Međuispit iz Uvoda u umjetnu inteligenciju (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 heurističko pretraživanje (7 pitanja)

1 (T) Algoritam A^* čvorove proširuje prema vrijednosti funkcije f. Ta funkcija za čvor n kombinira cijenu puta do čvora n, $g(n)$, te optimističnu procjenu cijene od čvora n do ciljnog stanja, $h(s)$. Neka je $h^*(s)$ stvarna minimalna cijena puta od s do ciljnog stanja te neka je $C^* = h^*(s_0)$ ukupna minimalna cijena puta od početnog do ciljnog stanja. **Što od navedenoga vrijedi?**

A Za čvor n koji je na putu minimalne cijene vrijedi $g(n) \leq f(n) \leq C^*$

B Za svaki čvor n vrijedi $g(n) + h^*(\text{state}(n)) = C^*$

C Za svaki čvor n vrijedi $g(n) \leq f(n) \leq C^*$

D Za čvor n_2 koji je sljedbenik čvora n_1 vrijedi $f(n_1) \leq f(n_2) \leq C^*$

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

A 1 B 7 C 5 D 6

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

A 78729 B 85257 C 85293 D 78696

5 (R) Algoritmom slijepog pretraživanja rješavamo problem misionara i kanibala. Tri misionara i tri kanibala potrebno je jednim čamcem prevesti s jedne strane obale rijeke na drugu, pri čemu se niti u jednom trenutku na jednoj strani obale ne smije naći više kanibala nego misionara. Čamac može prevesti najviše dvije osobe i ne može ploviti prazan. Stanje prikazujemo kao trojku MKC, gdje su M i K broj misionara odnosno kanibala na lijevoj obali rijeke, a C je strana obale na kojoj se nalazi čamac. Npr., u stanju 32D na lijevoj obali su tri misionara i dva kanibala (jedan kanibal je onda na desnoj obali), a čamac je na desnoj obali. Početno stanje je 33L, a završno 00D. Funkcija sljedbenika generira sva legitimna stanja, uključivo i ona koja rezultiraju neuspjehom i nemaju sljedbenika, npr., stanje 23D. Koristimo algoritam iterativnog pretraživanja u dubinu (IDS) bez liste posjećenih stanja. Kod proširenja čvora, algoritam čvorove djecu generira prema numeričkom redoslijedu; npr., stanje 13D prethodi stanju 23D. **Koliko će čvorova algoritam IDS ispitati prije nego što proširi čvor sa stanjem** 32D?

A 11 B 14 C 16 D 18

6 (T) Algoritam iterativnog pretraživanja u dubinu u svakoj iteraciji dubinsku granicu k povećava za jedan. **Što će se dogoditi ako dubinsku granicu povećamo za 2**?

A Algoritam više neće biti potpun
C Algoritam će se ubrzati za faktor $\log 2$
B Vremenska složenost će se smanjiti za faktor 4
D Algoritam više neće biti optimalan

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

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

2. Igranje igara (3 pitanja)

8 (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): $0, 3, 4, -3, 4, -1, 0, 2$. 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 1 B 3 C 2 D 4

9 (T) Algoritam minimax oslanja se na heurističku funkciju h, koja se poziva na svim stanjima igre na zadanoj dubini d. Neka je b faktor grananja igre. Neka su T i S gornje ograde vremenske odnosno prostorne složenosti. **Kakav utjecaj na gornje ograde složenosti algoritma minimax ima povećanje dubine d za jedan?**

A T raste b puta, S se ne mijenja
C T raste b puta, S se povećava za b
B T se povećava za b, S raste b puta
D T se povećava za 1, S raste b^{d+1} puta

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

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

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

A $A, D, J, \ldots$ B $A, C, G, \ldots$ C $A, D, I, \ldots$ D $A, C, H, \ldots$

3. Prikazivanje znanja i automatsko zaključivanje (6 pitanja)

11 (T) Logička posljedica središnji je koncept formalne logike. **Koji je nužan i dovoljan uvjet da formula G bude logička posljedica formule F?**

A Formula G je nezadovoljiva ako je $\neg F$ istina
B Formule F i G imaju identične vrijednosti istinitosti kroz sve intepretacije
C Svaki model formule G ujedno je i model formule F
D Formula G je istinita u svim modelima formule F

12 (P) Neka $A(x)$ označava "x je Amerikanac", $H(x)$ označava "x je hamburger", a $V(x,y)$ označava "x voli y". Pomoću ovih predikata formalizirajte u FOL rečenicu *"Neki Amerikanci ne vole hamburger"*. **Kako glasi ispravna formalizacija te rečenice?**

A $\exists x \forall y\Big(\big(A(x) \wedge H(y)\big) \rightarrow \neg V(x,y)\Big)$ C $\forall x\Big(A(x) \rightarrow \big(\forall y(H(y) \rightarrow \neg V(x,y))\big)\Big)$
B $\exists x\Big(A(x) \wedge \forall y\big(V(x,y) \rightarrow \neg H(y)\big)\Big)$ D $\exists x \forall y\Big(A(x) \wedge H(y) \wedge \neg V(x,y)\Big)$

13 (P) Rezolucija u FOL oslanja se na nalaženje najopćenitije zajedničke instance atoma pomoću algoritma MGU. Pritom, kao i u PL, važnu ulogu igra faktorizacija klauzula. Primijenite rezoluciju na par klauzula $\{P(x), R(x)\}$ i $\{\neg P(y), Q(a), Q(y)\}$. **Koja je od navedenih klauzula rezolventa ovog para klauzula?**

A $\{R(a), Q(a)\}$ B $\{R(a), Q(x)\}$ C $\{R(x), Q(a), Q(y)\}$ D $\{R(y)\}$

14 (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?**

A *Jednorog je magičan.*

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

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

D *Jednorog je obična smrtna životinja ili rogat.*

15 (R) Zadane su sljedeće dvije premise:

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

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

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

16 (T) Rezolucija opovrgavanjem u FOL je ispravna i potpuna. Međutim, dokazivanje valjanosti formule u FOL nije odlučivo. **Koji je efekt nedlučivosti FOL na ispravnost i potpunost rezolucije opovrgavanjem?**

A Ako je konjunkcija premisa i negacije zaključka zadovoljiva, rezolucijski postupak ne mora završiti

B Neke deduktivne posljedice dobivene rezolucijom opovrgavanjem nisu logičke posljedice skupa premisa

C Ako su premise nezadovoljive, rezolucija opovrgavanjem uvijek izvodi NIL klauzulu

D Rezolucijski postupak završava u konačno mnogo koraka ako i samo ako je cilj logička posljedica premisa

4. Logičko programiranje i ekspertni sustavi (4 pitanja)

17 (R) Baza znanja u Prologu modelira odnose između bioloških vrsta. Baza sadrži sljedeće činjenice i pravila:

```
podvrsta(krokodil, reptil).
podvrsta(ptica, reptil).
podvrsta(kokoška, ptica).
potomak(X, Y) :- podvrsta(X, Y).
potomak(X, Y) :- podvrsta(X, Z), potomak(Z, Y).
leti(X) :- potomak(X, ptica).
simbionti(X, Y) :- leti(Y), not(leti(X)).
```

Nad ovako definiranom bazom znanja izvodimo upit `simbionti(krokodil, kokoška)`. Prolog će za ovaj upit vratiti `True`. Nacrtajte Prologovo stablo dokaza za ovaj upit. Pritom za dokazivanje negiranog cilja koristite dva čvora (jedan za `not(P(X))` i drugi za `P(X)`), dok prazni stog ne brojite kao čvor. **Koliko čvorova ima stablo dokaza za ovaj upit?**

A 12 B 14 C 10 D 8

18 (T) Prolog programe izvodi provodeći rezoluciju opovrgavanjem nad klauzulama koje čine logički program. Pritom Prolog koristi stog na kojemu se nalaze atomi. **Koji atomi sačinjavaju Prologov stog?**

- A Atomi klauzula iz skupa premisa
- B Atomi klauzula iz skupa potpore
- C Atomi činjenica iz logičkog programa
- D Atomi glave pravila iz logičkog programa

19 (T) Kao i dokazivači teorema, ekspertni sustavi predstavljaju simbolički pristup umjetnoj inteligenciji. Međutim, ekspertni sustavi razlikuju se od dokazivača teorema po svojoj funkcionalnosti i svrsi. **Po čemu se ekspertni sustavi razlikuje od dokazivača teorema?**

- A Ekspertni sustavi znanje formaliziraju pravilima koja nalikuju implikacijama, dok dokazivači teorema rade s proizvoljnim logičkim formulama
- B Dokazivači teorema mogu poništiti jednom izvedenu formulu, dok ekspertni sustavi mogu samo dodavati činjenice u radnu memoriju
- C Dokazivači teorema oblikovani su za zaključivanje unutar specifične domene, dok ekspertni sustavi pokrivaju opće ljudsko znanje
- D Ekspertni sustavi mogu se koristiti za dijagnostiku ulančavanjem unazad, dok dokazivači teorema koriste rezoluciju, koja provodi ulančavanje unaprijed

20 (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 Pali dva pravila i izvodi $D = d_2$
- B Izvodi $E = e_2$ te kasnije $C = c_2$
- C Pali tri pravila i izvodi $D = d_1$
- D Završava s tri činjenice u radnoj memoriji

Sveučilište u Zagrebu
Fakultet elektrotehnike i računarstva

20. travnja 2023.

Međuispit iz Uvoda u umjetnu inteligenciju (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 heurističko pretraživanje (7 pitanja)

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

A 5 B 7 C 1 D 2

2 (T) Algoritam iterativnog pretraživanja u dubinu u svakoj iteraciji dubinsku granicu k povećava za jedan. **Što će se dogoditi ako dubinsku granicu povećamo za 2**?

A Algoritam više neće biti potpun
B Vremenska složenost će se smanjiti za faktor 4
C Algoritam će se ubrzati za faktor $\log 2$
D Algoritam više neće biti optimalan

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

A 78696 B 78729 C 85293 D 85257

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

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

5 (T) Algoritam A^* čvorove proširuje prema vrijednosti funkcije f. Ta funkcija za čvor n kombinira cijenu puta do čvora n, $g(n)$, te optimističnu procjenu cijene od čvora n do ciljnog stanja, $h(s)$. Neka je $h^*(s)$ stvarna minimalna cijena puta od s do ciljnog stanja te neka je $C^* = h^*(s_0)$ ukupna minimalna cijena puta od početnog do ciljnog stanja. **Što od navedenoga vrijedi?**

A Za čvor n_2 koji je sljedbenik čvora n_1 vrijedi $f(n_1) \leq f(n_2) \leq C^*$
B Za čvor n koji je na putu minimalne cijene vrijedi $g(n) \leq f(n) \leq C^*$
C Za svaki čvor n vrijedi $g(n) + h^*(\text{state}(n)) = C^*$
D Za svaki čvor n vrijedi $g(n) \leq f(n) \leq C^*$

6 (R) Algoritmom slijepog pretraživanja rješavamo problem misionara i kanibala. Tri misionara i tri kanibala potrebno je jednim čamcem prevesti s jedne strane obale rijeke na drugu, pri čemu se niti u jednom trenutku na jednoj strani obale ne smije naći više kanibala nego misionara. Čamac može prevesti najviše dvije osobe i ne može ploviti prazan. Stanje prikazujemo kao trojku MKC, gdje su M i K broj misionara odnosno kanibala na lijevoj obali rijeke, a C je strana obale na kojoj se nalazi čamac. Npr., u stanju 32D na lijevoj obali su tri misionara i dva kanibala (jedan kanibal je onda na desnoj obali), a čamac je na desnoj obali. Početno stanje je 33L, a završno 00D. Funkcija sljedbenika generira sva legitimna stanja, uključivo i ona koja rezultiraju neuspjehom i nemaju sljedbenika, npr., stanje 23D. Koristimo algoritam iterativnog pretraživanja u dubinu (IDS) bez liste posjećenih stanja. Kod proširenja čvora, algoritam čvorove djecu generira prema numeričkom redoslijedu; npr., stanje 13D prethodi stanju 23D. **Koliko će čvorova algoritam IDS ispitati prije nego što proširi čvor sa stanjem** 32D?

A 18 B 14 C 16 D 11

7 (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 $h(b) = 0$ C $h(b) = 4$ D $h(b) = 3$

2. Igranje igara (3 pitanja)

8 (T) Algoritam minimax oslanja se na heurističku funkciju h, koja se poziva na svim stanjima igre na zadanoj dubini d. Neka je b faktor grananja igre. Neka su T i S gornje ograde vremenske odnosno prostorne složenosti. **Kakav utjecaj na gornje ograde složenosti algoritma minimax ima povećanje dubine d za jedan?**

A T se povećava za b, S raste b puta

B T se povećava za 1, S raste b^{d+1} puta

C T raste b puta, S se povećava za b

D T raste b puta, S se ne mijenja

9 (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): $0, 3, 4, -3, 4, -1, 0, 2$. 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 3 B 1 C 2 D 4

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

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

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

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

3. Prikazivanje znanja i automatsko zaključivanje (6 pitanja)

11 (P) Neka $A(x)$ označava "x je Amerikanac", $H(x)$ označava "x je hamburger", a $V(x, y)$ označava "x voli y". Pomoću ovih predikata formalizirajte u FOL rečenicu *"Neki Amerikanci ne vole hamburger"*. **Kako glasi ispravna formalizacija te rečenice?**

A $\forall x\Big(A(x) \to \big(\exists y(H(y) \to \neg V(x, y))\big)\Big)$

B $\exists x \forall y\Big(A(x) \wedge \big(H(y) \to \neg V(x, y)\big)\Big)$

C $\exists x \forall y\Big(A(x) \wedge H(y) \wedge \neg V(x, y)\Big)$

D $\exists x\Big(A(x) \to \big(\exists y(H(y) \to \neg V(x, y))\big)\Big)$

12 (T) Rezolucija opovrgavanjem u FOL je ispravna i potpuna. Međutim, dokazivanje valjanosti formule u FOL nije odlučivo. **Koji je efekt nedlučivosti FOL na ispravnost i potpunost rezolucije opovrgavanjem?**

- A Ako je konjunkcija premisa i negacije zaključka zadovoljiva, rezolucijski postupak ne mora završiti
- B Rezolucijski postupak završava u konačno mnogo koraka ako i samo ako je cilj logička posljedica premisa
- C Ako su premise nezadovoljive, rezolucija opovrgavanjem uvijek izvodi NIL klauzulu
- D Neke deduktivne posljedice dobivene rezolucijom opovrgavanjem nisu logičke posljedice skupa premisa

13 (P) Rezolucija u FOL oslanja se na nalaženje najopćenitije zajedničke instance atoma pomoću algoritma MGU. Pritom, kao i u PL, važnu ulogu igra faktorizacija klauzula. Primijenite rezoluciju na par klauzula $\{P(x), R(x)\}$ i $\{\neg P(y), Q(a), Q(y)\}$. **Koja je od navedenih klauzula rezolventa ovog para klauzula?**

A $\{R(a), Q(a)\}$ B $\{R(x), Q(a), Q(y)\}$ C $\{R(a), Q(x)\}$ D $\{R(y)\}$

14 (R) Zadane su sljedeće dvije premise:

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

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

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

15 (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?**

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

16 (T) Logička posljedica središnji je koncept formalne logike. **Koji je nužan i dovoljan uvjet da formula G bude logička posljedica formule F?**

- A Formule F i G imaju identične vrijednosti istinitosti kroz sve intepretacije
- B Formula G je nezadovoljiva ako je $\neg F$ istina
- C Svaki model formule G ujedno je i model formule F
- D Formula G je istinita u svim modelima formule F

4. Logičko programiranje i ekspertni sustavi (4 pitanja)

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

18 (T) Prolog programe izvodi provodeći rezoluciju opovrgavanjem nad klauzulama koje čine logički program. Pritom Prolog koristi stog na kojemu se nalaze atomi. **Koji atomi sačinjavaju Prologov stog?**

- A Atomi klauzula iz skupa premisa
- B Atomi glave pravila iz logičkog programa
- C Atomi klauzula iz skupa potpore
- D Atomi činjenica iz logičkog programa

19 (R) Baza znanja u Prologu modelira odnose između bioloških vrsta. Baza sadrži sljedeće činjenice i pravila:

```
podvrsta(krokodil, reptil).
podvrsta(ptica, reptil).
podvrsta(kokoška, ptica).
potomak(X, Y) :- podvrsta(X, Y).
potomak(X, Y) :- podvrsta(X, Z), potomak(Z, Y).
leti(X) :- potomak(X, ptica).
simbionti(X, Y) :- leti(Y), not(leti(X)).
```

Nad ovako definiranom bazom znanja izvodimo upit `simbionti(krokodil, kokoška)`. Prolog će za ovaj upit vratiti `True`. Nacrtajte Prologovo stablo dokaza za ovaj upit. Pritom za dokazivanje negiranog cilja koristite dva čvora (jedan za `not(P(X))` i drugi za `P(X)`), dok prazni stog ne brojite kao čvor. **Koliko čvorova ima stablo dokaza za ovaj upit?**

A 12 B 14 C 8 D 10

20 (T) Kao i dokazivači teorema, ekspertni sustavi predstavljaju simbolički pristup umjetnoj inteligenciji. Međutim, ekspertni sustavi razlikuju se od dokazivača teorema po svojoj funkcionalnosti i svrsi. **Po čemu se ekspertni sustavi razlikuje od dokazivača teorema?**

A Dokazivači teorema oblikovani su za zaključivanje unutar specifične domene, dok ekspertni sustavi pokrivaju opće ljudsko znanje

B Ekspertni sustavi mogu se koristiti za dijagnostiku ulančavanjem unazad, dok dokazivači teorema koriste rezoluciju, koja provodi ulančavanje unaprijed

C Ekspertni sustavi znanje formaliziraju pravilima koja nalikuju implikacijama, dok dokazivači teorema rade s proizvoljnim logičkim formulama

D Dokazivači teorema mogu poništiti jednom izvedenu formulu, dok ekspertni sustavi mogu samo dodavati činjenice u radnu memoriju

Međuispit iz Uvoda u umjetnu inteligenciju (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 heurističko pretraživanje (7 pitanja)

1 (T) Algoritam iterativnog pretraživanja u dubinu u svakoj iteraciji dubinsku granicu k povećava za jedan. **Što će se dogoditi ako dubinsku granicu povećamo za 2**?

