \text{hibridna}, \text{dinamička})$. 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 0.856 B 0.694 C 0.431 D 0.799

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 $(1, (0, 1))$. 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 -0.109142 B -0.192488 C -0.024799 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: $a_1 =$ *lijevo*, $a_2 =$ *desno*, $a_3 =$ *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** $q(3, a_2)$?

A 1.25 B -2.5 C -1 D 2

6. Prirodom inspirirani optimizacijski algoritmi (3 pitanja)

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

A Algoritam bi generirao rješenja koja su neispravna (izvan intervala pretrage)

B Porasla bi 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

19 (R) Uporabom mravlje kolonije traži se ciklus kroz graf. Poznati su sljedeći podatci: $\tau_{1,2} = \frac{1}{\sqrt{3}}$, $\tau_{1,4} = 2$, $\tau_{1,6} = 0.5$, $\tau_{2,4} = \frac{1}{2}$, $\tau_{2,5} = \frac{1}{3}$, $\tau_{2,7} = 2$, $\tau_{3,5} = 1$, $\tau_{3,6} = 3$, $\tau_{3,7} = 10$, $\tau_{4,6} = 0.5$, $\tau_{5,7} = \sqrt{3}$, $\tau_{6,7} = 10\sqrt{2}$, $\eta_{1,2} = 3$, $\eta_{1,4} = \frac{1}{\sqrt{2}}$, $\eta_{1,6} = 2$, $\eta_{2,4} = 2\sqrt{2}$, $\eta_{2,5} = 3\sqrt{3}$, $\eta_{2,7} = 0.5$, $\eta_{3,5} = \sqrt{2}$, $\eta_{3,6} = 3\sqrt{3}$, $\eta_{3,7} = \frac{1}{2\sqrt{5}}$, $\eta_{4,6} = 0.5, \eta_{5,7} = 1, \eta_{6,7} = 0.1$. Također, sve dane vrijednosti su simetrične, tj. $\tau_{i,j} = \tau_{j,i}$ i $\eta_{i,j} = \eta_{j,i}$. Dodatno, $\alpha = 2$ i $\beta = 2$. Prvi mrav kreće iz čvora 1. Kada mrav treba donijeti vjerojatnosnu odluku, pretpostavite da će ishod slučajnog odabira odgovarati najvjerojatnijem. Ako iz nekog čvora mrav ne može dalje, konstrukcija ciklusa se prekida. Uz te pretpostavke odredite ciklus koji će taj mrav konstruirati. **Slijed od koja tri čvora je dio tog ciklusa?**

A 6, 4, 2 B Mrav neće uspjeti konstruirati ciklus C 5, 7, 3 D 5, 3, 6

20 (P) Genetskim algoritmom tražimo maksimum funkcije $f(x, y, z)$, koristeći binarnu reprezentaciju rješenja. Vrijednost svake od varijabli pretražuje se u intervalu $[10, 90]$, pri čemu je potrebno osigurati da se to pretraživanje provodi barem s preciznošću 0.001. **Od koliko se *minimalno* bitova treba sastojati kromosom?**

A 16 B 49 C 48 D 51

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

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

B Algoritam ne dostiže peti korak

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

D $O = [(f, 19)]$, $C = \{(a, 0), (b, 2), (c, 2), (d, 3), (e, 5)\}$

2 (P) Poželjno svojstvo aloritma pretraživanja jest da je optimalan. Međutim, algoritmi "najbolji prvi" i "uspon na vrh" nisu optimalni. Oba su ova algoritma "pohlepna", ali je algoritam "najbolji prvi" manje pohlepan od algoritma "uspon na vrh", pa u praksi postoji veća vjerojatnost da će algoritam "najboji prvi" pronaći optimalan put. Skicirajte prostor stanja $S = \{a, b, c, d, e, f\}$ s prijelazima $succ(a) = \{b, c\}$, $succ(b) = \{d, e\}$, $succ(c) = succ(d) = succ(e) = \{f\}$. Cijene svih prijelaza su jednake i iznose 1. Heuristika neka je $h(b) = 1$, $h(d) = h(e) = 3$ i $h(f) = 0$. Stanje a je početno stanje, a stanje f je ciljno stanje. **Za koju vrijednost heuristike za stanje c algoritam "uspona na vrh" *ne* pronalazi optimalno rješenje, ali ga algoritam "najbolji prvi" pronalazi?**

A $h(c) = 5$ B $h(c) = 4$ C $h(c) = 0$ D $h(c) = 2$

3 (T) Ako je za neki problem moguće konstruirati optimističnu heurističku funkciju, usmjereno pretraživanje općenito će biti bolje od slijepog pretraživanja. **Zašto je usmjereno pretraživanje s optimističnom heuristikom općenito (u prosjeku nad svim problemima) bolje od slijepog pretraživanja?**

A Nalazi kraći put do rješenja

B Ne može zaglaviti u beskonačnoj petlji

C Ima manju apriornu vremensku složenost

D Brže nalazi optimalno rješenje

4 (P) Alfa-beta podrezivanje može značajno smanjiti broj generiranih čvorova kod algoritma minimax. Ušteda ovisi o redoslijedu kojim se generiraju čvorovi. Optimalan redoslijed je onaj koji djecu čvora MAX generira silazno po minimax vrijednostima, a djecu čvora MIN uzlazno po minimax vrijednostima. Razmotite igru s dva moguća poteza u svakome stanju. Stablo se pretražuje do dubine 3. Vrijednosti heuristike igrača MAX za stanja u listovima stabla igre, kada su čvorovi-djeca generirani nasumičnim redoslijedom, su sljedeća (slijeva nadesno): $0, 3, -3, 4, 4, -1, 0, 2$. Na potezu je igrač MAX. Odredite broj čvorova koji će biti uklonjeni podrezivanjem alfa-beta. Zatim ponovite izračun, ali s optimalnim redoslijedom generiranja čvorova. **Koliko više čvorova će biti podrezano s optimalnim redoslijedom u odnosu na slučajni redoslijed generiranja čvorova?**

A 4 B 1 C 3 D 2

2. Prikazivanje znanja i automatsko zaključivanje (4 pitanja)

5 (T) Formalni sustavi za prikazivanje znanja razlikuju se po svojim ontološkim i epistemološkim pretpostavkama. **U čemu se očituje razlika u epistemološkim pretpostavkama između PL i FOL?**

- A FOL pretpostavlja postojanje objekata i relacija među njima
- B PL ne koristi kvantifikatore, ali ima konačan broj interpretacija
- C Dokazivanje logičke posljedice je odlučivo u PL, ali neodlučivo u FOL
- D Nema razlike u epistemološkim pretpostavkama između PL i FOL

6 (T) Pravila zaključivanja u idealnom su slučaju i ispravna i potpuna. **Kada je pravilo zaključivanja $F \vDash G$ ispravno?**

- A Akko je svaki model od F ujedno i model od G
- B Akko je $F \rightarrow \neg G$ kontradikcija
- C Akko formula $F \rightarrow G$ ima barem jedan model
- D Akko je G deduktivna posljedica od F

7 (R) Zadane su sljedeće tvrdnje: *Jednorog je besmrtan (B), ako je mitsko biće (A). Ako je jednorog besmrtan ili je obična smrtna životinja (C), onda je i rogat (D). Čim je jednorog rogat, onda je i magičan (E). No, ako nije besmrtan, onda je obična smrtna životinja.* **Koja se od sljedećih tvrdnji *ne* može dokazati rezolucijom opovrgavanjem?**

- A *Jednorog je mitsko biće.*
- B *Jednorog nije mitsko biće, ako nije rogat.*
- C *Jednorog je obična smrtna životinja ili rogat.*
- D *Ako je jednorog mitsko biće, onda je rogat.*

8 (P) Neka $R(x)$ označava "x je riječ", $U(x)$ označava "x je ugodan" te $I(x)$ označava "x je istinit". Koja od sljedećih formula predikatne logike predstavlja ispravnu formalizaciju rečenice *"Istinite riječi ne moraju biti ugodne"*?

