PB016 – závěrečná písemka

12. ledna 2005 *Jméno a příjmení:* ....

UČO: ....

1

Uvažujme tvrzení:

(i) Algoritmus Minimax je úplný i pro nekonečné herní stromy.

(ii) Cyklus se v Prologu standardně řeší pomocí rekurze.

(iii) Volba optimální funkce z konzistentních funkcí při induktivním učení se často řídí pravidlem Ockhamovy břitvy – nejjednodušší z konzistentních je optimální.

(iv) Algoritmus IDT pro učení formou rozhodovacích stromů vybírá vždy ten atribut, který má největší rozdíl v míře informace před a po testu na tento atribut.

(v) Neuronová síť je schopná vyjádřit nelineární funkce i při použití lineární aktivační funkce.

(A) Tvrzení (ii), (iii) jsou pravdivá a tvrzení (i), (iv), (v) jsou nepravdivá.

(B) Tvrzení (iii) je pravdivé a tvrzení (i), (ii), (iv), (v) jsou nepravdivá.

(C) Tvrzení (i), (ii), (iii), (iv) jsou pravdivá a tvrzení (v) je nepravdivé.

(D) Tvrzení (ii) je pravdivé a tvrzení (i), (iii), (iv), (v) jsou nepravdivá.

(E) Tvrzení (ii), (iii), (iv) jsou pravdivá a tvrzení (i), (v) jsou nepravdivá.

Odpověď:

2

Při jaké hodnotě α, resp. β, dojde procedurou Alfa-Beta k odříznutí uzlů a kterých?

```
moves(a,[b,c]).
moves(b,[d,e]). moves(c,[f,g]).
moves(d,[h,i]). moves(e,[j,k]). moves(f,[l,m]). moves(g,[n,o]).
staticval(h,4). staticval(i,2). staticval(j,1). staticval(k,3).
staticval(l,5). staticval(m,3). staticval(n,1). staticval(o,4).
```

(A) $\alpha = 4$, uzly **j** a **k** **(B)** $\alpha = 4$, uzel **k** **(C)** $\beta = 3$, uzel **o** **(D)** $\beta = 5$, uzly **g**, **n** a **o**

(E) Nedojde k odříznutí žádného uzlu.

Odpověď:

3

Máme daný model $m = \{P_1 = true, P_2 = false, P_3 = true\}$. Určete pravdivosti formulí:

a) $\neg P_2 \wedge (P_2 \vee P_3)$

b) $\neg P_1 \vee (P_3 \Rightarrow (P_3 \wedge P_2))$

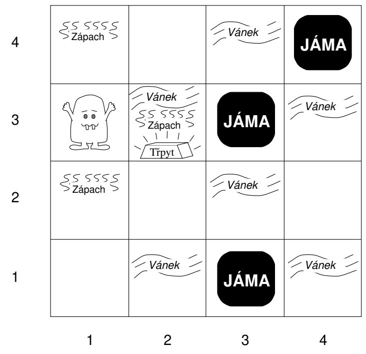
c) $P_1 \wedge (P_2 \Leftrightarrow P_3) \wedge (P_1 \Leftrightarrow P_2)$

(A) a) *true*, b) *false*, c) *true* **(B)** a) *false*, b) *true*, c) *false* **(C)** a) *true*, b) *true*, c) *false*

(D) a) *true*, b) *false*, c) *false* **(E)** a) *false*, b) *true*, c) *true*

Odpověď:

4



Máme instanci problému Wumpusovy jeskyně. Určete, která z následujících posloupností akcí *nemůže* být korektní posloupnost akcí logického agenta, tj. kdy návaznost akcí je proti logice tohoto problému.
Souřadnice jsou v pořadí [sloupec, řádek]. Před uvedenými kroky se v agentově bázi znalostí vyskytují pouze obecná pravidla Wumpusovy jeskyně.