A Algoritam više neće biti optimalan
B Algoritam više neće biti potpun
C Vremenska složenost će se smanjiti za faktor 4
D Algoritam će se ubrzati za faktor $\log 2$

2 (R) Algoritmom slijepog pretraživanja rješavamo problem misionara i kanibala. Tri misionara i tri kanibala potrebno je jednim čamcem prevesti s jedne strane obale rijeke na drugu, pri čemu se niti u jednom trenutku na jednoj strani obale ne smije naći više kanibala nego misionara. Čamac može prevesti najviše dvije osobe i ne može ploviti prazan. Stanje prikazujemo kao trojku MKC, gdje su M i K broj misionara odnosno kanibala na lijevoj obali rijeke, a C je strana obale na kojoj se nalazi čamac. Npr., u stanju 32D na lijevoj obali su tri misionara i dva kanibala (jedan kanibal je onda na desnoj obali), a čamac je na desnoj obali. Početno stanje je 33L, a završno 00D. Funkcija sljedbenika generira sva legitimna stanja, uključivo i ona koja rezultiraju neuspjehom i nemaju sljedbenika, npr., stanje 23D. Koristimo algoritam iterativnog pretraživanja u dubinu (IDS) bez liste posjećenih stanja. Kod proširenja čvora, algoritam čvorove djecu generira prema numeričkom redoslijedu; npr., stanje 13D prethodi stanju 23D. **Koliko će čvorova algoritam IDS ispitati prije nego što proširi čvor sa stanjem 32D?**

A 18 B 11 C 14 D 16

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

A 4 B 2 C 6 D 0

4 (T) Algoritam A^* čvorove proširuje prema vrijednosti funkcije f. Ta funkcija za čvor n kombinira cijenu puta do čvora n, $g(n)$, te optimističnu procjenu cijene od čvora n do ciljnog stanja, $h(s)$. Neka je $h^*(s)$ stvarna minimalna cijena puta od s do ciljnog stanja te neka je $C^* = h^*(s_0)$ ukupna minimalna cijena puta od početnog do ciljnog stanja. **Što od navedenoga vrijedi?**

A Za čvor n_2 koji je sljedbenik čvora n_1 vrijedi $f(n_1) \leq f(n_2) \leq C^*$
B Za čvor n koji je na putu minimalne cijene vrijedi $g(n) \leq f(n) \leq C^*$
C Za svaki čvor n vrijedi $g(n) \leq f(n) \leq C^*$
D Za svaki čvor n vrijedi $g(n) + h^*(\text{state}(n)) = C^*$

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

A $O = [(e, 7), (f, 10), (b, 13)],\ C = \{(a, 0), (b, 1), (c, 2), (d, 5)\}$
B $O = [(f, 9)],\ C = \{(a, 0), (b, 1), (c, 2), (d, 5), (e, 7)\}$
C Algoritam ne dostiže peti korak
D $O = [(f, 0), (e, 2), (b, 4)],\ C = \{(a, 0), (b, 1), (c, 2), (d, 5)\}$

6 (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 $h(c) = 2$

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

A 85257 B 78729 C 78696 D 85293

2. Igranje igara (3 pitanja)

8 (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): $0, 3, 4, -3, 4, -1, 0, 2$. 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 3 B 2 C 1 D 4

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

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

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

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

10 (T) Algoritam minimax oslanja se na heurističku funkciju h, koja se poziva na svim stanjima igre na zadanoj dubini d. Neka je b faktor grananja igre. Neka su T i S gornje ograde vremenske odnosno prostorne složenosti. **Kakav utjecaj na gornje ograde složenosti algoritma minimax ima povećanje dubine d za jedan?**

A T raste b puta, S se povećava za b
C T se povećava za 1, S raste b^{d+1} puta
B T se povećava za b, S raste b puta
D T raste b puta, S se ne mijenja

3. Prikazivanje znanja i automatsko zaključivanje (6 pitanja)

11 (R) Zadane su sljedeće dvije premise:

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

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

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

12 (P) Neka $A(x)$ označava "x je Amerikanac", $H(x)$ označava "x je hamburger", a $V(x,y)$ označava "x voli y". Pomoću ovih predikata formalizirajte u FOL rečenicu *"Neki Amerikanci ne vole hamburger"*. **Kako glasi ispravna formalizacija te rečenice?**

A $\exists x\Big(A(x) \land \exists y\big(H(y) \rightarrow \neg V(x,y)\big)\Big)$
B $\exists x \forall y\Big(A(x) \land H(y) \land \neg V(x,y)\Big)$
C $\exists x\Big(A(x) \land \forall y\big(H(y) \rightarrow \neg V(x,y)\big)\Big)$
D $\forall x\Big(A(x) \rightarrow \big(\exists y(H(y) \rightarrow \neg V(x,y))\big)\Big)$

13 (T) Rezolucija opovrgavanjem u FOL je ispravna i potpuna. Međutim, dokazivanje valjanosti formule u FOL nije odlučivo. **Koji je efekt neodlučivosti FOL na ispravnost i potpunost rezolucije opovrgavanjem?**

A Rezolucijski postupak završava u konačno mnogo koraka ako i samo ako je cilj logička posljedica premisa
B Ako su premise nezadovoljive, rezolucija opovrgavanjem uvijek izvodi NIL klauzulu
C Ako je konjunkcija premisa i negacije zaključka zadovoljiva, rezolucijski postupak ne mora završiti
D Neke deduktivne posljedice dobivene rezolucijom opovrgavanjem nisu logičke posljedice skupa premisa

14 (T) Logička posljedica središnji je koncept formalne logike. **Koji je nužan i dovoljan uvjet da formula G bude logička posljedica formule F?**

A Formula G je nezadovoljiva ako je $\neg F$ istina
B Svaki model formule G ujedno je i model formule F
C Formula G je istinita u svim modelima formule F
D Formule F i G imaju identične vrijednosti istinitosti kroz sve intepretacije

15 (P) Rezolucija u FOL oslanja se na nalaženje najopćenitije zajedničke instance atoma pomoću algoritma MGU. Pritom, kao i u PL, važnu ulogu igra faktorizacija klauzula. Primijenite rezoluciju na par klauzula $\{P(x), R(x)\}$ i $\{\neg P(y), Q(a), Q(y)\}$. **Koja je od navedenih klauzula rezolventa ovog para klauzula?**

A $\{R(a), Q(a)\}$ B $\{R(x), Q(a), Q(y)\}$ C $\{R(y)\}$ D $\{R(a), Q(x)\}$

16 (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?**

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

4. Logičko programiranje i ekspertni sustavi (4 pitanja)

17 (T) Prolog programe izvodi provodeći rezoluciju opovrgavanjem nad klauzulama koje čine logički program. Pritom Prolog koristi stog na kojemu se nalaze atomi. **Koji atomi sačinjavaju Prologov stog?**

A Atomi klauzula iz skupa premisa
B Atomi glave pravila iz logičkog programa
C Atomi klauzula iz skupa potpore
D Atomi činjenica iz logičkog programa

18 (T) Kao i dokazivači teorema, ekspertni sustavi predstavljaju simbolički pristup umjetnoj inteligenciji. Međutim, ekspertni sustavi razlikuju se od dokazivača teorema po svojoj funkcionalnosti i svrsi. **Po čemu se ekspertni sustavi razlikuje od dokazivača teorema?**

A Ekspertni sustavi znanje formaliziraju pravilima koja nalikuju implikacijama, dok dokazivači teorema rade s proizvoljnim logičkim formulama
B Dokazivači teorema mogu poništiti jednom izvedenu formulu, dok ekspertni sustavi mogu samo dodavati činjenice u radnu memoriju
C Ekspertni sustavi mogu se koristiti za dijagnostiku ulančavanjem unazad, dok dokazivači teorema koriste rezoluciju, koja provodi ulančavanje unaprijed
D Dokazivači teorema oblikovani su za zaključivanje unutar specifične domene, dok ekspertni sustavi pokrivaju opće ljudsko znanje

19 (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 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 $C = c_2$
C Odbacuje pravilo 3 te kasnije pali pravilo 5
B Pali tri pravila i izvodi $D = d_1$
D Izvodi $B = b_3$ te kasnije $A = a_2$

20 (R) Baza znanja u Prologu modelira odnose između bioloških vrsta. Baza sadrži sljedeće činjenice i pravila:

```
podvrsta(krokodil, reptil).
podvrsta(ptica, reptil).
podvrsta(kokoška, ptica).
potomak(X, Y) :- podvrsta(X, Y).
potomak(X, Y) :- podvrsta(X, Z), potomak(Z, Y).
leti(X) :- potomak(X, ptica).
simbionti(X, Y) :- leti(Y), not(leti(X)).
```

Nad ovako definiranom bazom znanja izvodimo upit `simbionti(krokodil, kokoška)`. Prolog će za ovaj upit vratiti `True`. Nacrtajte Prologovo stablo dokaza za ovaj upit. Pritom za dokazivanje negiranog cilja koristite dva čvora (jedan za `not(P(X))` i drugi za `P(X)`), dok prazni stog ne brojite kao čvor. **Koliko čvorova ima stablo dokaza za ovaj upit?**

A 12 B 10 C 8 D 14

Međuispit iz Uvoda u umjetnu inteligenciju (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 heurističko pretraživanje (7 pitanja)

1 (T) Algoritam iterativnog pretraživanja u dubinu u svakoj iteraciji dubinsku granicu k povećava za jedan. **Što će se dogoditi ako dubinsku granicu povećamo za 2**?