- A $\exists x\Big(R(x) \land U(x) \land \neg I(x)\Big)$
- B $\exists x\Big((R(x) \land I(x)) \rightarrow \neg U(x)\Big)$
- C $\forall x \neg\Big((R(x) \land I(x)) \rightarrow U(x)\Big)$
- D $\exists x \neg\Big((R(x) \land I(x)) \rightarrow U(x)\Big)$

3. Logičko programiranje i ekspertni sustavi (2 pitanja)

9 (R) Baza znanja ekspertnog sustava sadrži sljedeća pravila:

(1) AKO $(A = a_2) \land (D = d_2)$ ONDA $C = c_1$
(2) AKO $(F = f_3) \lor (B = b_3)$ ONDA $C = c_2$
(3) AKO $(E = e_1) \lor (B = b_1)$ ONDA $(A = a_1) \land (D = d_2)$
(4) AKO $F = f_1$ ONDA $D = d_1$
(5) AKO $F = f_2$ ONDA $E = e_2$
(6) AKO $(B = b_3) \lor (D = d_1)$ ONDA $(E = e_1) \land (A = a_2)$

Sustav koristimo za izvođenje vrijednosti varijable D ulančavanjem unazad. Prednost imaju pravila s manjim rednim brojem. Pravila koja su jednom palila više ne mogu paliti. Jednom izvedeni međuciljevi brišu se sa stoga, neovisno o njihovoj poziciji na stogu. Na moguće upite od strane sustava, korisnik odgovara sa $B = b_3$ i $F = f_1$. **Što radi ekspertni sustav pri izvođenju vrijednosti varijable D?**

- A Pali dva pravila i izvodi $D = d_2$
- B Odbacuje pravilo 3 te kasnije pali pravilo 5
- C Završava s tri činjenice u radnoj memoriji
- D Pali tri pravila i izvodi $D = d_1$

10 (T) Programski jezik Prolog omogućava uporabu negacije. Međutim, za razliku od FOL, uporaba negacije u Prologu je ograničena. **Kako je ograničena uporaba negacije u Prologu?**

- A Negirati se mogu samo atomi u tijelu pravila (u antecedentu implikacije)
- B Hornove klauzule uopće ne dopuštaju negaciju atoma
- C Negacija se ne može koristiti u rekurzivnim definicijama pravila
- D Negirati se može najviše jedan atom na desnoj strani pravila

4. Modeliranje neizvjesnosti (2 pitanja)

11 (T) Teorija vjerojatnosti i neizrazita logika na različit način modeliraju neizvjesnost znanja. Razmotrite sljedeću situaciju. Jako ste žedni i naišli ste na dvije boce, A i B. Na prvoj boci stoji oznaka $P(A = otrov) = 0.1$, a na drugoj $\mu_{otrov}(B) = 0.1$. Pretpostavite da je otrov smrtonosan samo ako ga se konzumira u punoj koncentraciji. Očito, u ovom slučaju vrijedi $P(A) = \mu(B)$, no boce A i B različitog su sadržaja. **Sadržaj koje boce biste popili, i zašto?**

A A, jer samo jedna od 10 takvih boca sadrži otrov

C B, jer boca sadrži tek blagu količinu otrova

B B, jer samo jedna od 10 takvih boca sadrži otrov

D A, jer boca sadrži tek blagu količinu otrova

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

A 0.853 B 0.785 C 0.714 D 0.897

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

13 (P) Algoritam ID3 koristi kriterij informacijske dobiti za odabir varijable u svakom čvoru stabla odluke. Razmotrite klasifikaciju skupa primjera D u tri klase, $y \in \{1, 2, 3\}$. Kandidati za korijen stabla su varijable x_1 i x_2. Varijabla x_1 ima moguće vrijednosti a i b, a varijabla x_2 ima moguće vrijednosti c i d. Za varijablu x_1, entropija podskupova $D_{x_1=a}$ i $D_{x_1=b}$ je minimalna moguća. Za varijablu x_2, u skupu D postoje tri primjera sa $x_2 = c$ te šest primjera sa $x_2 = d$. Entropija podskupova $D_{x_2=c}$ i $D_{x_2=d}$ je maksimalna moguća. **Koliko je informacijska dobit varijable x_1 veća od informacijske dobiti varijable x_2?**

A $\log_2 \frac{1}{3}$ B $\log_2 3$ C $\frac{1}{3} \log_2 3$ D $\frac{2}{3} \log_2 \frac{1}{3}$

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	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 $\mathbf{x}$ 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 x svidio?**

A 0.431 B 0.856 C 0.799 D 0.694

15 (T) Učenje težina umjetne neuronske mreže na odabranom aproksimacijskom problemu formuliramo kao optimizacijski problem koji potom rješavamo nekim optimizacijskim algoritmom. Pri tome učimo aproksimirati ponašanje nekog sustava za koji nemamo analitički opis, ali smo uspjeli mjerenjima prikupiti 1000 primjera. Naučenu mrežu potom želimo koristiti za predviđanje rada sustava. **Što od sljedećega vrijedi?**

- A Zahvaljujući teoremu "No free lunch" znamo da su metode koje se temelje na uporabi derivacija (poput algoritma propagacije pogreške unatrag) najbolji optimizacijski algoritmi za ovaj problem
- B Prilikom učenja težina ne smijemo koristiti sve primjere za učenje, već dio primjera trebamo koristiti za odluku u kojem trenutku treba prekinuti učenje
- C Učenje možemo provoditi postupkom propagacije pogreške unatrag i to samo unatražnim prolazima; unaprijedne prolaze možemo preskočiti
- D Najbolje je učenje provoditi što duže možemo, kako bi pogreška nad svim primjerima kojima raspolažemo pala na najmanju moguću 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 $(1, (0, 0))$. 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 -0.109142 C -0.024799 D -0.192488

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: $a_1 =$ *lijevo*, $a_2 =$ *desno*, $a_3 =$ *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** $q(3, a_2)$?

A 8 B 3 C 4.25 D 2.25

6. Prirodom inspirirani optimizacijski algoritmi (3 pitanja)

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

- A Porasla bi vjerojatnost da pretraga zapne u lokalnome optimumu
- B Porasla bi prosječna vrijednost funkcije dobrote u populaciji
- 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 (P) Genetskim algoritmom tražimo maksimum funkcije $f(x, y, z)$, koristeći binarnu reprezentaciju rješenja. Vrijednost svake od varijabli pretražuje se u intervalu $[-100, 100]$, pri čemu je potrebno osigurati da se to pretraživanje provodi barem s preciznošću 0.001. **Od koliko se *minimalno* bitova treba sastojati kromosom?**

A 54 B 51 C 53 D 17

20 (R) Uporabom mravlje kolonije traži se ciklus kroz graf. Poznati su sljedeći podatci: $\tau_{1,3} = 0.5$, $\tau_{1,6} = \frac{1}{\sqrt{3}}$, $\tau_{1,7} = 2$, $\tau_{2,4} = 1$, $\tau_{2,5} = \sqrt{3}$, $\tau_{2,6} = \frac{1}{3}$, $\tau_{3,4} = 3$, $\tau_{3,5} = 10\sqrt{2}$, $\tau_{3,7} = 0.5$, $\tau_{4,5} = 10$, $\tau_{5,6} = 2$, $\tau_{6,7} = \frac{1}{2}$. $\eta_{1,3} = 2$, $\eta_{1,6} = 3$, $\eta_{1,7} = \frac{1}{\sqrt{2}}$, $\eta_{2,4} = \sqrt{2}$, $\eta_{2,5} = 1$, $\eta_{2,6} = 3\sqrt{3}$, $\eta_{3,4} = 3\sqrt{3}$, $\eta_{3,5} = 0.1$, $\eta_{3,7} = 0.5$, $\eta_{4,5} = \frac{1}{2\sqrt{5}}$, $\eta_{5,6} = 0.5$, $\eta_{6,7} = 2\sqrt{2}$. Također, sve dane vrijednosti su simetrične, tj. $\tau_{i,j} = \tau_{j,i}$ i $\eta_{i,j} = \eta_{j,i}$. Dodatno, $\alpha = 2$ i $\beta = 2$. Prvi mrav kreće iz čvora 1. Kada mrav treba donijeti vjerojatnosnu odluku, pretpostavite da će ishod slučajnog odabira odgovarati najvjerojatnijem. Ako iz nekog čvora mrav ne može dalje, konstrukcija ciklusa se prekida. Uz te pretpostavke odredite ciklus koji će taj mrav konstruirati. **Slijed od koja tri čvora je dio tog ciklusa?**

A 6, 5, 2 B Mrav neće uspjeti konstruirati ciklus C 4, 3, 7 D 2, 4, 3

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

- A Algoritam ne dostiže šesti korak
- B $O = [(e, 5), (f, 20)]$, $C = \{(a, 0), (b, 2), (c, 2), (d, 3)\}$
- C $O = [(e, 2), (f, 0)]$, $C = \{(a, 16), (b, 6), (c, 14), (d, 3)\}$
- D $O = [(c, 2), (f, 20)]$, $C = \{(a, 0), (b, 2), (d, 4), (e, 6)\}$

2 (P) Alfa-beta podrezivanje može značajno smanjiti broj generiranih čvorova kod algoritma minimax. Ušteda ovisi o redoslijedu kojim se generiraju čvorovi. Optimalan redoslijed je onaj koji djecu čvora MAX generira silazno po minimax vrijednostima, a djecu čvora MIN uzlazno po minimax vrijednostima. Razmotite igru s dva moguća poteza u svakome stanju. Stablo se pretražuje do dubine 3. Vrijednosti heuristike igrača MAX za stanja u listovima stabla igre, kada su čvorovi-djeca generirani nasumičnim redoslijedom, su sljedeća (slijeva nadesno): $0, 3, -3, 4, 4, -1, 0, 2$. Na potezu je igrač MAX. Odredite broj čvorova koji će biti uklonjeni podrezivanjem alfa-beta. Zatim ponovite izračun, ali s optimalnim redoslijedom generiranja čvorova. **Koliko više čvorova će biti podrezano s optimalnim redoslijedom u odnosu na slučajni redoslijed generiranja čvorova?**