(A) $[1, 4]$ směr jih: vlevo → výstřel → krok → vpravo → krok → zvedni

(B) $[1, 4]$ směr východ: vpravo → krok → vlevo → krok → zvedni

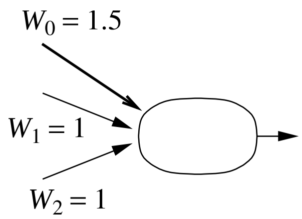
(C) $[1, 1]$ směr východ: krok → čelem vzad → krok → vpravo → krok → vpravo → krok → vlevo → krok → zvedni

(D) $[1, 4]$ směr jih: výstřel → vlevo → krok → vpravo → krok → zvedni

(E) $[1, 1]$ směr východ: vlevo → krok → čelem vzad → krok → vlevo → krok → vlevo → krok → krok → zvedni

Odpověď:

5



Máme zadanou neuronovou jednotku McCulloch & Pitts. Jakou Booleovskou funkci tento neuron řeší?

(A) not **(B)** xor **(C)** or **(D)** implikace **(E)** and

Odpověď:

6

Máme falešnou hrací kostku, která má nerovnoměrné rozložení pravděpodobností jednotlivých hodů:

1	0.05	4	0.23
2	0.19	5	0.08
3	0.12	6	0.33

Jaká je míra informace $I\big(\big\langle P(v_1), \ldots, P(v_n)\big\rangle\big) = \sum_{i=1}^{n} -P(v_i)\log_2 P(v_i)$ obsažená v odpovědi na otázku, jestli v dalším hodu na této kostce padne šestka? Zaokrouhlete na 2 destinná místa.
Hodnoty některých dvojkových logaritmů jsou: $\log_2 0.05 = -4.32$, $\log_2 0.08 = -3.64$, $\log_2 0.12 = -3.06$, $\log_2 0.19 = -2.4$, $\log_2 0.23 = -2.12$, $\log_2 0.25 = -2$, $\log_2 0.33 = -1.6$, $\log_2 0.36 = -1.47$, $\log_2 0.41 = -1.29$. $\log_2 0.59 = -0.76$, $\log_2 0.64 = -0.64$, $\log_2 0.67 = -0.58$, $\log_2 0.75 = -0.42$, $\log_2 1.19 = 0.25$, $\log_2 1.68 = 0.75$.

(A) 1 bit **(B)** 0.33 bitů **(C)** 0.92 bitů **(D)** 0.5 bitů **(E)** 0.59 bitů

Odpověď:

7

Mějme následující program:

```
p([],[]).
p([H|T],L):- p(T,L1), append(L1,[H],L).

?-p([a, b, [c, d, e]], L).
```

Určete hodnotu výsledného L.

(A) [e, d, c, b, a] **(B)** [[e, d, c], b, a] **(C)** [a, b, [c, d, e]] **(D)** [[c, d, e], b, a] **(E)** [c, d, e, b, a]

Odpověď:

8

Máme následující program:

```
?-op(100,xfx,to), dynamic(h/5).
h(1,A,B,C,[A to B]).
h(N,A,B,C,Moves) :- N>1, N1 is N-1, lemma(h(N1,A,C,B,Ms1)),
    h(N1,C,B,A,Ms2), xxx(Ms1,[A to B|Ms2],Moves).
lemma(P) :- P,asserta((P :- !)).
```

Určete, která definice predikátu **xxx/3** se nejlépe hodí do tohoto programu.