- A Algoritam više neće biti optimalan
- B Algoritam će se ubrzati za faktor $\log 2$
- C Algoritam više neće biti potpun
- D Vremenska složenost će se smanjiti za faktor 4

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

A 8   B 3   C 1   D 6

3 (T) Algoritam A^* čvorove proširuje prema vrijednosti funkcije f. Ta funkcija za čvor n kombinira cijenu puta do čvora n, $g(n)$, te optimističnu procjenu cijene od čvora n do ciljnog stanja, $h(s)$. Neka je $h^*(s)$ stvarna minimalna cijena puta od s do ciljnog stanja te neka je $C^* = h^*(s_0)$ ukupna minimalna cijena puta od početnog do ciljnog stanja. **Što od navedenoga vrijedi?**

- A Za svaki čvor n vrijedi $g(n) + h^*(\text{state}(n)) = C^*$
- B Za čvor n koji je na putu minimalne cijene vrijedi $g(n) \leq f(n) \leq C^*$
- C Za čvor n_2 koji je sljedbenik čvora n_1 vrijedi $f(n_1) \leq f(n_2) \leq C^*$
- D Za svaki čvor n vrijedi $g(n) \leq f(n) \leq C^*$

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

A 78729   B 78696   C 85257   D 85293

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

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

A $O = [(e, 7), (f, 10)]$, $C = \{(a, 0), (c, 2), (d, 5)\}$

B Algoritam ne dostiže peti korak

C $O = [(f, 0), (e, 2), (b, 4)]$, $C = \{(a, 0), (b, 1), (c, 2), (d, 5)\}$

D $O = [(f, 9)]$, $C = \{(a, 0), (b, 1), (c, 2), (d, 5), (e, 7)\}$

7 (R) Algoritmom slijepog pretraživanja rješavamo problem misionara i kanibala. Tri misionara i tri kanibala potrebno je jednim čamcem prevesti s jedne strane obale rijeke na drugu, pri čemu se niti u jednom trenutku na jednoj strani obale ne smije naći više kanibala nego misionara. Čamac može prevesti najviše dvije osobe i ne može ploviti prazan. Stanje prikazujemo kao trojku MKC, gdje su M i K broj misionara odnosno kanibala na lijevoj obali rijeke, a C je strana obale na kojoj se nalazi čamac. Npr., u stanju 32D na lijevoj obali su tri misionara i dva kanibala (jedan kanibal je onda na desnoj obali), a čamac je na desnoj obali. Početno stanje je 33L, a završno 00D. Funkcija sljedbenika generira sva legitimna stanja, uključivo i ona koja rezultiraju neuspjehom i nemaju sljedbenika, npr., stanje 23D. Koristimo algoritam iterativnog pretraživanja u dubinu (IDS) bez liste posjećenih stanja. Kod proširenja čvora, algoritam čvorove djecu generira prema numeričkom redoslijedu; npr., stanje 13D prethodi stanju 23D. **Koliko će čvorova algoritam IDS ispitati prije nego što proširi čvor sa stanjem** 32D?

A 14 B 11 C 18 D 16

2. Igranje igara (3 pitanja)

8 (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): $0, 3, 4, -3, 4, -1, 0, 2$. 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 2 D 1

9 (T) Algoritam minimax oslanja se na heurističku funkciju h, koja se poziva na svim stanjima igre na zadanoj dubini d. Neka je b faktor grananja igre. Neka su T i S gornje ograde vremenske odnosno prostorne složenosti. **Kakav utjecaj na gornje ograde složenosti algoritma minimax ima povećanje dubine d za jedan?**

A T raste b puta, S se povećava za b

B T se povećava za 1, S raste b^{d+1} puta

C T se povećava za b, S raste b puta

D T raste b puta, S se ne mijenja

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

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

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

A $A, D, J, \ldots$ B $A, C, G, \ldots$ C $A, D, I, \ldots$ D $A, C, H, \ldots$

3. Prikazivanje znanja i automatsko zaključivanje (6 pitanja)

11 (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?**

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

12 (T) Logička posljedica središnji je koncept formalne logike. **Koji je nužan i dovoljan uvjet da formula G bude logička posljedica formule F?**

- A Formula G je nezadovoljiva ako je $\neg F$ istina
- B Formula G je istinita u svim modelima formule F
- C Svaki model formule G ujedno je i model formule F
- D Formule F i G imaju identične vrijednosti istinitosti kroz sve intepretacije

13 (T) Rezolucija opovrgavanjem u FOL je ispravna i potpuna. Međutim, dokazivanje valjanosti formule u FOL nije odlučivo. **Koji je efekt nedlučivosti FOL na ispravnost i potpunost rezolucije opovrgavanjem?**

- A Ako je konjunkcija premisa i negacije zaključka zadovoljiva, rezolucijski postupak ne mora završiti
- B Rezolucijski postupak završava u konačno mnogo koraka ako i samo ako je cilj logička posljedica premisa
- C Ako su premise nezadovoljive, rezolucija opovrgavanjem uvijek izvodi NIL klauzulu
- D Neke deduktivne posljedice dobivene rezolucijom opovrgavanjem nisu logičke posljedice skupa premisa

14 (P) Neka $A(x)$ označava "x je Amerikanac", $H(x)$ označava "x je hamburger", a $V(x,y)$ označava "x voli y". Pomoću ovih predikata formalizirajte u FOL rečenicu *"Neki Amerikanci ne vole hamburger"*. **Kako glasi ispravna formalizacija te rečenice?**

- A $\exists x\Big(A(x) \rightarrow \big(\forall y(\neg V(x,y) \rightarrow H(y))\big)\Big)$
- C $\exists x\Big(A(x) \rightarrow \big(\forall y(\neg H(y) \lor \neg V(x,y))\big)\Big)$
- B $\exists x\Big(A(x) \land \forall y\big(\neg H(y) \lor \neg V(x,y)\big)\Big)$
- D $\exists x\forall y\Big(A(x) \land H(y) \land \neg V(x,y)\Big)$

15 (P) Rezolucija u FOL oslanja se na nalaženje najopćenitije zajedničke instance atoma pomoću algoritma MGU. Pritom, kao i u PL, važnu ulogu igra faktorizacija klauzula. Primijenite rezoluciju na par klauzula $\{P(x), R(x)\}$ i $\{\neg P(y), Q(a), Q(y)\}$. **Koja je od navedenih klauzula rezolventa ovog para klauzula?**

A $\{R(a), Q(a)\}$ B $\{R(y)\}$ C $\{R(a), Q(x)\}$ D $\{R(x), Q(a), Q(y)\}$

16 (R) Zadane su sljedeće dvije premise:

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

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

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

4. Logičko programiranje i ekspertni sustavi (4 pitanja)

17 (T) Kao i dokazivači teorema, ekspertni sustavi predstavljaju simbolički pristup umjetnoj inteligenciji. Međutim, ekspertni sustavi razlikuju se od dokazivača teorema po svojoj funkcionalnosti i svrsi. **Po čemu se ekspertni sustavi razlikuje od dokazivača teorema?**

- A Ekspertni sustavi znanje formaliziraju pravilima koja nalikuju implikacijama, dok dokazivači teorema rade s proizvoljnim logičkim formulama
- B Ekspertni sustavi mogu se koristiti za dijagnostiku ulančavanjem unazad, dok dokazivači teorema koriste rezoluciju, koja provodi ulančavanje unaprijed
- C Dokazivači teorema mogu poništiti jednom izvedenu formulu, dok ekspertni sustavi mogu samo dodavati činjenice u radnu memoriju
- D Dokazivači teorema oblikovani su za zaključivanje unutar specifične domene, dok ekspertni sustavi pokrivaju opće ljudsko znanje

18 (R) Baza znanja u Prologu modelira odnose između bioloških vrsta. Baza sadrži sljedeće činjenice i pravila:

```
podvrsta(krokodil, reptil).
podvrsta(ptica, reptil).
podvrsta(kokoška, ptica).
potomak(X, Y) :- podvrsta(X, Y).
potomak(X, Y) :- podvrsta(X, Z), potomak(Z, Y).
leti(X) :- potomak(X, ptica).
simbionti(X, Y) :- leti(Y), not(leti(X)).
```

Nad ovako definiranom bazom znanja izvodimo upit `simbionti(krokodil, kokoška)`. Prolog će za ovaj upit vratiti `True`. Nacrtajte Prologovo stablo dokaza za ovaj upit. Pritom za dokazivanje negiranog cilja koristite dva čvora (jedan za `not(P(X))` i drugi za `P(X)`), dok prazni stog ne brojite kao čvor. **Koliko čvorova ima stablo dokaza za ovaj upit?**

A 12 B 10 C 8 D 14

19 (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 Odbacuje pravilo 5 te kasnije pali pravilo 6
- D Završava s tri činjenice u radnoj memoriji

20 (T) Prolog programe izvodi provodeći rezoluciju opovrgavanjem nad klauzulama koje čine logički program. Pritom Prolog koristi stog na kojemu se nalaze atomi. **Koji atomi sačinjavaju Prologov stog?**

- A Atomi glave pravila iz logičkog programa
- B Atomi klauzula iz skupa premisa
- C Atomi klauzula iz skupa potpore
- D Atomi činjenica iz logičkog programa

Međuispit iz Uvoda u umjetnu inteligenciju (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 heurističko pretraživanje (7 pitanja)

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

- A $O = [(f, 9)],\ C = \{(a, 0), (b, 1), (c, 2), (d, 5), (e, 7)\}$
- B $O = [(f, 0), (e, 2), (b, 4)],\ C = \{(a, 0), (b, 1), (c, 2), (d, 5)\}$
- C $O = [(e, 7), (f, 10)],\ C = \{(a, 0), (b, 1), (c, 2), (d, 5)\}$
- D Algoritam ne dostiže peti korak

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 $h(c) = 2$, $h(d) = h(e) = 3$ 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 $h(b) = 4$ B $h(b) = 1$ C $h(b) = 5$ D $h(b) = 3$

3 (T) Algoritam A^* čvorove proširuje prema vrijednosti funkcije f. Ta funkcija za čvor n kombinira cijenu puta do čvora n, $g(n)$, te optimističnu procjenu cijene od čvora n do ciljnog stanja, $h(s)$. Neka je $h^*(s)$ stvarna minimalna cijena puta od s do ciljnog stanja te neka je $C^* = h^*(s_0)$ ukupna minimalna cijena puta od početnog do ciljnog stanja. **Što od navedenoga vrijedi?**

- A Za svaki čvor n vrijedi $g(n) \leq f(n) \leq C^*$
- B Za čvor n_2 koji je sljedbenik čvora n_1 vrijedi $f(n_1) \leq f(n_2) \leq C^*$
- C Za čvor n koji je na putu minimalne cijene vrijedi $g(n) \leq f(n) \leq C^*$
- D Za svaki čvor n vrijedi $g(n) + h^*(\text{state}(n)) = C^*$

4 (R) Algoritmom slijepog pretraživanja rješavamo problem misionara i kanibala. Tri misionara i tri kanibala potrebno je jednim čamcem prevesti s jedne strane obale rijeke na drugu, pri čemu se niti u jednom trenutku na jednoj strani obale ne smije naći više kanibala nego misionara. Čamac može prevesti najviše dvije osobe i ne može ploviti prazan. Stanje prikazujemo kao trojku MKC, gdje su M i K broj misionara odnosno kanibala na lijevoj obali rijeke, a C je strana obale na kojoj se nalazi čamac. Npr., u stanju 32D na lijevoj obali su tri misionara i dva kanibala (jedan kanibal je onda na desnoj obali), a čamac je na desnoj obali. Početno stanje je 33L, a završno 00D. Funkcija sljedbenika generira sva legitimna stanja, uključivo i ona koja rezultiraju neuspjehom i nemaju sljedbenika, npr., stanje 23D. Koristimo algoritam iterativnog pretraživanja u dubinu (IDS) bez liste posjećenih stanja. Kod proširenja čvora, algoritam čvorove djecu generira prema numeričkom redoslijedu; npr., stanje 13D prethodi stanju 23D. **Koliko će čvorova algoritam IDS ispitati prije nego što proširi čvor sa stanjem 32D?**

A 16 B 11 C 18 D 14

5 (T) Algoritam iterativnog pretraživanja u dubinu u svakoj iteraciji dubinsku granicu k povećava za jedan. **Što će se dogoditi ako dubinsku granicu povećamo za 2**?

A Algoritam će se ubrzati za faktor $\log 2$
B Algoritam više neće biti potpun
C Algoritam više neće biti optimalan
D Vremenska složenost će se smanjiti za faktor 4

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

A 78696 B 85293 C 78729 D 85257

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

A 7 B 2 C 6 D 9

2. Igranje igara (3 pitanja)

8 (T) Algoritam minimax oslanja se na heurističku funkciju h, koja se poziva na svim stanjima igre na zadanoj dubini d. Neka je b faktor grananja igre. Neka su T i S gornje ograde vremenske odnosno prostorne složenosti. **Kakav utjecaj na gornje ograde složenosti algoritma minimax ima povećanje dubine d za jedan?**

A T se povećava za b, S raste b puta
B T se povećava za 1, S raste b^{d+1} puta
C T raste b puta, S se ne mijenja
D T raste b puta, S se povećava za b

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

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

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

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

10 (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): $0, 3, 4, -3, 4, -1, 0, 2$. 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

3. Prikazivanje znanja i automatsko zaključivanje (6 pitanja)

11 (T) Logička posljedica središnji je koncept formalne logike. **Koji je nužan i dovoljan uvjet da formula G bude logička posljedica formule F?**

A Formula G je nezadovoljiva ako je $\neg F$ istina
B Formule F i G imaju identične vrijednosti istinitosti kroz sve intepretacije
C Svaki model formule G ujedno je i model formule F
D Formula G je istinita u svim modelima formule F

12 (P) Rezolucija u FOL oslanja se na nalaženje najopćenitije zajedničke instance atoma pomoću algoritma MGU. Pritom, kao i u PL, važnu ulogu igra faktorizacija klauzula. Primijenite rezoluciju na par klauzula $\{P(x), R(x)\}$ i $\{\neg P(y), Q(a), Q(y)\}$. **Koja je od navedenih klauzula rezolventa ovog para klauzula?**

A $\{R(a), Q(x)\}$ B $\{R(y)\}$ C $\{R(x), Q(a), Q(y)\}$ D $\{R(a), Q(a)\}$

13 (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?**

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

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

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

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

14 (P) Neka $A(x)$ označava "x je Amerikanac", $H(x)$ označava "x je hamburger", a $V(x, y)$ označava "x voli y". Pomoću ovih predikata formalizirajte u FOL rečenicu *"Neki Amerikanci ne vole hamburger"*. **Kako glasi ispravna formalizacija te rečenice?**

A $\exists x\Big(A(x) \rightarrow \big(\exists y(H(y) \rightarrow \neg V(x, y))\big)\Big)$ C $\exists x \forall y\Big(A(x) \wedge H(y) \wedge \neg V(x, y)\Big)$

B $\exists x\Big(A(x) \wedge \forall y\big(\neg H(y) \vee \neg V(x, y)\big)\Big)$ D $\exists x\Big(A(x) \wedge \exists y\big(H(y) \rightarrow \neg V(x, y)\big)\Big)$

15 (T) Rezolucija opovrgavanjem u FOL je ispravna i potpuna. Međutim, dokazivanje valjanosti formule u FOL nije odlučivo. **Koji je efekt nedlučivosti FOL na ispravnost i potpunost rezolucije opovrgavanjem?**

A Ako je konjunkcija premisa i negacije zaključka zadovoljiva, rezolucijski postupak ne mora završiti

B Rezolucijski postupak završava u konačno mnogo koraka ako i samo ako je cilj logička posljedica premisa

C Neke deduktivne posljedice dobivene rezolucijom opovrgavanjem nisu logičke posljedice skupa premisa

D Ako su premise nezadovoljive, rezolucija opovrgavanjem uvijek izvodi NIL klauzulu

16 (R) Zadane su sljedeće dvije premise:

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

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

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

4. Logičko programiranje i ekspertni sustavi (4 pitanja)

17 (T) Kao i dokazivači teorema, ekspertni sustavi predstavljaju simbolički pristup umjetnoj inteligenciji. Međutim, ekspertni sustavi razlikuju se od dokazivača teorema po svojoj funkcionalnosti i svrsi. **Po čemu se ekspertni sustavi razlikuje od dokazivača teorema?**

A Dokazivači teorema oblikovani su za zaključivanje unutar specifične domene, dok ekspertni sustavi pokrivaju opće ljudsko znanje

B Ekspertni sustavi znanje formaliziraju pravilima koja nalikuju implikacijama, dok dokazivači teorema rade s proizvoljnim logičkim formulama

C Dokazivači teorema mogu poništiti jednom izvedenu formulu, dok ekspertni sustavi mogu samo dodavati činjenice u radnu memoriju

D Ekspertni sustavi mogu se koristiti za dijagnostiku ulančavanjem unazad, dok dokazivači teorema koriste rezoluciju, koja provodi ulančavanje unaprijed

18 (R) Baza znanja u Prologu modelira odnose između bioloških vrsta. Baza sadrži sljedeće činjenice i pravila:

```
podvrsta(krokodil, reptil).
podvrsta(ptica, reptil).
podvrsta(kokoška, ptica).
potomak(X, Y) :- podvrsta(X, Y).
potomak(X, Y) :- podvrsta(X, Z), potomak(Z, Y).
leti(X) :- potomak(X, ptica).
simbionti(X, Y) :- leti(Y), not(leti(X)).
```

Nad ovako definiranom bazom znanja izvodimo upit `simbionti(krokodil, kokoška)`. Prolog će za ovaj upit vratiti `True`. Nacrtajte Prologovo stablo dokaza za ovaj upit. Pritom za dokazivanje negiranog cilja koristite dva čvora (jedan za `not(P(X))` i drugi za `P(X)`), dok prazni stog ne brojite kao čvor. **Koliko čvorova ima stablo dokaza za ovaj upit?**

A 10 B 14 C 8 D 12

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

20 (T) Prolog programe izvodi provodeći rezoluciju opovrgavanjem nad klauzulama koje čine logički program. Pritom Prolog koristi stog na kojemu se nalaze atomi. **Koji atomi sačinjavaju Prologov stog?**

A Atomi klauzula iz skupa premisa
C Atomi činjenica iz logičkog programa
B Atomi glave pravila iz logičkog programa
D Atomi klauzula iz skupa potpore

Međuispit iz Uvoda u umjetnu inteligenciju (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 heurističko pretraživanje (7 pitanja)

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

A 78729   B 85257   C 85293   D 78696

2 (T) Algoritam A^* čvorove proširuje prema vrijednosti funkcije f. Ta funkcija za čvor n kombinira cijenu puta do čvora n, $g(n)$, te optimističnu procjenu cijene od čvora n do ciljnog stanja, $h(s)$. Neka je $h^*(s)$ stvarna minimalna cijena puta od s do ciljnog stanja te neka je $C^* = h^*(s_0)$ ukupna minimalna cijena puta od početnog do ciljnog stanja. **Što od navedenoga vrijedi?**

A Za svaki čvor n vrijedi $g(n) + h^*(\text{state}(n)) = C^*$

B Za čvor n koji je na putu minimalne cijene vrijedi $g(n) \leq f(n) \leq C^*$

C Za čvor n_2 koji je sljedbenik čvora n_1 vrijedi $f(n_1) \leq f(n_2) \leq C^*$

D Za svaki čvor n vrijedi $g(n) \leq f(n) \leq C^*$

3 (R) Algoritmom slijepog pretraživanja rješavamo problem misionara i kanibala. Tri misionara i tri kanibala potrebno je jednim čamcem prevesti s jedne strane obale rijeke na drugu, pri čemu se niti u jednom trenutku na jednoj strani obale ne smije naći više kanibala nego misionara. Čamac može prevesti najviše dvije osobe i ne može ploviti prazan. Stanje prikazujemo kao trojku MKC, gdje su M i K broj misionara odnosno kanibala na lijevoj obali rijeke, a C je strana obale na kojoj se nalazi čamac. Npr., u stanju 32D na lijevoj obali su tri misionara i dva kanibala (jedan kanibal je onda na desnoj obali), a čamac je na desnoj obali. Početno stanje je 33L, a završno 00D. Funkcija sljedbenika generira sva legitimna stanja, uključivo i ona koja rezultiraju neuspjehom i nemaju sljedbenika, npr., stanje 23D. Koristimo algoritam iterativnog pretraživanja u dubinu (IDS) bez liste posjećenih stanja. Kod proširenja čvora, algoritam čvorove djecu generira prema numeričkom redoslijedu; npr., stanje 13D prethodi stanju 23D. **Koliko će čvorova algoritam IDS ispitati prije nego što proširi čvor sa stanjem 32D?**

A 16   B 18   C 11   D 14

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

A 5   B 4   C 6   D 1

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

A $O = [(e, 7), (f, 10), (b, 13)]$, $C = \{(a, 0), (b, 1), (c, 2), (d, 5)\}$

B $O = [(e, 7), (f, 10)]$, $C = \{(a, 0), (b, 1), (c, 2), (d, 5)\}$

C $O = [(f, 9)]$, $C = \{(a, 0), (b, 1), (c, 2), (d, 5), (e, 7)\}$

D Algoritam ne dostiže peti korak

6 (T) Algoritam iterativnog pretraživanja u dubinu u svakoj iteraciji dubinsku granicu k povećava za jedan. **Što će se dogoditi ako dubinsku granicu povećamo za 2**?

- A Algoritam će se ubrzati za faktor $\log 2$
- C Algoritam više neće biti optimalan
- B Algoritam više neće biti potpun
- D Vremenska složenost će se smanjiti za faktor 4

7 (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 $h(b) = 4$ C $h(b) = 1$ D $h(b) = 3$

2. Igranje igara (3 pitanja)

8 (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): $0, 3, 4, -3, 4, -1, 0, 2$. 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

9 (T) Algoritam minimax oslanja se na heurističku funkciju h, koja se poziva na svim stanjima igre na zadanoj dubini d. Neka je b faktor grananja igre. Neka su T i S gornje ograde vremenske odnosno prostorne složenosti. **Kakav utjecaj na gornje ograde složenosti algoritma minimax ima povećanje dubine d za jedan?**

- A T se povećava za b, S raste b puta
- C T raste b puta, S se ne mijenja
- B T raste b puta, S se povećava za b
- D T se povećava za 1, S raste b^{d+1} puta

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

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

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

A $A, D, J, \ldots$ B $A, C, G, \ldots$ C $A, C, H, \ldots$ D $A, D, I, \ldots$

3. Prikazivanje znanja i automatsko zaključivanje (6 pitanja)

11 (P) Neka $A(x)$ označava "x je Amerikanac", $H(x)$ označava "x je hamburger", a $V(x, y)$ označava "x voli y". Pomoću ovih predikata formalizirajte u FOL rečenicu *"Neki Amerikanci ne vole hamburger"*. **Kako glasi ispravna formalizacija te rečenice?**

- A $\exists x \forall y \Big(\big(A(x) \land H(y)\big) \rightarrow \neg V(x, y) \Big)$
- C $\exists x \Big(A(x) \land \forall y \big(\neg H(y) \lor \neg V(x, y) \big) \Big)$
- B $\forall x \exists y \Big(\big(A(x) \land H(y)\big) \rightarrow \neg V(x, y) \Big)$
- D $\exists x \forall y \Big(A(x) \land H(y) \land \neg V(x, y) \Big)$

12 (T) Rezolucija opovrgavanjem u FOL je ispravna i potpuna. Međutim, dokazivanje valjanosti formule u FOL nije odlučivo. **Koji je efekt nedlučivosti FOL na ispravnost i potpunost rezolucije opovrgavanjem?**

- A Ako su premise nezadovoljive, rezolucija opovrgavanjem uvijek izvodi NIL klauzulu
- B Rezolucijski postupak završava u konačno mnogo koraka ako i samo ako je cilj logička posljedica premisa
- C Neke deduktivne posljedice dobivene rezolucijom opovrgavanjem nisu logičke posljedice skupa premisa
- D Ako je konjunkcija premisa i negacije zaključka zadovoljiva, rezolucijski postupak ne mora završiti

13 (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?**

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

14 (T) Logička posljedica središnji je koncept formalne logike. **Koji je nužan i dovoljan uvjet da formula G bude logička posljedica formule F?**

- A Formula G je nezadovoljiva ako je $\neg F$ istina
- B Svaki model formule G ujedno je i model formule F
- C Formula G je istinita u svim modelima formule F
- D Formule F i G imaju identične vrijednosti istinitosti kroz sve intepretacije

15 (R) Zadane su sljedeće dvije premise:

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

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

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

16 (P) Rezolucija u FOL oslanja se na nalaženje najopćenitije zajedničke instance atoma pomoću algoritma MGU. Pritom, kao i u PL, važnu ulogu igra faktorizacija klauzula. Primijenite rezoluciju na par klauzula $\{P(x), R(x)\}$ i $\{\neg P(y), Q(a), Q(y)\}$. **Koja je od navedenih klauzula rezolventa ovog para klauzula?**

A $\{R(x), Q(a), Q(y)\}$ B $\{R(a), Q(x)\}$ C $\{R(y)\}$ D $\{R(a), Q(a)\}$

4. Logičko programiranje i ekspertni sustavi (4 pitanja)

17 (R) Baza znanja u Prologu modelira odnose između bioloških vrsta. Baza sadrži sljedeće činjenice i pravila:

```
podvrsta(krokodil, reptil).
podvrsta(ptica, reptil).
podvrsta(kokoška, ptica).
potomak(X, Y) :- podvrsta(X, Y).
potomak(X, Y) :- podvrsta(X, Z), potomak(Z, Y).
leti(X) :- potomak(X, ptica).
simbionti(X, Y) :- leti(Y), not(leti(X)).
```

Nad ovako definiranom bazom znanja izvodimo upit `simbionti(krokodil, kokoška)`. Prolog će za ovaj upit vratiti `True`. Nacrtajte Prologovo stablo dokaza za ovaj upit. Pritom za dokazivanje negiranog cilja koristite dva čvora (jedan za `not(P(X))` i drugi za `P(X)`), dok prazni stog ne brojite kao čvor. **Koliko čvorova ima stablo dokaza za ovaj upit?**

A 12 B 8 C 14 D 10

18 (T) Prolog programe izvodi provodeći rezoluciju opovrgavanjem nad klauzulama koje čine logički program. Pritom Prolog koristi stog na kojemu se nalaze atomi. **Koji atomi sačinjavaju Prologov stog?**

- A Atomi klauzula iz skupa potpore
- B Atomi klauzula iz skupa premisa
- C Atomi glave pravila iz logičkog programa
- D Atomi činjenica iz logičkog programa

19 (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 Odbacuje pravilo 3 te kasnije pali pravilo 5
- C Odbacuje pravilo 5 te kasnije pali pravilo 6
- D Pali tri pravila i izvodi $D = d_1$

20 (T) Kao i dokazivači teorema, ekspertni sustavi predstavljaju simbolički pristup umjetnoj inteligenciji. Međutim, ekspertni sustavi razlikuju se od dokazivača teorema po svojoj funkcionalnosti i svrsi. **Po čemu se ekspertni sustavi razlikuje od dokazivača teorema?**

- A Ekspertni sustavi znanje formaliziraju pravilima koja nalikuju implikacijama, dok dokazivači teorema rade s proizvoljnim logičkim formulama
- B Dokazivači teorema mogu poništiti jednom izvedenu formulu, dok ekspertni sustavi mogu samo dodavati činjenice u radnu memoriju
- C Ekspertni sustavi mogu se koristiti za dijagnostiku ulančavanjem unazad, dok dokazivači teorema koriste rezoluciju, koja provodi ulančavanje unaprijed
- D Dokazivači teorema oblikovani su za zaključivanje unutar specifične domene, dok ekspertni sustavi pokrivaju opće ljudsko znanje

Međuispit iz Uvoda u umjetnu inteligenciju (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 heurističko pretraživanje (7 pitanja)

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

A 78729 B 85293 C 85257 D 78696

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

A 9 B 0 C 2 D 3

3 (R) Algoritmom slijepog pretraživanja rješavamo problem misionara i kanibala. Tri misionara i tri kanibala potrebno je jednim čamcem prevesti s jedne strane obale rijeke na drugu, pri čemu se niti u jednom trenutku na jednoj strani obale ne smije naći više kanibala nego misionara. Čamac može prevesti najviše dvije osobe i ne može ploviti prazan. Stanje prikazujemo kao trojku MKC, gdje su M i K broj misionara odnosno kanibala na lijevoj obali rijeke, a C je strana obale na kojoj se nalazi čamac. Npr., u stanju 32D na lijevoj obali su tri misionara i dva kanibala (jedan kanibal je onda na desnoj obali), a čamac je na desnoj obali. Početno stanje je 33L, a završno 00D. Funkcija sljedbenika generira sva legitimna stanja, uključivo i ona koja rezultiraju neuspjehom i nemaju sljedbenika, npr., stanje 23D. Koristimo algoritam iterativnog pretraživanja u dubinu (IDS) bez liste posjećenih stanja. Kod proširenja čvora, algoritam čvorove djecu generira prema numeričkom redoslijedu; npr., stanje 13D prethodi stanju 23D. **Koliko će čvorova algoritam IDS ispitati prije nego što proširi čvor sa stanjem** 32D?

A 14 B 11 C 16 D 18

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

A $O = [(f, 9)]$, $C = \{(a, 0), (b, 1), (c, 2), (d, 5), (e, 7)\}$

B $O = [(e, 7), (f, 10), (b, 13)]$, $C = \{(a, 0), (b, 1), (c, 2), (d, 5)\}$

C Algoritam ne dostiže peti korak

D $O = [(f, 0), (e, 2), (b, 4)]$, $C = \{(a, 0), (b, 1), (c, 2), (d, 5)\}$

5 (T) Algoritam iterativnog pretraživanja u dubinu u svakoj iteraciji dubinsku granicu k povećava za jedan. **Što će se dogoditi ako dubinsku granicu povećamo za 2**?

A Algoritam više neće biti potpun

B Algoritam će se ubrzati za faktor $\log 2$

C Algoritam više neće biti optimalan

D Vremenska složenost će se smanjiti za faktor 4

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

7 (T) Algoritam A^* čvorove proširuje prema vrijednosti funkcije f. Ta funkcija za čvor n kombinira cijenu puta do čvora n, $g(n)$, te optimističnu procjenu cijene od čvora n do ciljnog stanja, $h(s)$. Neka je $h^*(s)$ stvarna minimalna cijena puta od s do ciljnog stanja te neka je $C^* = h^*(s_0)$ ukupna minimalna cijena puta od početnog do ciljnog stanja. **Što od navedenoga vrijedi?**

A Za svaki čvor n vrijedi $g(n) \le f(n) \le C^*$

B Za svaki čvor n vrijedi $g(n) + h^*(\text{state}(n)) = C^*$

C Za čvor n_2 koji je sljedbenik čvora n_1 vrijedi $f(n_1) \le f(n_2) \le C^*$

D Za čvor n koji je na putu minimalne cijene vrijedi $g(n) \le f(n) \le C^*$

2. Igranje igara (3 pitanja)

8 (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): $0, 3, 4, -3, 4, -1, 0, 2$. 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 1 B 4 C 3 D 2

9 (T) Algoritam minimax oslanja se na heurističku funkciju h, koja se poziva na svim stanjima igre na zadanoj dubini d. Neka je b faktor grananja igre. Neka su T i S gornje ograde vremenske odnosno prostorne složenosti. **Kakav utjecaj na gornje ograde složenosti algoritma minimax ima povećanje dubine d za jedan?**

A T se povećava za 1, S raste b^{d+1} puta

B T raste b puta, S se povećava za b

C T raste b puta, S se ne mijenja

D T se povećava za b, S raste b puta

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

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

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

A $A, C, G, \ldots$ B $A, D, J, \ldots$ C $A, D, I, \ldots$ D $A, C, H, \ldots$

3. Prikazivanje znanja i automatsko zaključivanje (6 pitanja)

11 (T) Logička posljedica središnji je koncept formalne logike. **Koji je nužan i dovoljan uvjet da formula G bude logička posljedica formule F?**

- A Formule F i G imaju identične vrijednosti istinitosti kroz sve intepretacije
- B Svaki model formule G ujedno je i model formule F
- C Formula G je istinita u svim modelima formule F
- D Formula G je nezadovoljiva ako je $\neg F$ istina

12 (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?**

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

13 (T) Rezolucija opovrgavanjem u FOL je ispravna i potpuna. Međutim, dokazivanje valjanosti formule u FOL nije odlučivo. **Koji je efekt neodlučivosti FOL na ispravnost i potpunost rezolucije opovrgavanjem?**

- A Rezolucijski postupak završava u konačno mnogo koraka ako i samo ako je cilj logička posljedica premisa
- B Ako su premise nezadovoljive, rezolucija opovrgavanjem uvijek izvodi NIL klauzulu
- C Neke deduktivne posljedice dobivene rezolucijom opovrgavanjem nisu logičke posljedice skupa premisa
- D Ako je konjunkcija premisa i negacije zaključka zadovoljiva, rezolucijski postupak ne mora završiti

14 (R) Zadane su sljedeće dvije premise:

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

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

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

15 (P) Neka $A(x)$ označava "x je Amerikanac", $H(x)$ označava "x je hamburger", a $V(x,y)$ označava "x voli y". Pomoću ovih predikata formalizirajte u FOL rečenicu *"Neki Amerikanci ne vole hamburger"*. **Kako glasi ispravna formalizacija te rečenice?**

- A $\exists x\Big(A(x) \to \big(\forall y \neg(H(y) \lor \neg V(x,y))\big)\Big)$
- C $\exists x\Big(A(x) \land \forall y\big(\neg H(y) \lor \neg V(x,y)\big)\Big)$
- B $\exists x \forall y\Big(A(x) \land H(y) \land \neg V(x,y)\Big)$
- D $\forall x \exists y\Big(\big(A(x) \land H(y)\big) \to \neg V(x,y)\Big)$

16 (P) Rezolucija u FOL oslanja se na nalaženje najopćenitije zajedničke instance atoma pomoću algoritma MGU. Pritom, kao i u PL, važnu ulogu igra faktorizacija klauzula. Primijenite rezoluciju na par klauzula $\{P(x), R(x)\}$ i $\{\neg P(y), Q(a), Q(y)\}$. **Koja je od navedenih klauzula rezolventa ovog para klauzula?**

- A $\{R(y)\}$
- B $\{R(a), Q(a)\}$
- C $\{R(x), Q(a), Q(y)\}$
- D $\{R(a), Q(x)\}$

4. Logičko programiranje i ekspertni sustavi (4 pitanja)

17 (R) Baza znanja u Prologu modelira odnose između bioloških vrsta. Baza sadrži sljedeće činjenice i pravila:

```
podvrsta(krokodil, reptil).
podvrsta(ptica, reptil).
podvrsta(kokoška, ptica).
potomak(X, Y) :- podvrsta(X, Y).
potomak(X, Y) :- podvrsta(X, Z), potomak(Z, Y).
leti(X) :- potomak(X, ptica).
simbionti(X, Y) :- leti(Y), not(leti(X)).
```

Nad ovako definiranom bazom znanja izvodimo upit `simbionti(krokodil, kokoška)`. Prolog će za ovaj upit vratiti `True`. Nacrtajte Prologovo stablo dokaza za ovaj upit. Pritom za dokazivanje negiranog cilja koristite dva čvora (jedan za `not(P(X))` i drugi za `P(X)`), dok prazni stog ne brojite kao čvor. **Koliko čvorova ima stablo dokaza za ovaj upit?**

A 8 B 10 C 12 D 14

18 (T) Prolog programe izvodi provodeći rezoluciju opovrgavanjem nad klauzulama koje čine logički program. Pritom Prolog koristi stog na kojemu se nalaze atomi. **Koji atomi sačinjavaju Prologov stog?**

A Atomi klauzula iz skupa premisa

B Atomi glave pravila iz logičkog programa

C Atomi činjenica iz logičkog programa

D Atomi klauzula iz skupa potpore

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

C Odbacuje pravilo 5 te kasnije pali pravilo 6

D Pali tri pravila i izvodi $D = d_1$

20 (T) Kao i dokazivači teorema, ekspertni sustavi predstavljaju simbolički pristup umjetnoj inteligenciji. Međutim, ekspertni sustavi razlikuju se od dokazivača teorema po svojoj funkcionalnosti i svrsi. **Po čemu se ekspertni sustavi razlikuje od dokazivača teorema?**

A Ekspertni sustavi mogu se koristiti za dijagnostiku ulančavanjem unazad, dok dokazivači teorema koriste rezoluciju, koja provodi ulančavanje unaprijed

B Dokazivači teorema oblikovani su za zaključivanje unutar specifične domene, dok ekspertni sustavi pokrivaju opće ljudsko znanje

C Dokazivači teorema mogu poništiti jednom izvedenu formulu, dok ekspertni sustavi mogu samo dodavati činjenice u radnu memoriju

D Ekspertni sustavi znanje formaliziraju pravilima koja nalikuju implikacijama, dok dokazivači teorema rade s proizvoljnim logičkim formulama

Međuispit iz Uvoda u umjetnu inteligenciju (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 heurističko pretraživanje (7 pitanja)

1 (T) Algoritam iterativnog pretraživanja u dubinu u svakoj iteraciji dubinsku granicu k povećava za jedan. **Što će se dogoditi ako dubinsku granicu povećamo za 2**?

A Algoritam više neće biti potpun
C Algoritam će se ubrzati za faktor $\log 2$
B Algoritam više neće biti optimalan
D Vremenska složenost će se smanjiti za faktor 4

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

A 85293 B 85257 C 78696 D 78729

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

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

4 (R) Algoritmom slijepog pretraživanja rješavamo problem misionara i kanibala. Tri misionara i tri kanibala potrebno je jednim čamcem prevesti s jedne strane obale rijeke na drugu, pri čemu se niti u jednom trenutku na jednoj strani obale ne smije naći više kanibala nego misionara. Čamac može prevesti najviše dvije osobe i ne može ploviti prazan. Stanje prikazujemo kao trojku MKC, gdje su M i K broj misionara odnosno kanibala na lijevoj obali rijeke, a C je strana obale na kojoj se nalazi čamac. Npr., u stanju 32D na lijevoj obali su tri misionara i dva kanibala (jedan kanibal je onda na desnoj obali), a čamac je na desnoj obali. Početno stanje je 33L, a završno 00D. Funkcija sljedbenika generira sva legitimna stanja, uključivo i ona koja rezultiraju neuspjehom i nemaju sljedbenika, npr., stanje 23D. Koristimo algoritam iterativnog pretraživanja u dubinu (IDS) bez liste posjećenih stanja. Kod proširenja čvora, algoritam čvorove djecu generira prema numeričkom redoslijedu; npr., stanje 13D prethodi stanju 23D. **Koliko će čvorova algoritam IDS ispitati prije nego što proširi čvor sa stanjem 32D?**

A 16 B 14 C 11 D 18

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

6 (T) Algoritam A^* čvorove proširuje prema vrijednosti funkcije f. Ta funkcija za čvor n kombinira cijenu puta do čvora n, $g(n)$, te optimističnu procjenu cijene od čvora n do ciljnog stanja, $h(s)$. Neka je $h^*(s)$ stvarna minimalna cijena puta od s do ciljnog stanja te neka je $C^* = h^*(s_0)$ ukupna minimalna cijena puta od početnog do ciljnog stanja. **Što od navedenoga vrijedi?**

- A Za čvor n_2 koji je sljedbenik čvora n_1 vrijedi $f(n_1) \leq f(n_2) \leq C^*$
- B Za čvor n koji je na putu minimalne cijene vrijedi $g(n) \leq f(n) \leq C^*$
- C Za svaki čvor n vrijedi $g(n) + h^*(\text{state}(n)) = C^*$
- D Za svaki čvor n vrijedi $g(n) \leq f(n) \leq C^*$

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

A 1 B 7 C 4 D 2

2. Igranje igara (3 pitanja)

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

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

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

A $A, D, I, \ldots$ B $A, C, G, \ldots$ C $A, D, J, \ldots$ D $A, C, H, \ldots$

9 (T) Algoritam minimax oslanja se na heurističku funkciju h, koja se poziva na svim stanjima igre na zadanoj dubini d. Neka je b faktor grananja igre. Neka su T i S gornje ograde vremenske odnosno prostorne složenosti. **Kakav utjecaj na gornje ograde složenosti algoritma minimax ima povećanje dubine d za jedan?**

- A T raste b puta, S se ne mijenja
- B T se povećava za b, S raste b puta
- C T se povećava za 1, S raste b^{d+1} puta
- D T raste b puta, S se povećava za b

10 (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): $0, 3, 4, -3, 4, -1, 0, 2$. 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 1 B 2 C 3 D 4

3. Prikazivanje znanja i automatsko zaključivanje (6 pitanja)

11 (P) Neka $A(x)$ označava "x je Amerikanac", $H(x)$ označava "x je hamburger", a $V(x, y)$ označava "x voli y". Pomoću ovih predikata formalizirajte u FOL rečenicu *"Neki Amerikanci ne vole hamburger"*. **Kako glasi ispravna formalizacija te rečenice?**

- A $\exists x\Big(A(x) \land \forall y\big(V(x, y) \to \neg H(y)\big)\Big)$
- B $\exists x \forall y\Big(A(x) \land H(y) \land \neg V(x, y)\Big)$
- C $\exists x\Big(A(x) \to \big(\forall y(\neg H(y) \lor \neg V(x, y))\big)\Big)$
- D $\exists x\Big(A(x) \to \big(\forall y(V(x, y) \to \neg H(y))\big)\Big)$

12 (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?**

A *Jednorog je obična smrtna životinja ili magičan.*

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

C *Jednorog je mitsko biće.*

D *Jednorog je magičan.*

13 (T) Logička posljedica središnji je koncept formalne logike. **Koji je nužan i dovoljan uvjet da formula G bude logička posljedica formule F?**

A Svaki model formule G ujedno je i model formule F

B Formule F i G imaju identične vrijednosti istinitosti kroz sve intepretacije

C Formula G je nezadovoljiva ako je $\neg F$ istina

D Formula G je istinita u svim modelima formule F

14 (P) Rezolucija u FOL oslanja se na nalaženje najopćenitije zajedničke instance atoma pomoću algoritma MGU. Pritom, kao i u PL, važnu ulogu igra faktorizacija klauzula. Primijenite rezoluciju na par klauzula $\{P(x), R(x)\}$ i $\{\neg P(y), Q(a), Q(y)\}$. **Koja je od navedenih klauzula rezolventa ovog para klauzula?**

A $\{R(x), Q(a), Q(y)\}$ B $\{R(a), Q(x)\}$ C $\{R(y)\}$ D $\{R(a), Q(a)\}$

15 (T) Rezolucija opovrgavanjem u FOL je ispravna i potpuna. Međutim, dokazivanje valjanosti formule u FOL nije odlučivo. **Koji je efekt nedlučivosti FOL na ispravnost i potpunost rezolucije opovrgavanjem?**

A Ako je konjunkcija premisa i negacije zaključka zadovoljiva, rezolucijski postupak ne mora završiti

B Rezolucijski postupak završava u konačno mnogo koraka ako i samo ako je cilj logička posljedica premisa

C Ako su premise nezadovoljive, rezolucija opovrgavanjem uvijek izvodi NIL klauzulu

D Neke deduktivne posljedice dobivene rezolucijom opovrgavanjem nisu logičke posljedice skupa premisa

16 (R) Zadane su sljedeće dvije premise:

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

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

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

4. Logičko programiranje i ekspertni sustavi (4 pitanja)

17 (T) Prolog programe izvodi provodeći rezoluciju opovrgavanjem nad klauzulama koje čine logički program. Pritom Prolog koristi stog na kojemu se nalaze atomi. **Koji atomi sačinjavaju Prologov stog?**

A Atomi glave pravila iz logičkog programa

B Atomi klauzula iz skupa potpore

C Atomi činjenica iz logičkog programa

D Atomi klauzula iz skupa premisa

18 (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 Pali tri pravila i izvodi $D = d_1$

B Završava s tri činjenice u radnoj memoriji

C Odbacuje pravilo 3 te kasnije pali pravilo 5

D Pali dva pravila i izvodi $D = d_2$

19 (T) Kao i dokazivači teorema, ekspertni sustavi predstavljaju simbolički pristup umjetnoj inteligenciji. Međutim, ekspertni sustavi razlikuju se od dokazivača teorema po svojoj funkcionalnosti i svrsi. **Po čemu se ekspertni sustavi razlikuje od dokazivača teorema?**

- A Ekspertni sustavi znanje formaliziraju pravilima koja nalikuju implikacijama, dok dokazivači teorema rade s proizvoljnim logičkim formulama
- B Dokazivači teorema oblikovani su za zaključivanje unutar specifične domene, dok ekspertni sustavi pokrivaju opće ljudsko znanje
- C Ekspertni sustavi mogu se koristiti za dijagnostiku ulančavanjem unazad, dok dokazivači teorema koriste rezoluciju, koja provodi ulančavanje unaprijed
- D Dokazivači teorema mogu poništiti jednom izvedenu formulu, dok ekspertni sustavi mogu samo dodavati činjenice u radnu memoriju

20 (R) Baza znanja u Prologu modelira odnose između bioloških vrsta. Baza sadrži sljedeće činjenice i pravila:

```
podvrsta(krokodil, reptil).
podvrsta(ptica, reptil).
podvrsta(kokoška, ptica).
potomak(X, Y) :- podvrsta(X, Y).
potomak(X, Y) :- podvrsta(X, Z), potomak(Z, Y).
leti(X) :- potomak(X, ptica).
simbionti(X, Y) :- leti(Y), not(leti(X)).
```

Nad ovako definiranom bazom znanja izvodimo upit `simbionti(krokodil, kokoška)`. Prolog će za ovaj upit vratiti `True`. Nacrtajte Prologovo stablo dokaza za ovaj upit. Pritom za dokazivanje negiranog cilja koristite dva čvora (jedan za `not(P(X))` i drugi za `P(X)`), dok prazni stog ne brojite kao čvor. **Koliko čvorova ima stablo dokaza za ovaj upit?**

A 10   B 8   C 14   D 12

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