A 4 B 2 C 1 D 3

3 (T) Ako je za neki problem moguće konstruirati optimističnu heurističku funkciju, usmjereno pretraživanje općenito će biti bolje od slijepog pretraživanja. **Zašto je usmjereno pretraživanje s optimističnom heuristikom općenito (u prosjeku nad svim problemima) bolje od slijepog pretraživanja?**

- A Brže nalazi optimalno rješenje
- B Ne može zaglaviti u beskonačnoj petlji
- C Ima manju apriornu vremensku složenost
- D Nalazi kraći put do rješenja

4 (P) Poželjno svojstvo aloritma pretraživanja jest da je optimalan. Međutim, algoritmi "najbolji prvi" i "uspon na vrh" nisu optimalni. Oba su ova algoritma "pohlepna", ali je algoritam "najbolji prvi" manje pohlepan od algoritma "uspon na vrh", pa u praksi postoji veća vjerojatnost da će algoritam "najboji prvi" pronaći optimalan put. Skicirajte prostor stanja $S = \{a, b, c, d, e, f\}$ s prijelazima $succ(a) = \{b, c\}$, $succ(b) = \{d, e\}$, $succ(c) = succ(d) = succ(e) = \{f\}$. Cijene svih prijelaza su jednake i iznose 1. Heuristika neka je $h(c) = 2$, $h(d) = h(e) = 3$ i $h(f) = 0$. Stanje a je početno stanje, a stanje f je ciljno stanje. **Za koju vrijednost heuristike za stanje b algoritam "uspona na vrh" *ne* pronalazi optimalno rješenje, ali ga algoritam "najbolji prvi" pronalazi?**

A $h(b) = 3$ B $h(b) = 5$ C $h(b) = 0$ D $h(b) = 4$

2. Prikazivanje znanja i automatsko zaključivanje (4 pitanja)

5 (R) Zadane su sljedeće tvrdnje: *Jednorog je besmrtan (B), ako je mitsko biće (A). Ako je jednorog besmrtan ili je obična smrtna životinja (C), onda je i rogat (D). Čim je jednorog rogat, onda je i magičan (E). No, ako nije besmrtan, onda je obična smrtna životinja.* **Koja se od sljedećih tvrdnji *ne* može dokazati rezolucijom opovrgavanjem?**

- A *Jednorog je obična smrtna životinja ili rogat.*
- B *Ako je jednorog rogat, onda je jednorog rogat.*
- C *Ako jednorog nije obična smrtna životinja, onda je besmrtan.*
- D *Ako je jednorog obična smrtna životinja, onda nije besmrtan.*

6 (T) Pravila zaključivanja u idealnom su slučaju i ispravna i potpuna. **Kada je pravilo zaključivanja $F \vDash G$ ispravno?**

- A Akko je svaki model od F ujedno i model od G
- B Akko formula $F \to G$ ima barem jedan model
- C Akko je $F \to \neg G$ kontradikcija
- D Akko je G deduktivna posljedica od F

7 (T) Formalni sustavi za prikazivanje znanja razlikuju se po svojim ontološkim i epistemološkim pretpostavkama. **U čemu se očituje razlika u epistemološkim pretpostavkama između PL i FOL?**

- A FOL pretpostavlja postojanje objekata i relacija među njima
- B Dokazivanje logičke posljedice je odlučivo u PL, ali neodlučivo u FOL
- C Nema razlike u epistemološkim pretpostavkama između PL i FOL
- D PL ne koristi kvantifikatore, ali ima konačan broj interpretacija

8 (P) Neka $R(x)$ označava "x je riječ", $U(x)$ označava "x je ugodan" te $I(x)$ označava "x je istinit". Koja od sljedećih formula predikatne logike predstavlja ispravnu formalizaciju rečenice *"Istinite riječi ne moraju biti ugodne"*?

- A $\exists x\Big((R(x) \land I(x)) \to \neg U(x)\Big)$
- B $\exists x\Big(R(x) \land U(x) \land \neg I(x)\Big)$
- C $\neg\forall x\Big((R(x) \land I(x)) \to U(x)\Big)$
- D $\forall x\Big((R(x) \land \neg I(x)) \to \neg U(x)\Big)$

3. Logičko programiranje i ekspertni sustavi (2 pitanja)

9 (R) Baza znanja ekspertnog sustava sadrži sljedeća pravila:

(1) AKO $(A = a_2) \land (D = d_2)$ ONDA $C = c_1$
(2) AKO $(F = f_3) \lor (B = b_3)$ ONDA $C = c_2$
(3) AKO $(E = e_1) \lor (B = b_1)$ ONDA $(A = a_1) \land (D = d_2)$
(4) AKO $F = f_1$ ONDA $D = d_1$
(5) AKO $F = f_2$ ONDA $E = e_2$
(6) AKO $(B = b_3) \lor (D = d_1)$ ONDA $(E = e_1) \land (A = a_2)$

Sustav koristimo za izvođenje vrijednosti varijable D ulančavanjem unazad. Prednost imaju pravila s manjim rednim brojem. Pravila koja su jednom palila više ne mogu paliti. Jednom izvedeni međuciljevi brišu se sa stoga, neovisno o njihovoj poziciji na stogu. Na možebitne upite od strane sustava, korisnik odgovara sa $B = b_3$ i $F = f_1$. **Što radi ekspertni sustav pri izvođenju vrijednosti varijable D?**

- A Završava s tri činjenice u radnoj memoriji
- B Pali tri pravila i izvodi $D = d_1$
- C Odbacuje pravilo 3 te kasnije pali pravilo 5
- D Izvodi $B = b_3$ te kasnije $A = a_2$

10 (T) Programski jezik Prolog omogućava uporabu negacije. Međutim, za razliku od FOL, uporaba negacije u Prologu je ograničena. **Kako je ograničena uporaba negacije u Prologu?**

- A Negacija se ne može koristiti u rekurzivnim definicijama pravila
- B Negirati se može najviše jedan atom na desnoj strani pravila
- C Hornove klauzule uopće ne dopuštaju negaciju atoma
- D Negirati se mogu samo atomi u tijelu pravila (u antecedentu implikacije)

4. Modeliranje neizvjesnosti (2 pitanja)

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

[A] 0.897 [B] 0.853 [C] 0.714 [D] 0.785

12 (T) Teorija vjerojatnosti i neizrazita logika na različit način modeliraju neizvjesnost znanja. Razmotrite sljedeću situaciju. Jako ste žedni i naišli ste na dvije boce, A i B. Na prvoj boci stoji oznaka $P(A = otrov) = 0.1$, a na drugoj $\mu_{otrov}(B) = 0.1$. Pretpostavite da je otrov smrtonosan samo ako ga se konzumira u punoj koncentraciji. Očito, u ovom slučaju vrijedi $P(A) = \mu(B)$, no boce A i B različitog su sadržaja. **Sadržaj koje boce biste popili, i zašto?**

[A] B, jer boca sadrži tek blagu količinu otrova
[B] B, jer samo jedna od 10 takvih boca sadrži otrov
[C] A, jer samo jedna od 10 takvih boca sadrži otrov
[D] A, jer boca sadrži tek blagu količinu otrova

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 $(1, (0, 0))$. 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] -0.109142 [C] -0.024799 [D] -0.192488

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	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 $\mathbf{x}$ 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 x svidio?**

[A] 0.694 [B] 0.856 [C] 0.799 [D] 0.431

15 (P) Algoritam ID3 koristi kriterij informacijske dobiti za odabir varijable u svakom čvoru stabla odluke. Razmotrite klasifikaciju skupa primjera D u tri klase, $y \in \{1, 2, 3\}$. Kandidati za korijen stabla su varijable x_1 i x_2. Varijabla x_1 ima moguće vrijednosti a i b, a varijabla x_2 ima moguće vrijednosti c i d. Za varijablu x_1, entropija podskupova

$D_{x_1=a}$ i $D_{x_1=b}$ je minimalna moguća. Za varijablu x_2, u skupu D postoje tri primjera sa $x_2 = c$ te šest primjera sa $x_2 = d$. Entropija podskupova $D_{x_2=c}$ i $D_{x_2=d}$ je maksimalna moguća. **Koliko je informacijska dobit varijable x_1 veća od informacijske dobiti varijable x_2?**

A $\frac{2}{3}\log_2\frac{1}{3}$ B $\log_2\frac{1}{3}$ C $\frac{1}{3}\log_2 3$ D $\log_2 3$

16 (T) Učenje težina umjetne neuronske mreže na odabranom aproksimacijskom problemu formuliramo kao optimizacijski problem koji potom rješavamo nekim optimizacijskim algoritmom. Pri tome učimo aproksimirati ponašanje nekog sustava za koji nemamo analitički opis, ali smo uspjeli mjerenjima prikupiti 1000 primjera. Naučenu mrežu potom želimo koristiti za predviđanje rada sustava. **Što od sljedećega vrijedi?**

A Zahvaljujući teoremu "No free lunch" znamo da su metode koje se temelje na uporabi derivacija (poput algoritma propagacije pogreške unatrag) najbolji optimizacijski algoritmi za ovaj problem

B Najbolje je učenje provoditi što duže možemo, kako bi pogreška nad svim primjerima kojima raspolažemo pala na najmanju moguću vrijednost

C Prilikom učenja težina ne smijemo koristiti sve primjere za učenje, već dio primjera trebamo koristiti za odluku u kojem trenutku treba prekinuti učenje

D Učenje možemo provoditi postupkom propagacije pogreške unatrag i to samo unatražnim prolazima; unaprijedne prolaze možemo preskočiti

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: $a_1 = \textit{lijevo}$, $a_2 = \textit{desno}$, $a_3 = \textit{sagni-se-i-pokupi-bocu}$. 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 $q(3, a_2)$?**

A 4.25 B 8 C 3 D 2.25

6. Prirodom inspirirani optimizacijski algoritmi (3 pitanja)

18 (R) Uporabom mravlje kolonije traži se ciklus kroz graf. Poznati su sljedeći podatci: $\tau_{1,2} = 2$, $\tau_{1,4} = 0.5$, $\tau_{1,7} = \frac{1}{\sqrt{3}}$, $\tau_{2,4} = 0.5$, $\tau_{2,7} = \frac{1}{2}$, $\tau_{3,5} = 1$, $\tau_{3,6} = \sqrt{3}$, $\tau_{3,7} = \frac{1}{3}$, $\tau_{4,5} = 3$, $\tau_{4,6} = 10\sqrt{2}$, $\tau_{5,6} = 10$, $\tau_{6,7} = 2$. $\eta_{1,2} = \frac{1}{\sqrt{2}}$, $\eta_{1,4} = 2$, $\eta_{1,7} = 3$, $\eta_{2,4} = 0.5$, $\eta_{2,7} = 2\sqrt{2}$, $\eta_{3,5} = \sqrt{2}$, $\eta_{3,6} = 1$, $\eta_{3,7} = 3\sqrt{3}$, $\eta_{4,5} = 3\sqrt{3}$, $\eta_{4,6} = 0.1$, $\eta_{5,6} = \frac{1}{2\sqrt{5}}$, $\eta_{6,7} = 0.5$. Također, sve dane vrijednosti su simetrične, tj. $\tau_{i,j} = \tau_{j,i}$ i $\eta_{i,j} = \eta_{j,i}$. Dodatno, $\alpha = 2$ i $\beta = 2$. Prvi mrav kreće iz čvora 1. Kada mrav treba donijeti vjerojatnosnu odluku, pretpostavite da će ishod slučajnog odabira odgovarati najvjerojatnijem. Ako iz nekog čvora mrav ne može dalje, konstrukcija ciklusa se prekida. Uz te pretpostavke odredite ciklus koji će taj mrav konstruirati. **Slijed od koja tri čvora je dio tog ciklusa?**

A 1, 7, 2 B Mrav neće uspjeti konstruirati ciklus C 5, 4, 2 D 7, 2, 4

19 (P) Genetskim algoritmom tražimo maksimum funkcije $f(x, y, z)$, koristeći binarnu reprezentaciju rješenja. Vrijednost svake od varijabli pretražuje se u intervalu $[10, 90]$, pri čemu je potrebno osigurati da se to pretraživanje provodi barem s preciznošću 0.001. **Od koliko se *minimalno* bitova treba sastojati kromosom?**

A 48 B 49 C 16 D 51

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

A Algoritam bi generirao rješenja koja su neispravna (izvan intervala pretrage)

B Porasla bi 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

31. kolovoza 2023.

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) Poželjno svojstvo aloritma pretraživanja jest da je optimalan. Međutim, algoritmi "najbolji prvi" i "uspon na vrh" nisu optimalni. Oba su ova algoritma "pohlepna", ali je algoritam "najbolji prvi" manje pohlepan od algoritma "uspon na vrh", pa u praksi postoji veća vjerojatnost da će algoritam "najboji prvi" pronaći optimalan put. Skicirajte prostor stanja $S = \{a, b, c, d, e, f\}$ s prijelazima $succ(a) = \{b, c\}$, $succ(b) = \{d, e\}$, $succ(c) = succ(d) = succ(e) = \{f\}$. Cijene svih prijelaza su jednake i iznose 1. Heuristika neka je $h(b) = 1$, $h(d) = h(e) = 3$ i $h(f) = 0$. Stanje a je početno stanje, a stanje f je ciljno stanje. **Za koju vrijednost heuristike za stanje c algoritam "uspona na vrh" *ne* pronalazi optimalno rješenje, ali ga algoritam "najbolji prvi" pronalazi?**

A $h(c) = 5$ B $h(c) = 2$ C $h(c) = 4$ D $h(c) = 0$

2 (T) Ako je za neki problem moguće konstruirati optimističnu heurističku funkciju, usmjereno pretraživanje općenito će biti bolje od slijepog pretraživanja. **Zašto je usmjereno pretraživanje s optimističnom heuristikom općenito (u prosjeku nad svim problemima) bolje od slijepog pretraživanja?**

A Brže nalazi optimalno rješenje
B Ne može zaglaviti u beskonačnoj petlji
C Ima manju apriornu vremensku složenost
D Nalazi kraći put do rješenja

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

A Algoritam ne dostiže peti korak

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

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

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

4 (P) Alfa-beta podrezivanje može značajno smanjiti broj generiranih čvorova kod algoritma minimax. Ušteda ovisi o redoslijedu kojim se generiraju čvorovi. Optimalan redoslijed je onaj koji djecu čvora MAX generira silazno po minimax vrijednostima, a djecu čvora MIN uzlazno po minimax vrijednostima. Razmotite igru s dva moguća poteza u svakome stanju. Stablo se pretražuje do dubine 3. Vrijednosti heuristike igrača MAX za stanja u listovima stabla igre, kada su čvorovi-djeca generirani nasumičnim redoslijedom, su sljedeća (slijeva nadesno): $0, 3, -3, 4, 4, -1, 0, 2$. Na potezu je igrač MAX. Odredite broj čvorova koji će biti uklonjeni podrezivanjem alfa-beta. Zatim ponovite izračun, ali s optimalnim redoslijedom generiranja čvorova. **Koliko više čvorova će biti podrezano s optimalnim redoslijedom u odnosu na slučajni redoslijed generiranja čvorova?**

A 4 B 3 C 1 D 2

2. Prikazivanje znanja i automatsko zaključivanje (4 pitanja)

5 (R) Zadane su sljedeće tvrdnje: *Jednorog je besmrtan (B), ako je mitsko biće (A). Ako je jednorog besmrtan ili je obična smrtna životinja (C), onda je i rogat (D). Čim je jednorog rogat, onda je i magičan (E). No, ako nije besmrtan, onda je obična smrtna životinja.* **Koja se od sljedećih tvrdnji *ne* može dokazati rezolucijom opovrgavanjem?**

A *Jednorog je obična smrtna životinja ili magičan.*

B *Jednorog je obična smrtna životinja ili rogat.*

C *Ako je jednorog mitsko biće, onda je rogat.*

D *Ako je jednorog obična smrtna životinja, onda nije besmrtan.*

6 (T) Pravila zaključivanja u idealnom su slučaju i ispravna i potpuna. **Kada je pravilo zaključivanja $F \vDash G$ ispravno?**

A Akko je svaki model od F ujedno i model od G

B Akko formula $F \rightarrow G$ ima barem jedan model

C Akko je G deduktivna posljedica od F

D Akko je $F \rightarrow \neg G$ kontradikcija

7 (P) Neka $R(x)$ označava "x je riječ", $U(x)$ označava "x je ugodan" te $I(x)$ označava "x je istinit". Koja od sljedećih formula predikatne logike predstavlja ispravnu formalizaciju rečenice *"Istinite riječi ne moraju biti ugodne"*?

A $\forall x \neg \Big((R(x) \wedge I(x)) \rightarrow U(x) \Big)$

B $\exists x \neg \Big((R(x) \wedge I(x)) \rightarrow U(x) \Big)$

C $\exists x \neg \Big(R(x) \wedge I(x) \wedge U(x) \Big)$

D $\exists x \Big(R(x) \wedge U(x) \wedge \neg I(x) \Big)$

8 (T) Formalni sustavi za prikazivanje znanja razlikuju se po svojim ontološkim i epistemološkim pretpostavkama. **U čemu se očituje razlika u epistemološkim pretpostavkama između PL i FOL?**

A Nema razlike u epistemološkim pretpostavkama između PL i FOL

B Dokazivanje logičke posljedice je odlučivo u PL, ali neodlučivo u FOL

C PL ne koristi kvantifikatore, ali ima konačan broj interpretacija

D FOL pretpostavlja postojanje objekata i relacija među njima

3. Logičko programiranje i ekspertni sustavi (2 pitanja)

9 (R) Baza znanja ekspertnog sustava sadrži sljedeća pravila:

(1) AKO $(A = a_2) \wedge (D = d_2)$ ONDA $C = c_1$

(2) AKO $(F = f_3) \vee (B = b_3)$ ONDA $C = c_2$

(3) AKO $(E = e_1) \vee (B = b_1)$ ONDA $(A = a_1) \wedge (D = d_2)$

(4) AKO $F = f_1$ ONDA $D = d_1$

(5) AKO $F = f_2$ ONDA $E = e_2$

(6) AKO $(B = b_3) \vee (D = d_1)$ ONDA $(E = e_1) \wedge (A = a_2)$

Sustav koristimo za izvođenje vrijednosti varijable D ulančavanjem unazad. Prednost imaju pravila s manjim rednim brojem. Pravila koja su jednom palila više ne mogu paliti. Jednom izvedeni međuciljevi brišu se sa stoga, neovisno o njihovoj poziciji na stogu. Na možebitne upite od strane sustava, korisnik odgovara sa $B = b_3$ i $F = f_1$. **Što radi ekspertni sustav pri izvođenju vrijednosti varijable D?**

A Izvodi $E = e_2$ te kasnije $C = c_2$

B Odbacuje pravilo 3 te kasnije pali pravilo 5

C Pali dva pravila i izvodi $D = d_2$

D Završava s tri činjenice u radnoj memoriji

10 (T) Programski jezik Prolog omogućava uporabu negacije. Međutim, za razliku od FOL, uporaba negacije u Prologu je ograničena. **Kako je ograničena uporaba negacije u Prologu?**

A Negacija se ne može koristiti u rekurzivnim definicijama pravila

B Negirati se mogu samo atomi u tijelu pravila (u antecedentu implikacije)

C Hornove klauzule uopće ne dopuštaju negaciju atoma

D Negirati se može najviše jedan atom na desnoj strani pravila

4. Modeliranje neizvjesnosti (2 pitanja)

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

A 0.897 B 0.785 C 0.853 D 0.714

12 (T) Teorija vjerojatnosti i neizrazita logika na različit način modeliraju neizvjesnost znanja. Razmotrite sljedeću situaciju. Jako ste žedni i naišli ste na dvije boce, A i B. Na prvoj boci stoji oznaka $P(A = otrov) = 0.1$, a na drugoj $\mu_{otrov}(B) = 0.1$. Pretpostavite da je otrov smrtonosan samo ako ga se konzumira u punoj koncentraciji. Očito, u ovom slučaju vrijedi $P(A) = \mu(B)$, no boce A i B različitog su sadržaja. **Sadržaj koje boce biste popili, i zašto?**

A B, jer boca sadrži tek blagu količinu otrova
C A, jer samo jedna od 10 takvih boca sadrži otrov
B A, jer boca sadrži tek blagu količinu otrova
D B, jer samo jedna od 10 takvih boca sadrži otrov

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 $(1, (0, 0))$. 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 -0.192488 C -0.109142 D -0.024799

14 (T) Učenje težina umjetne neuronske mreže na odabranom aproksimacijskom problemu formuliramo kao optimizacijski problem koji potom rješavamo nekim optimizacijskim algoritmom. Pri tome učimo aproksimirati ponašanje nekog sustava za koji nemamo analitički opis, ali smo uspjeli mjerenjima prikupiti 1000 primjera. Naučenu mrežu potom želimo koristiti za predviđanje rada sustava. **Što od sljedećega vrijedi?**

A Prilikom učenja težina ne smijemo koristiti sve primjere za učenje, već dio primjera trebamo koristiti za odluku u kojem trenutku treba prekinuti učenje

B Učenje možemo provoditi postupkom propagacije pogreške unatrag i to samo unatražnim prolazima; unaprijedne prolaze možemo preskočiti

C Najbolje je učenje provoditi što duže možemo, kako bi pogreška nad svim primjerima kojima raspolažemo pala na najmanju moguću vrijednost

D Zahvaljujući teoremu "No free lunch" znamo da su metode koje se temelje na uporabi derivacija (poput algoritma propagacije pogreške unatrag) najbolji optimizacijski algoritmi za ovaj problem

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

i	Evaluacija	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: $\mathbf{x}$ = (lijena, interpreter, hibridna, 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 0.856 B 0.694 C 0.431 D 0.799

16 (P) Algoritam ID3 koristi kriterij informacijske dobiti za odabir varijable u svakom čvoru stabla odluke. Razmotrite klasifikaciju skupa primjera D u tri klase, $y \in \{1, 2, 3\}$. Kandidati za korijen stabla su varijable x_1 i x_2. Varijabla x_1 ima moguće vrijednosti a i b, a varijabla x_2 ima moguće vrijednosti c i d. Za varijablu x_1, entropija podskupova $D_{x_1=a}$ i $D_{x_1=b}$ je minimalna moguća. Za varijablu x_2, u skupu D postoje tri primjera sa $x_2 = c$ te šest primjera sa $x_2 = d$. Entropija podskupova $D_{x_2=c}$ i $D_{x_2=d}$ je maksimalna moguća. **Koliko je informacijska dobit varijable x_1 veća od informacijske dobiti varijable x_2?**

A $\frac{1}{3}\log_2 3$ B $\log_2 3$ C $\log_2 \frac{1}{3}$ D $\frac{2}{3}\log_2 \frac{1}{3}$

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: $a_1 =$ *lijevo*, $a_2 =$ *desno*, $a_3 =$ *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 $q(3, a_2)$?**

A 8 B 3 C 2.25 D 4.25

6. Prirodom inspirirani optimizacijski algoritmi (3 pitanja)

18 (R) Uporabom mravlje kolonije traži se ciklus kroz graf. Poznati su sljedeći podatci: $\tau_{1,2} = 2$, $\tau_{1,4} = 0.5$, $\tau_{1,7} = \frac{1}{\sqrt{3}}$, $\tau_{2,4} = 0.5$, $\tau_{2,7} = \frac{1}{2}$, $\tau_{3,5} = 1$, $\tau_{3,6} = \sqrt{3}$, $\tau_{3,7} = \frac{1}{3}$, $\tau_{4,5} = 3$, $\tau_{4,6} = 10\sqrt{2}$, $\tau_{5,6} = 10$, $\tau_{6,7} = 2$. $\eta_{1,2} = \frac{1}{\sqrt{2}}$, $\eta_{1,4} = 2$, $\eta_{1,7} = 3$, $\eta_{2,4} = 0.5$, $\eta_{2,7} = 2\sqrt{2}$, $\eta_{3,5} = \sqrt{2}$, $\eta_{3,6} = 1$, $\eta_{3,7} = 3\sqrt{3}$, $\eta_{4,5} = 3\sqrt{3}$, $\eta_{4,6} = 0.1$, $\eta_{5,6} = \frac{1}{2\sqrt{5}}$, $\eta_{6,7} = 0.5$. Također, sve dane vrijednosti su simetrične, tj. $\tau_{i,j} = \tau_{j,i}$ i $\eta_{i,j} = \eta_{j,i}$. Dodatno, $\alpha = 2$ i $\beta = 2$. Prvi mrav kreće iz čvora 1. Kada mrav treba donijeti vjerojatnosnu odluku, pretpostavite da će ishod slučajnog odabira odgovarati najvjerojatnijem. Ako iz nekog čvora mrav ne može dalje, konstrukcija ciklusa se prekida. Uz te pretpostavke odredite ciklus koji će taj mrav konstruirati. **Slijed od koja tri čvora je dio tog ciklusa?**

A 4, 2, 1 B 1, 7, 2 C Mrav neće uspjeti konstruirati ciklus D 3, 5, 4

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

A Porasla bi vjerojatnost da pretraga zapne u lokalnome optimumu

B Porasla bi prosječna vrijednost funkcije dobrote u populaciji

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

20 (P) Genetskim algoritmom tražimo maksimum funkcije $f(x, y, z)$, koristeći binarnu reprezentaciju rješenja. Vrijednost svake od varijabli pretražuje se u intervalu $[-100, 100]$, pri čemu je potrebno osigurati da se to pretraživanje provodi barem s preciznošću 0.001. **Od koliko se *minimalno* bitova treba sastojati kromosom?**

A 53 B 17 C 54 D 51

 31. kolovoza 2023.

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

A Algoritam ne dostiže peti korak

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

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

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

2 (T) Ako je za neki problem moguće konstruirati optimističnu heurističku funkciju, usmjereno pretraživanje općenito će biti bolje od slijepog pretraživanja. **Zašto je usmjereno pretraživanje s optimističnom heuristikom općenito (u prosjeku nad svim problemima) bolje od slijepog pretraživanja?**

A Ne može zaglaviti u beskonačnoj petlji

C Nalazi kraći put do rješenja

B Brže nalazi optimalno rješenje

D Ima manju apriornu vremensku složenost

3 (P) Poželjno svojstvo aloritma pretraživanja jest da je optimalan. Međutim, algoritmi "najbolji prvi" i "uspon na vrh" nisu optimalni. Oba su ova algoritma "pohlepna", ali je algoritam "najbolji prvi" manje pohlepan od algoritma "uspon na vrh", pa u praksi postoji veća vjerojatnost da će algoritam "najboji prvi" pronaći optimalan put. Skicirajte prostor stanja $S = \{a, b, c, d, e, f\}$ s prijelazima $succ(a) = \{b, c\}$, $succ(b) = \{d, e\}$, $succ(c) = succ(d) = succ(e) = \{f\}$. Cijene svih prijelaza su jednake i iznose 1. Heuristika neka je $h(c) = 2$, $h(d) = h(e) = 3$ i $h(f) = 0$. Stanje a je početno stanje, a stanje f je ciljno stanje. **Za koju vrijednost heuristike za stanje b algoritam "uspona na vrh" *ne* pronalazi optimalno rješenje, ali ga algoritam "najbolji prvi" pronalazi?**

A $h(b) = 3$ B $h(b) = 4$ C $h(b) = 5$ D $h(b) = 1$

4 (P) Alfa-beta podrezivanje može značajno smanjiti broj generiranih čvorova kod algoritma minimax. Ušteda ovisi o redoslijedu kojim se generiraju čvorovi. Optimalan redoslijed je onaj koji djecu čvora MAX generira silazno po minimax vrijednostima, a djecu čvora MIN uzlazno po minimax vrijednostima. Razmotite igru s dva moguća poteza u svakome stanju. Stablo se pretražuje do dubine 3. Vrijednosti heuristike igrača MAX za stanja u listovima stabla igre, kada su čvorovi-djeca generirani nasumičnim redoslijedom, su sljedeća (slijeva nadesno): $0, 3, -3, 4, 4, -1, 0, 2$. Na potezu je igrač MAX. Odredite broj čvorova koji će biti uklonjeni podrezivanjem alfa-beta. Zatim ponovite izračun, ali s optimalnim redoslijedom generiranja čvorova. **Koliko više čvorova će biti podrezano s optimalnim redoslijedom u odnosu na slučajni redoslijed generiranja čvorova?**

A 4 B 1 C 2 D 3

2. Prikazivanje znanja i automatsko zaključivanje (4 pitanja)

5 (T) Formalni sustavi za prikazivanje znanja razlikuju se po svojim ontološkim i epistemološkim pretpostavkama. **U čemu se očituje razlika u epistemološkim pretpostavkama između PL i FOL?**

- A FOL pretpostavlja postojanje objekata i relacija među njima
- B PL ne koristi kvantifikatore, ali ima konačan broj interpretacija
- C Dokazivanje logičke posljedice je odlučivo u PL, ali neodlučivo u FOL
- D Nema razlike u epistemološkim pretpostavkama između PL i FOL

6 (T) Pravila zaključivanja u idealnom su slučaju i ispravna i potpuna. **Kada je pravilo zaključivanja $F \vDash G$ ispravno?**

- A Akko je svaki model od F ujedno i model od G
- B Akko je G deduktivna posljedica od F
- C Akko formula $F \to G$ ima barem jedan model
- D Akko je $F \to \neg G$ kontradikcija

7 (P) Neka $R(x)$ označava "x je riječ", $U(x)$ označava "x je ugodan" te $I(x)$ označava "x je istinit". Koja od sljedećih formula predikatne logike predstavlja ispravnu formalizaciju rečenice *"Istinite riječi ne moraju biti ugodne"*?

- A $\neg\forall x\Big((R(x) \land U(x)) \to I(x)\Big)$
- B $\exists x\Big((R(x) \land I(x)) \to \neg U(x)\Big)$
- C $\exists x\Big(R(x) \land U(x) \land \neg I(x)\Big)$
- D $\exists x\Big(R(x) \land I(x) \land \neg U(x)\Big)$

8 (R) Zadane su sljedeće tvrdnje: *Jednorog je besmrtan (B), ako je mitsko biće (A). Ako je jednorog besmrtan ili je obična smrtna životinja (C), onda je i rogat (D). Čim je jednorog rogat, onda je i magičan (E). No, ako nije besmrtan, onda je obična smrtna životinja.* **Koja se od sljedećih tvrdnji *ne* može dokazati rezolucijom opovrgavanjem?**

- A *Jednorog nije mitsko biće, ako nije rogat.*
- B *Jednorog je obična smrtna životinja ili rogat.*
- C *Ako je jednorog obična smrtna životinja, onda nije besmrtan.*
- D *Ako jednorog nije obična smrtna životinja, onda je besmrtan.*

3. Logičko programiranje i ekspertni sustavi (2 pitanja)

9 (R) Baza znanja ekspertnog sustava sadrži sljedeća pravila:

(1) AKO $(A = a_2) \land (D = d_2)$ ONDA $C = c_1$
(2) AKO $(F = f_3) \lor (B = b_3)$ ONDA $C = c_2$
(3) AKO $(E = e_1) \lor (B = b_1)$ ONDA $(A = a_1) \land (D = d_2)$
(4) AKO $F = f_1$ ONDA $D = d_1$
(5) AKO $F = f_2$ ONDA $E = e_2$
(6) AKO $(B = b_3) \lor (D = d_1)$ ONDA $(E = e_1) \land (A = a_2)$

Sustav koristimo za izvođenje vrijednosti varijable D ulančavanjem unazad. Prednost imaju pravila s manjim rednim brojem. Pravila koja su jednom palila više ne mogu paliti. Jednom izvedeni međuciljevi brišu se sa stoga, neovisno o njihovoj poziciji na stogu. Na moguće upite od strane sustava, korisnik odgovara sa $B = b_3$ i $F = f_1$. **Što radi ekspertni sustav pri izvođenju vrijednosti varijable D?**

- A Izvodi $E = e_2$ te kasnije $C = c_2$
- B Završava s tri činjenice u radnoj memoriji
- C Pali tri pravila i izvodi $D = d_1$
- D Izvodi $B = b_3$ te kasnije $A = a_2$

10 (T) Programski jezik Prolog omogućava uporabu negacije. Međutim, za razliku od FOL, uporaba negacije u Prologu je ograničena. **Kako je ograničena uporaba negacije u Prologu?**

- A Hornove klauzule uopće ne dopuštaju negaciju atoma
- B Negirati se može najviše jedan atom na desnoj strani pravila
- C Negacija se ne može koristiti u rekurzivnim definicijama pravila
- D Negirati se mogu samo atomi u tijelu pravila (u antecedentu implikacije)

4. Modeliranje neizvjesnosti (2 pitanja)

11 (T) Teorija vjerojatnosti i neizrazita logika na različit način modeliraju neizvjesnost znanja. Razmotrite sljedeću situaciju. Jako ste žedni i naišli ste na dvije boce, A i B. Na prvoj boci stoji oznaka $P(A = otrov) = 0.1$, a na drugoj $\mu_{otrov}(B) = 0.1$. Pretpostavite da je otrov smrtonosan samo ako ga se konzumira u punoj koncentraciji. Očito, u ovom slučaju vrijedi $P(A) = \mu(B)$, no boce A i B različitog su sadržaja. **Sadržaj koje boce biste popili, i zašto?**

- A A, jer boca sadrži tek blagu količinu otrova
- B A, jer samo jedna od 10 takvih boca sadrži otrov
- C B, jer samo jedna od 10 takvih boca sadrži otrov
- D B, jer boca sadrži tek blagu količinu otrova

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

A 0.897 B 0.853 C 0.714 D 0.785

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

13 (T) Učenje težina umjetne neuronske mreže na odabranom aproksimacijskom problemu formuliramo kao optimizacijski problem koji potom rješavamo nekim optimizacijskim algoritmom. Pri tome učimo aproksimirati ponašanje nekog sustava za koji nemamo analitički opis, ali smo uspjeli mjerenjima prikupiti 1000 primjera. Naučenu mrežu potom želimo koristiti za predviđanje rada sustava. **Što od sljedećega vrijedi?**

- A Zahvaljujući teoremu "No free lunch" znamo da su metode koje se temelje na uporabi derivacija (poput algoritma propagacije pogreške unatrag) najbolji optimizacijski algoritmi za ovaj problem
- B Učenje možemo provoditi postupkom propagacije pogreške unatrag i to samo unatražnim prolazima; unaprijedne prolaze možemo preskočiti
- C Prilikom učenja težina ne smijemo koristiti sve primjere za učenje, već dio primjera trebamo koristiti za odluku u kojem trenutku treba prekinuti učenje
- D Najbolje je učenje provoditi što duže možemo, kako bi pogreška nad svim primjerima kojima raspolažemo pala na najmanju moguću vrijednost

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 $(1, (1, 1))$. 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 −0.192488 B −0.058547 C −0.109142 D −0.024799

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

i	Evaluacija	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: $\mathbf{x} = (\text{striktna}, \text{interpreter}, \text{hibridna}, \text{dinamička})$. 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 0.799 B 0.431 C 0.856 D 0.694

16 (P) Algoritam ID3 koristi kriterij informacijske dobiti za odabir varijable u svakom čvoru stabla odluke. Razmotrite klasifikaciju skupa primjera D u tri klase, $y \in \{1, 2, 3\}$. Kandidati za korijen stabla su varijable x_1 i x_2. Varijabla x_1 ima moguće vrijednosti a i b, a varijabla x_2 ima moguće vrijednosti c i d. Za varijablu x_1, entropija podskupova $D_{x_1=a}$ i $D_{x_1=b}$ je minimalna moguća. Za varijablu x_2, u skupu D postoje tri primjera sa $x_2 = c$ te šest primjera sa $x_2 = d$. Entropija podskupova $D_{x_2=c}$ i $D_{x_2=d}$ je maksimalna moguća. **Koliko je informacijska dobit varijable x_1 veća od informacijske dobiti varijable x_2?**

A $\frac{1}{3}\log_2 3$ B $\frac{2}{3}\log_2 \frac{1}{3}$ C $\log_2 3$ D $\log_2 \frac{1}{3}$

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: $a_1 = \textit{lijevo}$, $a_2 = \textit{desno}$, $a_3 = \textit{sagni-se-i-pokupi-bocu}$. 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 $q(3, a_2)$?**

A 3 B 2.25 C 8 D 4.25

6. Prirodom inspirirani optimizacijski algoritmi (3 pitanja)

18 (R) Uporabom mravlje kolonije traži se ciklus kroz graf. Poznati su sljedeći podatci: $\tau_{1,3} = 0.5$, $\tau_{1,6} = \frac{1}{\sqrt{3}}$, $\tau_{1,7} = 2$, $\tau_{2,4} = 1$, $\tau_{2,5} = \sqrt{3}$, $\tau_{2,6} = \frac{1}{3}$, $\tau_{3,4} = 3$, $\tau_{3,5} = 10\sqrt{2}$, $\tau_{3,7} = 0.5$, $\tau_{4,5} = 10$, $\tau_{5,6} = 2$, $\tau_{6,7} = \frac{1}{2}$. $\eta_{1,3} = 2$, $\eta_{1,6} = 3$, $\eta_{1,7} = \frac{1}{\sqrt{2}}$, $\eta_{2,4} = \sqrt{2}$, $\eta_{2,5} = 1$, $\eta_{2,6} = 3\sqrt{3}$, $\eta_{3,4} = 3\sqrt{3}$, $\eta_{3,5} = 0.1$, $\eta_{3,7} = 0.5$, $\eta_{4,5} = \frac{1}{2\sqrt{5}}$, $\eta_{5,6} = 0.5$, $\eta_{6,7} = 2\sqrt{2}$. Također, sve dane vrijednosti su simetrične, tj. $\tau_{i,j} = \tau_{j,i}$ i $\eta_{i,j} = \eta_{j,i}$. Dodatno, $\alpha = 2$ i $\beta = 2$. Prvi mrav kreće iz čvora 1. Kada mrav treba donijeti vjerojatnosnu odluku, pretpostavite da će ishod slučajnog odabira odgovarati najvjerojatnijem. Ako iz nekog čvora mrav ne može dalje, konstrukcija ciklusa se prekida. Uz te pretpostavke odredite ciklus koji će taj mrav konstruirati. **Slijed od koja tri čvora je dio tog ciklusa?**

A 4, 3, 7 B 6, 7, 3 C 6, 5, 2 D Mrav neće uspjeti konstruirati ciklus

19 (P) Genetskim algoritmom tražimo maksimum funkcije $f(x, y, z)$, koristeći binarnu reprezentaciju rješenja. Vrijednost svake od varijabli pretražuje se u intervalu $[-10, 15]$, pri čemu je potrebno osigurati da se to pretraživanje provodi barem s preciznošću 0.01. **Od koliko se *minimalno* bitova treba sastojati kromosom?**

A 36 B 11 C 33 D 34

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

A Algoritam bi mogao vratiti rješenje koje je lošije od nekog ranije generiranog rješenja

B Porasla bi vjerojatnost da pretraga zapne u lokalnome optimumu

C Algoritam bi generirao rješenja koja su neispravna (izvan intervala pretrage)

D Porasla bi 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) Ako je za neki problem moguće konstruirati optimističnu heurističku funkciju, usmjereno pretraživanje općenito će biti bolje od slijepog pretraživanja. **Zašto je usmjereno pretraživanje s optimističnom heuristikom općenito (u prosjeku nad svim problemima) bolje od slijepog pretraživanja?**

A Brže nalazi optimalno rješenje

B Ima manju apriornu vremensku složenost

C Ne može zaglaviti u beskonačnoj petlji

D Nalazi kraći put do rješenja

2 (P) Alfa-beta podrezivanje može značajno smanjiti broj generiranih čvorova kod algoritma minimax. Ušteda ovisi o redoslijedu kojim se generiraju čvorovi. Optimalan redoslijed je onaj koji djecu čvora MAX generira silazno po minimax vrijednostima, a djecu čvora MIN uzlazno po minimax vrijednostima. Razmotite igru s dva moguća poteza u svakome stanju. Stablo se pretražuje do dubine 3. Vrijednosti heuristike igrača MAX za stanja u listovima stabla igre, kada su čvorovi-djeca generirani nasumičnim redoslijedom, su sljedeća (slijeva nadesno): $0, 3, -3, 4, 4, -1, 0, 2$. Na potezu je igrač MAX. Odredite broj čvorova koji će biti uklonjeni podrezivanjem alfa-beta. Zatim ponovite izračun, ali s optimalnim redoslijedom generiranja čvorova. **Koliko više čvorova će biti podrezano s optimalnim redoslijedom u odnosu na slučajni redoslijed generiranja čvorova?**

A 4 B 2 C 3 D 1

3 (P) Poželjno svojstvo aloritma pretraživanja jest da je optimalan. Međutim, algoritmi "najbolji prvi" i "uspon na vrh" nisu optimalni. Oba su ova algoritma "pohlepna", ali je algoritam "najbolji prvi" manje pohlepan od algoritma "uspon na vrh", pa u praksi postoji veća vjerojatnost da će algoritam "najboji prvi" pronaći optimalan put. Skicirajte prostor stanja $S = \{a, b, c, d, e, f\}$ s prijelazima $succ(a) = \{b, c\}$, $succ(b) = \{d, e\}$, $succ(c) = succ(d) = succ(e) = \{f\}$. Cijene svih prijelaza su jednake i iznose 1. Heuristika neka je $h(b) = 1$, $h(d) = h(e) = 3$ i $h(f) = 0$. Stanje a je početno stanje, a stanje f je ciljno stanje. **Za koju vrijednost heuristike za stanje c algoritam "uspona na vrh" *ne* pronalazi optimalno rješenje, ali ga algoritam "najbolji prvi" pronalazi?**

A $h(c) = 5$ B $h(c) = 4$ C $h(c) = 2$ D $h(c) = 0$

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

A Algoritam ne dostiže peti korak

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

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

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

2. Prikazivanje znanja i automatsko zaključivanje (4 pitanja)

5 (T) Formalni sustavi za prikazivanje znanja razlikuju se po svojim ontološkim i epistemološkim pretpostavkama. **U čemu se očituje razlika u epistemološkim pretpostavkama između PL i FOL?**

- A FOL pretpostavlja postojanje objekata i relacija među njima
- B Nema razlike u epistemološkim pretpostavkama između PL i FOL
- C Dokazivanje logičke posljedice je odlučivo u PL, ali neodlučivo u FOL
- D PL ne koristi kvantifikatore, ali ima konačan broj interpretacija

6 (P) Neka $R(x)$ označava "x je riječ", $U(x)$ označava "x je ugodan" te $I(x)$ označava "x je istinit". Koja od sljedećih formula predikatne logike predstavlja ispravnu formalizaciju rečenice *"Istinite riječi ne moraju biti ugodne"*?

- A $\neg\exists x\Big((R(x) \wedge I(x)) \rightarrow U(x)\Big)$
- C $\exists x\Big(R(x) \wedge U(x) \wedge \neg I(x)\Big)$
- B $\exists x\Big((R(x) \wedge I(x)) \rightarrow \neg U(x)\Big)$
- D $\exists x\Big(R(x) \wedge I(x) \wedge \neg U(x)\Big)$

7 (R) Zadane su sljedeće tvrdnje: *Jednorog je besmrtan (B), ako je mitsko biće (A). Ako je jednorog besmrtan ili je obična smrtna životinja (C), onda je i rogat (D). Čim je jednorog rogat, onda je i magičan (E). No, ako nije besmrtan, onda je obična smrtna životinja.* **Koja se od sljedećih tvrdnji *ne* može dokazati rezolucijom opovrgavanjem?**

- A *Ako je jednorog rogat, onda je jednorog rogat.*
- B *Jednorog je mitsko biće.*
- C *Jednorog je obična smrtna životinja ili magičan.*
- D *Ako je jednorog mitsko biće, onda je rogat.*

8 (T) Pravila zaključivanja u idealnom su slučaju i ispravna i potpuna. **Kada je pravilo zaključivanja $F \vDash G$ ispravno?**

- A Akko je G deduktivna posljedica od F
- C Akko je $F \rightarrow \neg G$ kontradikcija
- B Akko je svaki model od F ujedno i model od G
- D Akko formula $F \rightarrow G$ ima barem jedan model

3. Logičko programiranje i ekspertni sustavi (2 pitanja)

9 (R) Baza znanja ekspertnog sustava sadrži sljedeća pravila:

(1) AKO $(A = a_2) \wedge (D = d_2)$ ONDA $C = c_1$
(2) AKO $(F = f_3) \vee (B = b_3)$ ONDA $C = c_2$
(3) AKO $(E = e_1) \vee (B = b_1)$ ONDA $(A = a_1) \wedge (D = d_2)$
(4) AKO $F = f_1$ ONDA $D = d_1$
(5) AKO $F = f_2$ ONDA $E = e_2$
(6) AKO $(B = b_3) \vee (D = d_1)$ ONDA $(E = e_1) \wedge (A = a_2)$

Sustav koristimo za izvođenje vrijednosti varijable D ulančavanjem unazad. Prednost imaju pravila s manjim rednim brojem. Pravila koja su jednom palila više ne mogu paliti. Jednom izvedeni međuciljevi brišu se sa stoga, neovisno o njihovoj poziciji na stogu. Na možebitne upite od strane sustava, korisnik odgovara sa $B = b_3$ i $F = f_1$. **Što radi ekspertni sustav pri izvođenju vrijednosti varijable D?**

- A Odbacuje pravilo 3 te kasnije pali pravilo 5
- C Završava s tri činjenice u radnoj memoriji
- B Izvodi $E = e_2$ te kasnije $C = c_2$
- D Izvodi $B = b_3$ te kasnije $A = a_2$

10 (T) Programski jezik Prolog omogućava uporabu negacije. Međutim, za razliku od FOL, uporaba negacije u Prologu je ograničena. **Kako je ograničena uporaba negacije u Prologu?**

- A Negirati se može najviše jedan atom na desnoj strani pravila
- B Negacija se ne može koristiti u rekurzivnim definicijama pravila
- C Hornove klauzule uopće ne dopuštaju negaciju atoma
- D Negirati se mogu samo atomi u tijelu pravila (u antecedentu implikacije)

4. Modeliranje neizvjesnosti (2 pitanja)

11 (T) Teorija vjerojatnosti i neizrazita logika na različit način modeliraju neizvjesnost znanja. Razmotrite sljedeću situaciju. Jako ste žedni i naišli ste na dvije boce, A i B. Na prvoj boci stoji oznaka $P(A = otrov) = 0.1$, a na drugoj $\mu_{otrov}(B) = 0.1$. Pretpostavite da je otrov smrtonosan samo ako ga se konzumira u punoj koncentraciji. Očito, u ovom slučaju vrijedi $P(A) = \mu(B)$, no boce A i B različitog su sadržaja. **Sadržaj koje boce biste popili, i zašto?**

A A, jer samo jedna od 10 takvih boca sadrži otrov

B A, jer boca sadrži tek blagu količinu otrova

C B, jer boca sadrži tek blagu količinu otrova

D B, jer samo jedna od 10 takvih boca sadrži otrov

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

A 0.714 B 0.853 C 0.897 D 0.785

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

13 (P) Algoritam ID3 koristi kriterij informacijske dobiti za odabir varijable u svakom čvoru stabla odluke. Razmotrite klasifikaciju skupa primjera D u tri klase, $y \in \{1, 2, 3\}$. Kandidati za korijen stabla su varijable x_1 i x_2. Varijabla x_1 ima moguće vrijednosti a i b, a varijabla x_2 ima moguće vrijednosti c i d. Za varijablu x_1, entropija podskupova $D_{x_1=a}$ i $D_{x_1=b}$ je minimalna moguća. Za varijablu x_2, u skupu D postoje tri primjera sa $x_2 = c$ te šest primjera sa $x_2 = d$. Entropija podskupova $D_{x_2=c}$ i $D_{x_2=d}$ je maksimalna moguća. **Koliko je informacijska dobit varijable x_1 veća od informacijske dobiti varijable x_2?**

A $\log_2 3$ B $\frac{2}{3} \log_2 \frac{1}{3}$ C $\log_2 \frac{1}{3}$ D $\frac{1}{3} \log_2 3$

14 (T) Učenje težina umjetne neuronske mreže na odabranom aproksimacijskom problemu formuliramo kao optimizacijski problem koji potom rješavamo nekim optimizacijskim algoritmom. Pri tome učimo aproksimirati ponašanje nekog sustava za koji nemamo analitički opis, ali smo uspjeli mjerenjima prikupiti 1000 primjera. Naučenu mrežu potom želimo koristiti za predviđanje rada sustava. **Što od sljedećega vrijedi?**

A Najbolje je učenje provoditi što duže možemo, kako bi pogreška nad svim primjerima kojima raspolažemo pala na najmanju moguću vrijednost

B Zahvaljujući teoremu "No free lunch" znamo da su metode koje se temelje na uporabi derivacija (poput algoritma propagacije pogreške unatrag) najbolji optimizacijski algoritmi za ovaj problem

C Učenje možemo provoditi postupkom propagacije pogreške unatrag i to samo unatražnim prolazima; unaprijedne prolaze možemo preskočiti

D Prilikom učenja težina ne smijemo koristiti sve primjere za učenje, već dio primjera trebamo koristiti za odluku u kojem trenutku treba prekinuti učenje

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 $(1, (0, 0))$. 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 -0.109142 B -0.192488 C -0.058547 D -0.024799

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

i	Evaluacija	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 $\mathbf{x}$ sa sljedećim karakteristikama: $\mathbf{x} = (\text{striktna}, \text{interpreter}, \text{hibridna}, \text{dinamička})$. 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 0.856 B 0.799 C 0.431 D 0.694

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: $a_1 = lijevo$, $a_2 = desno$, $a_3 = sagni\text{-}se\text{-}i\text{-}pokupi\text{-}bocu$. 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** $q(3, a_2)$?

A 1.25 B -2.5 C -1 D 2

6. Prirodom inspirirani optimizacijski algoritmi (3 pitanja)

18 (R) Uporabom mravlje kolonije traži se ciklus kroz graf. Poznati su sljedeći podatci: $\tau_{1,2} = \frac{1}{\sqrt{3}}$, $\tau_{1,4} = 2$, $\tau_{1,6} = 0.5$, $\tau_{2,4} = \frac{1}{2}$, $\tau_{2,5} = \frac{1}{3}$, $\tau_{2,7} = 2$, $\tau_{3,5} = 1$, $\tau_{3,6} = 3$, $\tau_{3,7} = 10$, $\tau_{4,6} = 0.5$, $\tau_{5,7} = \sqrt{3}$, $\tau_{6,7} = 10\sqrt{2}$, $\eta_{1,2} = 3$, $\eta_{1,4} = \frac{1}{\sqrt{2}}$, $\eta_{1,6} = 2$, $\eta_{2,4} = 2\sqrt{2}$, $\eta_{2,5} = 3\sqrt{3}$, $\eta_{2,7} = 0.5$, $\eta_{3,5} = \sqrt{2}$, $\eta_{3,6} = 3\sqrt{3}$, $\eta_{3,7} = \frac{1}{2\sqrt{5}}$, $\eta_{4,6} = 0.5$, $\eta_{5,7} = 1$, $\eta_{6,7} = 0.1$. Također, sve dane vrijednosti su simetrične, tj. $\tau_{i,j} = \tau_{j,i}$ i $\eta_{i,j} = \eta_{j,i}$. Dodatno, $\alpha = 2$ i $\beta = 2$. Prvi mrav kreće iz čvora 1. Kada mrav treba donijeti vjerojatnosnu odluku, pretpostavite da će ishod slučajnog odabira odgovarati najvjerojatnijem. Ako iz nekog čvora mrav ne može dalje, konstrukcija ciklusa se prekida. Uz te pretpostavke odredite ciklus koji će taj mrav konstruirati. **Slijed od koja tri čvora je dio tog ciklusa?**

A 6, 4, 2 B 2, 7, 5 C 3, 6, 4 D Mrav neće uspjeti konstruirati ciklus

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

A Porasla bi vjerojatnost da pretraga zapne u lokalnome optimumu

B Porasla bi prosječna vrijednost funkcije dobrote u populaciji

C Algoritam bi mogao vratiti rješenje koje je lošije od nekog ranije generiranog rješenja

D Algoritam bi generirao rješenja koja su neispravna (izvan intervala pretrage)

20 (P) Genetskim algoritmom tražimo maksimum funkcije $f(x, y, z)$, koristeći binarnu reprezentaciju rješenja. Vrijednost svake od varijabli pretražuje se u intervalu $[-20, 10]$, pri čemu je potrebno osigurati da se to pretraživanje provodi barem s preciznošću 0.05. **Od koliko se *minimalno* bitova treba sastojati kromosom?**

A 27 B 30 C 28 D 9

Grupa	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	D	B	C	D	A	A	D	B	B	D	D	D	B	D	D	C	A	B	C	A
B	B	D	A	C	B	D	C	B	D	A	A	D	D	C	A	A	A	C	B	D
C	C	D	A	D	D	B	C	C	B	B	D	B	B	B	B	C	B	D	C	D
D	A	D	D	A	D	A	A	D	A	A	C	A	B	A	B	D	C	A	A	C
E	B	A	A	C	D	A	C	C	D	D	A	A	D	D	D	C	A	C	D	D
F	B	A	B	A	D	A	B	A	C	B	A	A	B	A	D	B	D	A	A	C
G	B	B	D	A	D	A	D	C	D	D	D	B	C	C	C	C	D	A	A	B
H	A	A	C	D	B	D	B	B	D	D	C	C	A	D	B	D	B	C	A	B