(A) `xxx(A,L,[A|L]).` `xxx(A,[H|T1],[H|T2]):- xxx(A,T1,T2).`

(B) `xxx(A,[A|T],T).` `xxx(A,[H|T1],[H|T2]) :- xxx(A,T1,T2).`

(C) `xxx(X,[Y|_], _) :- X == Y.` `xxx(X,[Y|T], L) :- X \== Y, xxx(X,T,L).`

(D) `xxx(A, [A|B], _).` `xxx(A, [B|C], D) :- xxx(A, C, D).`

(E) `xxx([], A, A).` `xxx([A|B], C, [A|D]) :- xxx(B, C, D).`

Odpověď:

9

Máme zadanou pravidlovou část DC gramatiky určitého typu českých vět:

```
S --> NP VP
NP --> ADJ NP
NP --> N
NP --> PREP N
VP --> VP ADV
VP --> V
VP --> VP NP
```

Zkratky neterminálů znamenají: S – věta, NP – jmenná skupina, VP – slovesná skupina, N – podstatné jméno, ADJ – přídavné jméno, V – sloveso, ADV – příslovce, PREP – předložka.
Určete, která z následujících vět *není generovatelná* touto gramatikou.

(A) Raněný rybář vyskočil včas z lodi. **(B)** Malý studentík navštěvoval vysokou školu.

(C) Burzovní akcie náhle poklesly. **(D)** Pes pokousal vystrašené malé dítě.

(E) Krásná krajina zaujme určitě každého turistu.

Odpověď:

10

Mějme AND/OR graf definovaný takto:

a ---> or: [b/1, c/3].
b ---> and: [d/1, e/1].
c ---> and: [f/2, g/1].
e ---> or: [h/6].
f ---> or: [h/2, i/3].
goal(d). goal(g). goal(h).

Jaké maximální hodnoty dosáhla F hodnota uzlu **b** v průběhu řešení?

(A) 7 **(B)** 9 **(C)** 2 **(D)** 11 **(E)** 8

Odpověď:

11

Mějme predikát **andor(T,ST)** prohledávání AND/OR grafu do hloubky definován následujícím způsobem:

andor(Node,Node) :- goal(Node).
andor(Node,Node → Tree) :- Node → or:Nodes, solveall(Nodes,Tree).
andor(Node,Node → and:Trees) :- Node → and:Nodes, solveall(Nodes,Trees).
solveall([],[]).
solveall([Node|Nodes],[Tree|Trees]) :- andor(Node,Tree), solveall(Nodes,Trees).

Řešení OR uzlů zjevně není správné. Opravte jej změnou na 2. řádku.

(A) **andor(Node,Tree) :- Node → or:Nodes, andor(Node,Tree).**

(B) **andor(Node,Node → Tree) :- Node → or:Nodes, member(Node1,Nodes), andor(Node1,Tree).**

(C) **andor(Node,Tree) :- Node → or:Nodes, andor(Nodes,Tree).**

(D) **andor(Node,Node → Tree) :- Node → or:Nodes, member(Node,Nodes), andor(Node,Tree).**

(E) **andor(Node,Node → Trees) :- Node → or:Nodes, andor(Node,Trees).**

Odpověď:

12

Uvažujme tvrzení:

(i) Algoritmus A* expanduje všechny uzly, které mají $f(n) < C^*$ (C^* je cena optimálního řešení) a některé uzly, které mají $f(n) = C^*$.

(ii) Stanovení limitů pro hodnoty α a β v algoritmu Alfa-Beta u *nedeterministických* her nemá vliv na množství ořezaných uzlů.

(iii) Algoritmus Alfa-Beta je efektivnější než Minimax, protože odřezává uzly, které nemůžou mít vliv na výsledný optimální tah.

(iv) Extenzionalismus predikátové logiky označuje její neschopnost vyjádřit intenze, tj. funkce, jejichž hodnoty závisí na světě a čase.

(v) Inkrementální formulace problémů s omezujícími podmínkami umožňuje převést řešení na standardní prohledávání do hloubky.

(A) Právě jedno z uvedených tvrzení je pravdivé. **(B)** Přesně dvě z uvedených tvrzení jsou pravdivá.

(C) Přesně dvě z uvedených tvrzení jsou nepravdivá. **(D)** Právě jedno z uvedených tvrzení je nepravdivé.

(E) Všechna uvedená tvrzení jsou pravdivá.

Odpověď: