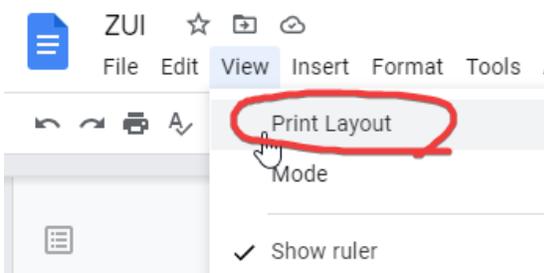


ZUI otázky z prvního testu (a druhého)



Doporučuji zrušit si print layout

Question # 1 Type: Free answer, Maximum score: 1, Score:1

Co dělal AI systém Shakey?

Answer:

byl to multipurpose robot se spoustou senzorů, který využíval STRIPS planning a A* algoritmus.
Shakey - protože se při pohybu strasne otrasal

Question # 1 Type: Free answer, Maximum score: 1, Score:1

Uspořádejte následující AI systémy od nejstaršího po nejnovější:

DeepBlue
AlphaGo
Shakey
AlphaStar

Answer:

Shakey
DeepBlue
AlphaGo
AlphaStar

Question # 2 Type: Single answer, Maximum score: 1, Score:1

Kde se poprvé objevilo slovní spojení "artificial intelligence"?

- A v disertační práci Allana Turinga
- B v pracích Leonarda da Vinci
- C v návrhu konání workshopu v Dartmouth **Correct answer**
- Clear answer

Question # 2 Type: Single answer, Maximum score: 1, Score:1

Kde se poprvé objevilo slovní spojení "artificial intelligence"?

- A v disertační práci Allana Turinga
- B v návrhu konání workshopu v Dartmouth **Correct answer**
- C v pracích Leonarda da Vinci
- Clear answer

Question # 2 Type: Free answer, Maximum score: 1, Score:1

Co dělal AI systém ELIZA?

Answer:

chatbot
- AI, které dokáže vést smysluplnou konverzaci

Question # 3 Type: Free answer, Maximum score: 2, Score:1.5 

Co je Turingův test a k čemu slouží?

Answer:

Test, zda-li je AI nerozlišitelné od člověka
vpodstatě jeden člověk dostává odpovědi od dvou stran - od AI a od člověka. Pokud nedokáže rozlišit, která strana je AI a která člověk, tak AI prošlo Turingovým testem.

Teacher's comment:

Chybi zminka o testovani kvality AI

1. **Fail:** Stroj selže v přesvědčivém simulování lidského chování a myšlení a je snadno identifikovatelný jako stroj.
2. **Pass:** Stroj úspěšně přesvědčí člověka, že je skutečnou osobou, aspoň na určitou dobu nebo ve specifickém kontextu. Nicméně, po delším rozhovoru může být jeho nedostatek lidskosti odhalen.
3. **Pass with distinction:** Stroj přesvědčí člověka, že je lidská osoba, tak přesvědčivě, že je obtížné nebo nemožné jej odhalit jako stroj i po delším interakci.
4. **Superintelligence:** Hypotetická úroveň, kdy je stroj schopen nejen přesvědčivě simulovat lidské chování a myšlení, ale dokonce překonat lidskou inteligenci v mnoha oblastech.

Question # 3 Type: Free answer, Maximum score: 2, Score:2

Co je Turingův test a k čemu slouží?

Answer:

K rozlišení, zda se počítač / robot chová inteligentně nebo ne.

V jedné místnosti je sám člověk, ve zbylých dvou sami počítač a další člověk. Člověk v první místnosti pokládá otázky, které jsou předány počítači a druhému člověku, kteří na ně odpoví a jejich odpovědi jsou předány zpět prvnímu člověku. Pokud první člověk (Ten, co pokládal otázky.) není schopen rozlišit,

Question # 4 Type: Single answer, Maximum score: 1, Score:0

Kterou z následujících možností by zvolil racionální agent?

- A Dolar za každý hod férovou mincí při kterém padne panna, dokud nepadne první orel.
- B Počet dolarů daný jedním hodem standardní šestistěnnou kostkou. **Correct answer**
- C 3 dolary a 1 cent
- Clear answer

Question # 4 Type: Multiple answers, Maximum score: 1, Score:1

Kterou loterii by zvolil racionální agent?

- A 100% šance získat \$8 **Correct answer**
- B 30% šance získat \$20
- C 10% šance získat \$40
- D 2% šance získat \$400 **Correct answer**

Question # 5 Type: Free answer, Maximum score: 8, Score:8

Reprezentujte následující pravděpodobnostní distribuci jako Bayesovskou síť.

	toothache		¬toothache	
	catch	¬catch	catch	¬catch
cavity	0.108	0.012	0.072	0.008
¬cavity	0.016	0.064	0.144	0.576

Definujte všechny potřebné komponenty. Pro pravděpodobnostní tabulky definujte kterou pravděpodobnostní distribuci reprezentují a můžete vynechat hodnoty, které jsou doplňkem uvedených hodnot do 1.

Answer:

```
Toothache | T | F |
          0.2 0.8
```

```
      Toothache
```

```
Catch | T   | F
      T 0.124 0.216
      F 0.76  0.584
```

```
      | Cavity
```

```
Toothache\Catch| T   F
      T      T 0.108 0.016
      T      F 0.012 0.064
      F      T 0.072 0.144
      F      F 0.008 0.576
```

graf by vypadal cca takhle:

```
Toothache -> catch -> cavity
```

```
  \-----A (<- = šipka od toothache ke cavity)
```

Question # 6 Type: Single answer, Maximum score: 1

Kolik položek má sdružená pravděpodobnostní distribuce nad n binárními proměnnými?

A n^3

B 2^n **Correct answer**

C $n!$

D n

E n^2

Clear answer

Question # 8 Type: Multiple answers, Maximum score: 1, Score:1

Označte všechny operace, které musíme provést na STRIPS akci a , aby jsme vytvořili její relaxovanou variantu a^+ .

- A Smažeme její cenu (cost)
- B Smažeme pozitivní efekty (add effects)
- C Smažeme její negativní efekty (delete effects) **Correct answer**
- D Smažeme její předpoklady (preconditions)

Question # 8 Type: Multiple answers, Maximum score: 1

Které z následujících tvrzení o automatickém plánování jsou pravdivé?

- A Plánování vyžaduje úplný formální popis problému. **Correct answer**
- B Jeden z algoritmů běžně používaných v plánování je A^* . **Correct answer**
- C Korektní plán je typicky nalezen v čase polynomiálním ve velikosti popisu plánovacího problému.

Question # 9 Type: Single answer, Maximum score: 1, Score:0

Uvažujme POMDP, kde:

- Počet stavů = 15
- Počet možných pozorování = 4
- Počet akcí = 2

Jaký je počet α -vektorů po 2 krocích plné iterace hodnot (value iteration)?

- A $4 \cdot 4^2 = 64$
- B $15 \cdot 4 = 60$
- C $2 \cdot 2^4 = 32$ **Correct answer**
- D 4
- Clear answer

Question # 9 Type: Single answer, Maximum score: 1, Score:0

Uvažujme POMDP, kde:

- Počet stavů = 10
- Počet možných pozorování = 2
- Počet akcí = 4

Jaký je počet α -vektorů po 2 krocích plné iterace hodnot (value iteration)?

- A $10 \cdot 4 = 40$
- B 4
- C $4 \cdot 4^2 = 64$ **Correct answer**
- D $4 \cdot 2^4 = 64$
- Clear answer

Co je α -vektor?

Q 10

- A Lineární funkce vyjadřující očekávanou pravděpodobnost pro zvolený stav světa a danou strategii v závislosti na měnícím se beliefu agenta.
- B Lineární funkce vyjadřující očekávaný zisk pro danou strategii agenta v závislosti na měnícím se beliefu agenta. **Correct answer**
- C Lineární funkce vyjadřující očekávaný zisk pro iniciační belief agenta v závislosti na měnící se strategii.
- Clear answer

Question # 11 Type: Multiple answers, Maximum score: 1, Score:1

Vyznačte ta tvrzení o POMDP, která jsou pravdivá.

- A Belief je pravděpodobnostní distribuce nad stavy. **Correct answer**
- B α -vektor odpovídá očekávané hodnotě vždy jedné akce agenta.
- C Hodnota v iniciačním beliefu se v případě algoritmu iterace hodnot (value iteration) blíží optimální hodnotě shora.

Question # 9 Type: Single answer, Maximum score: 1, Score:0

Uvažujme POMDP, kde:

- Počet stavů = 15
- Počet možných pozorování = 4
- Počet akcí = 2

Jaký je počet α -vektorů po 2 krocích plné iterace hodnot (value iteration)?

- A 4
- B $15 \cdot 4 = 60$
- C $4 \cdot 4^2 = 64$
- D $2 \cdot 2^4 = 32$ **Correct answer**
- Clear answer

Question # 11 Type: Multiple answers, Maximum score: 1, Score:0

Vyznačte ta tvrzení o POMDP, která jsou pravdivá.

- A Belief je pravděpodobnostní distribuce nad možnými akcemi.
- B Množina všech dosažitelných pravděpodobnostních distribucí, které odpovídají možným belief stavům v problému, nezávisí na strategii agenta. **Correct answer**
- C POMDP odpovídají MDP se spojitým stavovým prostorem. **Correct answer**

Question # 12 Type: Free answer, Maximum score: 6, Score:5

Předpokládejme, že agent v multi-armed bandit problému pozoruje následující sekvenci odměn:

A	2	1	1.5			0		
B				1.5			0	1.5
C					1.5			

Jaká je pravděpodobnost výběru následující akce (a) greedy agentem, (b) ϵ -greedy agentem s explorační 0.1 a (c) UCB agentem s exploračním parametrem nastaveným na 4. Zapište všechny tři pravděpodobnostní distribuce.

Answer:

$$A = 4.5/4 = 1.125$$

$$B = 3/3 = 1$$

$$C = 1.5/1 = 1.5$$

$$a) p(A) = p(B) = 0, p(C) = 1$$

$$b) p(A) = p(B) = 0.05, p(C) = 0.9$$

c) nevím přesně, jak to přepsat do pravděpodobnosti, ale hodnoty jsou

$$A = 1.125 + 4\sqrt{\log 9/4}$$

$$B = 1 + 4\sqrt{\log 9/3}$$

$$C = 1.5 + 4\sqrt{\log 9/1}$$

z čehož největší je C, tedy UCB agent by zvolil C

a)

vybere vždy C, protože je tam největší avg gain

$$P(C) = 1$$

$$P(A) = 0$$

$$P(B) = 0$$

b)

vybere C s pravděpodobností 0.9, a náhodnou akci (A,B,C) s 0.1

$$P(C) = 0.933$$

$$P(A) = 0.033$$

$$P(B) = 0.033$$

Question # 14 Type: Multiple answers, Maximum score: 3, Score:3

Vyznačte které z následujících tvrzení jsou pravdivé.

- A Prohledávání do šířky vždy nalezne řešení, pokud toto existuje. **Correct answer**
- B Každá konzistentní heuristika je také přípustní heuristikou. **Correct answer**
- C Počet nesprávně seřazených čísel (tj. menší číslo je za větším) je přípustnou heuristikou pro 8-puzzle problém.
- D Prohledávání do hloubky má vyšší paměťové nároky než prohledávání do šířky.
- E Předpokládejme, že všechny ceny (costs) akcí jsou celá čísla a jsou striktně pozitivní. Funkce $h(x) = 1$ pro všechny stavy x , kromě cíle, je přípustnou heuristickou funkcí. **Correct answer**

Question # 15 Type: Multiple answers, Maximum score: 7, Score:4.2

Mějme problém s vysáváním místnosti reprezentované jako obdélník s $X \times Y$ čtvercovými políčky označenými jako (x, y) . Robotický vysavač musí vyčistit nečistoty na těchto polích. Množství nečistot je reprezentováno pomocí hodnoty od 0 do 100 u každého pole (x, y) . Zásobník robotického vysavače má kapacitu pro nečistoty m , tento zásobník lze vysypat během návštěvy základny umístěné na poli $(0, 0)$.

Počáteční místo pro robotický vysavač je $(0, 0)$ a můžete předpokládat, že robotický vysavač je prázdný. V jednom kroku se robotický vysavač může pohybovat z aktuálního pole na sousední ve 4 směrech (nahoru, dolů, vlevo, vpravo) nebo vysát 1 jednotku nečistot.

Cílem je najít nejkratší posloupnost pohybu, která vyčistí celou místnost a vrátit vysavač do základny na políčku $(0, 0)$.

Nechť n_{ij} je množství nečistot poli (i, j) . Jsou následující funkce přípustné heuristické funkce pro algoritmus A^* pro tento problém, nebo ne?

- A $h = \max_{i \in X, j \in Y} n_{ij}$ **Correct answer**
 - B $h = \max_{i \in X, j \in Y} \text{sgn}(n_{ij}) \cdot (\max\{i, j\})$ **Correct answer**
 - C $h = \sum_{i \in X, j \in Y} (i + j)$
 - D $h = \sum_{i \in X, j \in Y} n_{ij}$ **Correct answer**
 - E $h = \max_{i \in X, j \in Y} \text{sgn}(n_{ij}) \cdot (i + j)$ **Correct answer**
 - F $h = \sum_{i \in X, j \in Y} \text{sgn}(n_{ij}) \cdot (\max\{i, j\})$
 - G $h = \sum_{i \in X, j \in Y} \text{sgn}(n_{ij})$ **Correct answer**
-

Question # 15 Type: Multiple answers, Maximum score: 7, Score:7

Mějme problém s vysáváním místnosti reprezentované jako obdélník s $X \times Y$ čtvercovými políčky označenými (x, y) . k robotických vysavačů musí vyčistit nečistoty na těchto polích. Množství nečistot je reprezentováno pomocí hodnoty od 0 do 100 u každého pole (x, y) .

Zásobník každého robotického vysavače má kapacitu pro nečistoty m , tento zásobník lze vysypat během návštěvy základny umístěné na poli $(0, 0)$. Počáteční místo pro každý robotický vysavač je náhodně, ale pevně dané někde v místnosti. Můžete předpokládat, že každý robotický vysavač je prázdný.

V jednom kroku se každý robotický vysavač může pohybovat z aktuálního pole na sousední ve 4 směrech (nahoru, dolů, vlevo, vpravo) nebo vysát 1 jednotku nečistot. Všechny vysavače se hýbou simultánně a na jednom políčku může být nejvýše jeden vysavač (toto neplatí pro políčko $(0, 0)$, kde může být zároveň více vysavačů).

Cílem problému je najít nejkratší posloupnost pohybu, která vyčistí celou místnost a vrátit všechny vysavače do základny na políčku $(0, 0)$.

Nechť n_{ij} je množství nečistot poli (i, j) . Jsou následující funkce přípustné heuristické funkce pro algoritmus A^* pro tento problém nebo ne?

- A $h = \max_{i \in X, j \in Y} n_{ij}$ Correct answer
- B $h = \max_{i \in X, j \in Y} \frac{n_{ij}}{k} + (\max\{i, j\})$ Correct answer
- C $h = \sum_{i \in X, j \in Y} (i + j)$
- D $h = \sum_{i \in X, j \in Y} n_{ij}$
- E $h = \max_{i \in X, j \in Y} \frac{n_{ij}}{k}$ Correct answer
- F $h = \sum_{i \in X, j \in Y} \text{sgn}(n_{ij})$
- G $h = \max_{i \in X, j \in Y} \frac{n_{ij}}{k} \cdot (\max\{i, j\})$

Vyznačte které z následujících tvrzení jsou pravdivé.

Q 15

- A Prohledávání do hloubky má vyšší paměťové nároky než prohledávání do šířky.
- B Prohledávání do hloubky najde vždy optimální řešení i v nekonečném stavovém prostoru.
- C Každá konzistentní heuristika je také přípustní heuristikou. Correct answer
- D Předpokládejme, že všechny ceny (costs) akcí jsou celá čísla a jsou striktně pozitivní. Funkce $h(x) = 1$ pro všechny stavy x , kromě cíle, je přípustnou heuristickou funkcí. Correct answer
- E Počet nesprávně umístěných čísel je přípustnou heuristikou pro 8-puzzle problém. Correct answer

Question # 15 Type: Free answer, Maximum score: 4, Score:4

Uvažte následující CSP

- Proměnné: x_1, x_2, x_3, x_4
- Domény pro každou proměnnou: $\{0, 1, 2, 3\}$
- Omezení:

$$x_1 \neq x_2$$

$$x_1 = x_3 + 1$$

$$x_2 < x_3$$

$$x_4 = 2 \cdot x_2$$

Zapište domény pro každou proměnnou po provedení kompletního AC-3 algoritmu:

- $x_1 =$
- $x_2 =$
- $x_3 =$
- $x_4 =$

Answer:

$$X_1 = \{2, 3\}$$

$$X_2 = \{0, 1\}$$

$$X_3 = \{1, 2\}$$

$$X_4 = \{0, 2\}$$

Question # 16 Type: Multiple answers, Maximum score: 3, Score:3

Vyznačte které z následujících tvrzení jsou pravdivé.

- A Prohledávání do hloubky má vyšší paměťové nároky než prohledávání do šířky.
- B Počet nesprávně umístěných čísel je přípustnou heuristikou pro 8-puzzle problém. **Correct answer**
- C Předpokládejme, že všechny ceny (costs) akcí jsou celá čísla a jsou striktně pozitivní. Funkce $h(x) = 1$ pro všechny stavy x , kromě cíle, je přípustnou heuristickou funkcí. **Correct answer**
- D Každá konzistentní heuristika je také přípustní heuristikou. **Correct answer**
- E Prohledávání do hloubky najde vždy optimální řešení i v nekonečném stavovém prostoru.

Question # 16 Type: Multiple answers, Maximum score: 1, Score:1

Algoritmus CSP

- A nemusí vždy najít řešení **Correct answer**
- B hledá nejkratší cestu ze startu do cíle
- C hledá přípustné přiřazení hodnot proměnných z jejich domén **Correct answer**

Question # 16 Type: Multiple answers, Maximum score: 3, Score:1.5

Vyznačte které z následujících tvrzení jsou pravdivé.

- A Počet nesprávně seřazených čísel (tj. menší číslo je za větším) je přípustnou heuristikou pro 8-puzzle problém.
- B Každá přípustná heuristika je také konzistentní heuristikou.
- C Obousměrné (bidirectional) informované prohledávání vždy expanduje méně uzlů než jednosměrné prohledávání.
- D Počet nesprávně umístěných čísel je přípustnou heuristikou pro 8-puzzle problém. **Correct answer**
- E Za předpokladu, že existuje řešení a všechny ceny (costs) za akce se rovnají 2, prohledávání do šířky vždy najde optimální řešení i v nekonečném stavovém prostoru. **Correct answer**

Question # 16 Type: Multiple answers, Maximum score: 3, Score:3

Vyznačte které z následujících tvrzení jsou pravdivé.

- A Každá přípustná heuristika je také konzistentní heuristikou.
- B Za předpokladu, že existuje řešení a všechny ceny (costs) za akce se rovnají 2, prohledávání do šířky vždy najde optimální řešení i v nekonečném stavovém prostoru. **Correct answer**
- C Počet nesprávně seřazených čísel (tj. menší číslo je za větším) je přípustnou heuristikou pro 8-puzzle problém.
- D Obousměrné (bidirectional) informované prohledávání vždy expanduje méně uzlů než jednosměrné prohledávání.

Question # 17 Type: Free answer, Maximum score: 4, Score:4

Uvažte následující CSP

- Proměnné: x_1, x_2, x_3, x_4
- Domény pro každou proměnnou: $\{0, 1, 2, 3\}$
- Omezení:

$$x_1 \neq x_2$$

$$x_1 = x_3 + 1$$

$$x_2 < x_3$$

$$x_4 = 2 \cdot x_2$$

Zapište domény pro každou proměnnou po provedení kompletního AC-3 algoritmu:

- $x_1 =$
- $x_2 =$
- $x_3 =$
- $x_4 =$

Answer:

$X_1 = \{2, 3\}$
 $X_2 = \{0, 1\}$
 $X_3 = \{1, 2\}$
 $X_4 = \{0, 2\}$

Question # 17 Type: Free answer, Maximum score: 4, Score:3.5

Uvažte následující CSP

- Proměnné: x_1, x_2, x_3, x_4
- Domény pro každou proměnnou: $\{0, 1, 2, 3\}$
- Omezení:

$$x_1 > x_2$$

$$x_2 > x_3$$

$$x_4 = 1 + x_2$$

Zapište domény pro každou proměnnou po provedení kompletního AC-3 algoritmu:

- $x_1 =$
- $x_2 =$
- $x_3 =$
- $x_4 =$

Answer:

$x_1 = \{2, 3\}$
 $x_2 = \{1, 2\}$
 $x_3 = \{0\}$
 $x_4 = \{2, 3\}$

Teacher's comment:

take hodnota $x_3=1$ je možná

Question # 18 Type: Multiple answers, Maximum score: 1, Score:1

Která z následujících tvrzení jsou pravdivá?

- A Negamax prohledá stejné uzly herního stromu jako minimax. **Correct answer**
- B Alpha-beta může prohledat méně různých uzlů herního stromu jako Negascout.
- C Negascout navštíví uzly v herním stromě nejvýše dvakrát.

Question # 18 Type: Single answer, Maximum score: 1, Score:1

Je následující tvrzení pravdivé?

Pro řešení CSP můžeme využít i algoritmus A^* .

- A Ano **Correct answer**
- B Ne
- Clear answer

Question # 20 Type: Free answer, Maximum score: 4, Score:4

Předpokládejme, že agent v multi-armed bandit problému pozoruje následující sekvenci odměn:

A	1		0	0.5	2	1.5		
B		0.5						1.5
C							0	

Jaká je pravděpodobnost výběru následující akce (a) greedy agentem, (b) *epsilon*-greedy agentem s explorační 0.1 a (c) UCB agentem s exploračním parametrem nastaveným na 4. Zapište všechny tři pravděpodobnostní distribuce.

Answer:

$$A = (1+0+0.5+2+1.5) / 5 = 1$$

$$B = (0.5+1.5) * 2 = 1$$

$$C = 0$$

a)

$$P(A) = 0.5$$

$$P(B) = 0.5$$

$$P(C) = 0$$

b)

$$P(A) = 0.5 * 0.9 + 0.1 / 3 = 0.4833$$

$$P(B) = 0.5 * 0.9 + 0.1 / 3 = 0.4833$$

$$P(C) = 0 + 0.1 / 3 = 0.0333$$

c)

$$P(A) = 0$$

$$P(B) = 0$$

$$P(C) = 1$$

Question # 18 Type: Single answer, Maximum score: 1, Score:1

Čemu odpovídá hrana v CSP prohledávacím stromě (search tree)?

A Proměnné CSP problému.

B Hodnotě přiřazené proměnné (uzlu, ze kterého hrana vychází) **Correct answer**

C Množině dostupných hodnot pro proměnnou reprezentovanou uzlem, ze kterého hrana vychází.

Clear answer

Question # 19 Type: Single answer, Maximum score: 1, Score:1

Čemu odpovídá uzel v CSP prohledávacím stromě (search tree)?

- A Hodnotě přiřazené některé z proměnných CSP problému.
- B Proměnné CSP problému **Correct answer**
- C Množině nepřirazených proměnných.
- Clear answer

Question # 19 Type: Multiple answers, Maximum score: 1, Score:1

Které z následujících tvrzení o algoritmu AlphaGo jsou pravdivé?

- A Algoritmus AlphaGo nebyl zatím v hraní Go překonán žádným člověkem ani jiným AI systémem.
- B Algoritmus používal tři různé policy a jedna z nich aproximuje lidskou hru. **Correct answer**
- C Algoritmus porazil profesionálního hráče ve hře Go **Correct answer**

Question # 19 Type: Single answer, Maximum score: 1, Score:0

Je následující tvrzení pravdivé?

AC-3 algoritmus je spuštěn v rámci řešení CSP prohledáváním nejvýše jednou.

- A Ano
- B Ne **Correct answer**

Question # 20 Type: Single answer, Maximum score: 1, Score:0

Uvažujme standardní alfa-beta prořezávání s upraveným počátečním alfa-beta intervalem nastaveným na $[w, x]$. Algoritmus vrátí hodnotu $y < w$. Které z následujících tvrzení je pravdivé?

- A Přesná hodnota hry není známa, ale je menší nebo rovna hodnotě w .
- B Hodnota hry je známá a je rovna hodnotě w .
- C Přesná hodnota hry není známa, ale je menší nebo rovna hodnotě y . **Correct answer**
- D Hodnota hry je známá a je rovna hodnotě y .
- Clear answer

Question # 22 Type: Multiple answers, Maximum score: 1, Score:0.5

Které z následujících tvrzení o algoritmu AlphaGo jsou pravdivé?

- A Algoritmus AlphaGo nebyl zatím v hraní Go překonán žádným člověkem ani jiným AI systémem.
- B Algoritmus používal tři různé policy a jedna z nich aproximuje lidskou hru. **Correct answer**
- C Algoritmus porazil profesionálního hráče ve hře Go **Correct answer**

Question # 23 Type: Multiple answers, Maximum score: 1, Score:0

Co musíme nezbytně udělat pro transformaci epizodické úlohy do ekvivalentní pokračující úlohy ve zpětnovazebném učení?

- A Nastavit diskontní míru $\gamma \in (0, 1)$.
- B Nahradit terminální stavy absorbujícími stavy (i.e., self-loops). **Correct answer**
- C Přeskálovat odměny tak, aby byly menší než jedna.

Question # 24 Type: Multiple answers, Maximum score: 1, Score:0

Předpokládejme MDP s A akcemi a S stavy. Které z následujících tvrzení jsou pravda?

- A Časová složitost u jednoho kroku Q-learning je $O(S^2 * A)$.
- B Časová složitost u jednoho kroku Q-learning je $O(A)$. **Correct answer**
- C Prostorová složitost u tabulkového Q-learning je $O(S)$.
- D Prostorová složitost u tabulkového Q-learning je $O(S^2 * A)$.
- E Prostorová složitost u tabulkového Q-learning je $O(S * A)$. **Correct answer**
- F Časová složitost u jednoho kroku Q-learning je $O(S * A)$.
- G Prostorová složitost u tabulkového Q-learning je $O(S * A^2)$.
- I Časová složitost u jednoho kroku Q-learning je $O(S * A^2)$.

Question # 24 Type: Multiple answers, Maximum score: 1, Score:0

Jaké vstupy musí mít agent ve zpětnovazebním učení, aby mohl najít optimální strategii (optimal policy)?

- A Množina všech možných stavů
- B Přesný popis odměnových dynamik
- C Funkce hodnot (Value function)
- D Množina všech dostupných akcí Correct answer
- E Simulátor prostředí Correct answer
- F Počet možných stavů
- G Konečná délka horizontu
- I Přesný popis přechodových dynamik

Question # 18 Type: Multiple answers, Maximum score: 1, Score:0.5

Která z následujících tvrzení jsou pravdivá?

- A UCT může prohledat i uzly, které alpha-beta nebude prohledávat. Correct answer
- B Negascout vždy navštíví méně uzlů než minimax.
- C Negamax navštíví každý uzel v herním stromě nejvýše raz. Correct answer

Question # 16 Type: Multiple answers, Maximum score: 1, Score:0.5

Algoritmus CSP

- A hledá přípustné přiřazení hodnot proměných z jejich domén Correct answer
- B hledá nejkratší cestu ze startu do cíle
- C nemusí vždy najít řešení Correct answer

Question # 11 Type: Multiple answers, Maximum score: 1, Score:0

Vyznačte ta tvrzení o POMDP, která jsou pravdivá.

- A Belief je pravděpodobnostní distribuce nad stavy. Correct answer
- B α -vektor odpovídá očekávané hodnotě vždy jedné akce agenta.
- C Hodnota v iniciálním beliefu se v případě algoritmu iterace hodnot (value iteration) blíží optimální hodnotě shora.

TODO:

Question # 1 Type: Free answer, Maximum score: 1, Score:0

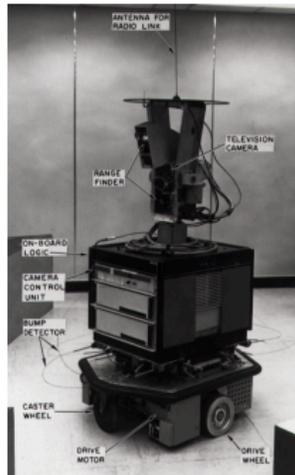
Co dělal AI systém Shakey?

Answer:

General-purpose mobile robot developed at Standord Research Institute

Combined navigation, computer vision, natural language processing, etc.

Programmed primarily in **Lisp**. General purpose planning using **STRIPS**. The project also introduced the **A*** algorithm and generalized Hough transform.



Question # 3 Type: Free answer, Maximum score: 2, Score:0.5

Popište krátce co znamená pojem "AI Winter" a proč k němu dochází.

Answer:

1974-1980: The First AI Winter

Large drop of funding and interest in AI

- Initial hype did not deliver impactful applications
- UK funding cut due to Lighthill report
"In no part of the field have the discoveries made so far produced the major impact that was then promised"
- DARPA focussed on "mission-oriented direct research"
- Negative results of Minsky & Papert on perceptrons decreased popularity of connectionism



1987-1993: The Second AI Winter

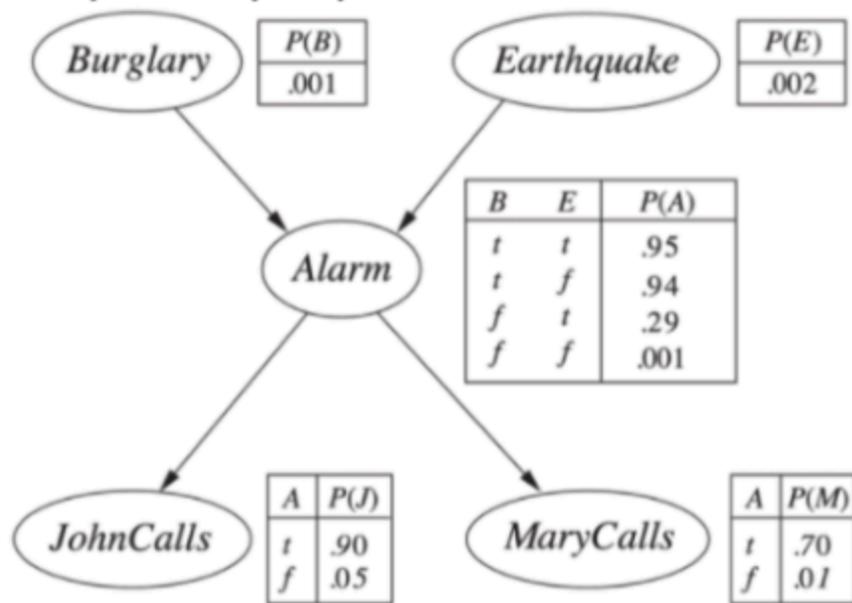
Large drop of funding and interest in AI

- The expert systems did not meet the expectations, they were too expensive to maintain
- Drop of market for specialized AI HW as general purpose HW became more powerful
- High expectations / promises of many startups failed



Question # 5 Type: Free answer, Maximum score: 8, Score:0

Uvažujme následující Bayesovskou síť.

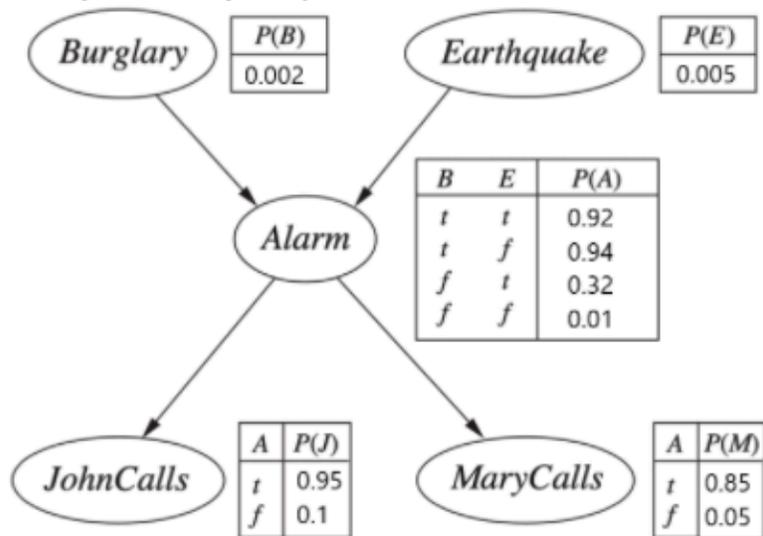


Jaká je pravděpodobnost, že oba John i Mary zavolají, když víme, že nastalo zemětřesením (*Earthquake*). Popište postup, jak jste k výsledku došli.

Answer:

Question # 5 Type: Free answer, Maximum score: 8, Score:0.5

Uvažujme následující Bayesovskou síť.



Jaká je pravděpodobnost, že alespoň jeden ze sousedů (John nebo Mary) zavolá, když víme, že nastalo vloupání (*Burglary*). Popište postup, jak jste k výsledku došli.

Answer:

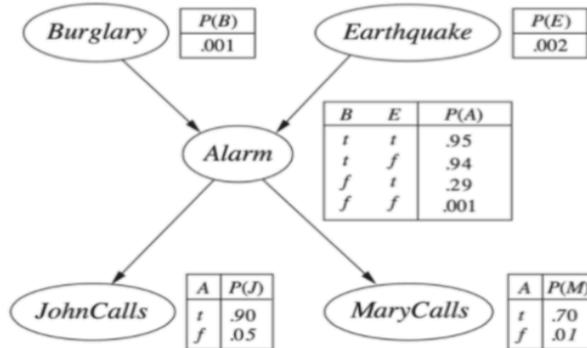
<https://www.youtube.com/watch?v=hEZjPZ-Ze0A> part 1

<https://www.youtube.com/watch?v=iz7KI2gcmlk> part 2

pomuze pochopit jak se pocita

Question # 5 Type: Free answer, Maximum score: 8, Score:8

Uvažujme následující Bayesovskou síť.



Jaká je pravděpodobnost, že alespoň jeden ze sousedů (John nebo Mary) zavolá, když víme, že nastalo vloupání (Burglary). Popište postup, jak jste k výsledku došli.

Answer:

$$P(j \text{ or } m | b) = P(j | b) + P(m | b) - P(j \text{ and } m | b)$$

1)

$$P(j, m | b) = (1 / P(b)) * \sum_{a \in \{t, f\}} (\sum_{e \in \{t, f\}} P(j | a) * P(m | a) * P(a | b, e) * P(b) * P(e)) =$$

$$(1 / P(b)) * (P(j | a) * P(m | a) * P(a | b, e) * P(b) * P(e) + P(j | a) * P(m | a) * P(a | b, -e) * P(b) * P(-e) + P(j | -a) * P(m | -a) * P(-a | b, e) * P(b) * P(e) + P(j | -a) * P(m | -a) * P(-a | b, -e) * P(b) * P(-e)) =$$

$$(1 / .001) * (.90 * .70 * .95 * .001 * .002 + .90 * .70 * .94 * .001 * .998 + .05 * .01 * .05 * .001 * .002 + .05 * .01 * .06 * .001 * .998) = 0.5922$$

$$2) P(j | b) = (1 / P(b)) * \sum_{m \in \{t, f\}} (\sum_{a \in \{t, f\}} (\sum_{e \in \{t, f\}} P(j | a) * P(m | a) * P(a | b, e) * P(b) * P(e))) =$$

$$(1 / P(b)) * (P(j | a) * P(m | a) * P(a | b, e) * P(b) * P(e) + P(j | a) * P(m | a) * P(a | b, -e) * P(b) * P(-e) + P(j | -a) * P(m | -a) * P(-a | b, e) * P(b) * P(e) + P(j | -a) * P(m | -a) * P(-a | b, -e) * P(b) * P(-e) + P(j | a) * P(-m | a) * P(a | b, e) * P(b) * P(e) + P(j | a) * P(-m | a) * P(a | b, -e) * P(b) * P(-e) + P(j | -a) * P(-m | -a) * P(-a | b, e) * P(b) * P(e) + P(j | -a) * P(-m | -a) * P(-a | b, -e) * P(b) * P(-e)) =$$

$$(1 / .001) * (.90 * .70 * .95 * .001 * .002 + .90 * .70 * .94 * .001 * .998 + .05 * .01 * .05 * .001 * .002 + .05 * .01 * .06 * .001 * .998 + .90 * (1-.70) * .95 * .001 * .002 + .90 * (1-.70) * .94 * .001 * .998 + .05 * (1-.01) * .05 * .001 * .002 + .05 * (1-.01) * .06 * .001 * .998) = 0.8490$$

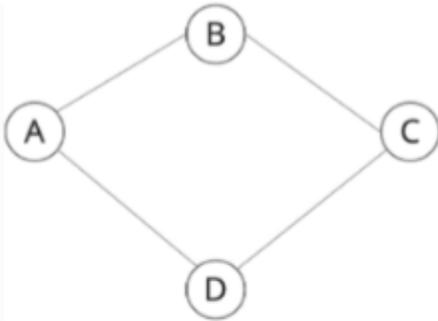
$$3) P(m | b) = (1 / P(b)) * \sum_{j \in \{t, f\}} (\sum_{a \in \{t, f\}} (\sum_{e \in \{t, f\}} P(j | a) * P(m | a) * P(a | b, e) * P(b) * P(e))) =$$

$$(1 / P(b)) * (P(j | a) * P(m | a) * P(a | b, e) * P(b) * P(e) + P(j | a) * P(m | a) * P(a | b, -e) * P(b) * P(-e) + P(j | -a) * P(m | -a) * P(-a | b, e) * P(b) * P(e) + P(j | -a) * P(m | -a) * P(-a | b, -e) * P(b) * P(-e) + P(-j | a) * P(m | a) * P(a | b, e) * P(b) * P(e) + P(-j | a) * P(m | a) * P(a | b, -e) * P(b) * P(-e) + P(-j | -a) * P(m | -a) * P(-a | b, e) * P(b) * P(e) + P(-j | -a) * P(m | -a) * P(-a | b, -e) * P(b) * P(-e)) =$$

$$(1 / .001) * (.90 * .70 * .95 * .001 * .002 + .90 * .70 * .94 * .001 * .998 + .05 * .01 * .05 * .001 * .002 + .05 * .01 * .06 * .001 * .998 + (1-.90) * .70 * .95 * .001 * .002 + (1-.90) * .70 * .94 * .001 * .998 + (1-.05) * .01 * .05 * .001 * .002 + (1-.05) * .01 * .06 * .001 * .998) = 0.6586$$

$$4) P(j \text{ or } m | b) = 0.8490 + 0.6586 - 0.5922 = 0.9154$$

Question # 7 Type: Free answer, Maximum score: 6, Score:0



Uvažujme města A, B, C, D propojeny cestami na obrázku. Ve městě A je zásilka. Ve městě C je auto. Auto se může přesunout do sousedícího města, naložit nebo vyložit zásilku.

Pomocí formalismu STRIPS popište:

- (a) tuto iniciální situaci
- (b) akce naložení zásilky, vyložení zásilky a přesunu auta

Answer:

a)
 $P = \{$
 $car_at_a/b/c/d$
 $box_at_a/b/c/d$
 box_loaded
 $adjacent_a_b, adjacent_b_c, adjacent_c_d, adjacent_d_a$
 $\}$
 $I = \{car_at_c, box_at_a\}$
 $G = \{\}$
 b)
 $load = \langle \{cat_at_x, box_at_x\}, \{box_loaded\}, \{box_at_x\} \rangle$
 $unload = \langle \{cat_at_x, box_loaded\}, \{box_at_x\}, \{box_loaded\} \rangle$
 $move_x_y = \langle cat_at_x, adjacent_x_y, \{cat_at_y\}, \{cat_at_x\} \rangle$
 (Misto x a y by to asi melo byt pro vsechny a,b,c,d a kombinace)
 (and yes I realize car turned into a cat somehow)

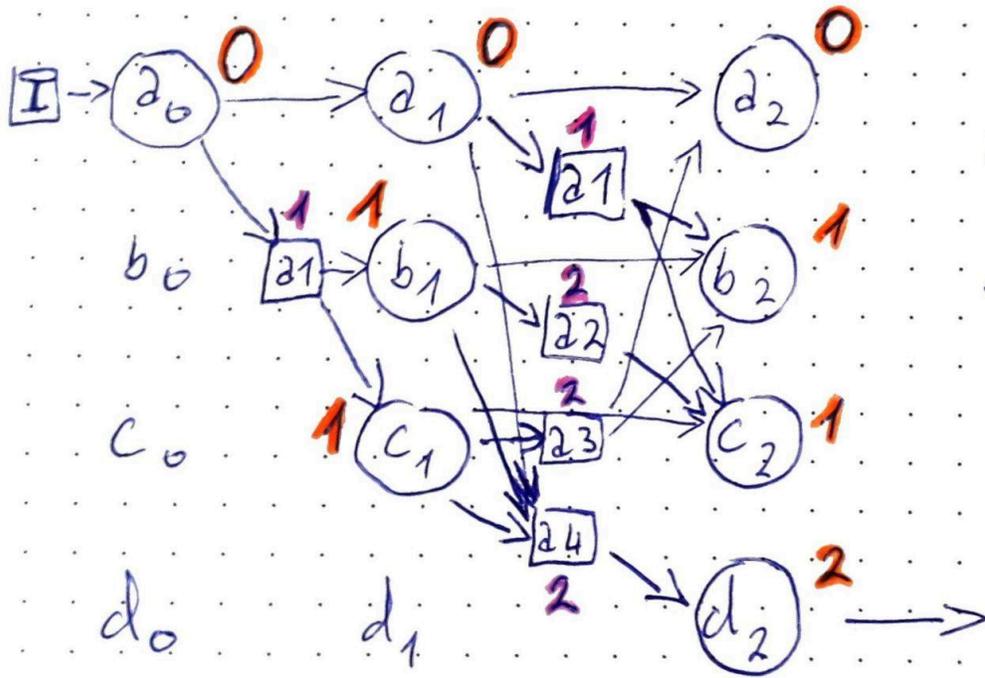
Question # 7 Type: Free answer, Maximum score: 6, Score:0

Uvažujte STRIPS problém s predikáty a, b, c, d , iniciálním stavem $I = \{a\}$, akcemi

- $a_1 = \langle \{a\}, \{b, c\}, \emptyset \rangle$
 - $a_2 = \langle \{b\}, \{c\}, \emptyset \rangle$
 - $a_3 = \langle \{c\}, \{a, b\}, \emptyset \rangle$
 - $a_4 = \langle \{a, b, c\}, \{d\}, \emptyset \rangle$
- a cílovým stavem $G = \{d\}$.

- (a) Jaká je hodnota heuristiky h_{max} v iniciálním stavu?
- (b) Jaká je hodnota každé z akcí během výpočtu h_{max} v iniciálním stavu?

Answer:



a) Hodnota je 2

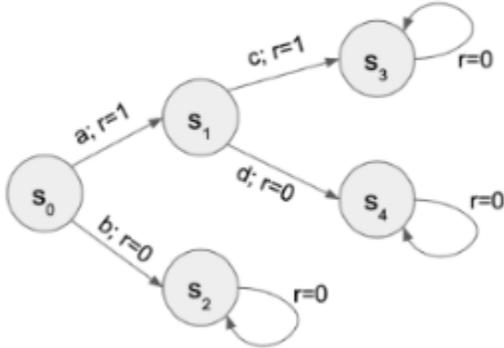
b) Hodnoty akcí jsou

	a_1	a_2	a_3	a_4
	1	2	2	2

Viz https://cw.fel.cvut.cz/old/_media/courses/a4m33pah/03-relaxation.pdf

Question # 13 Type: Free answer, Maximum score: 5, Score:1

Vypočítejte přesné hodnoty optimálních value funkcí ve všech neabsorbujících stavech a pro všechny akce v následujícím MDP. Předpokládejme diskontní míru (discount factor) 0.5.



Answer:

$$V(s_0) = \max(1 + 0.5 \cdot V(s_1), 0 + 0.5 \cdot V(s_2)) = \max(1.5, 0) = 1.5$$

$$V(s_1) = \max(1 + 0.5 \cdot V(s_3), 0 + 0.5 \cdot V(s_4)) = \max(1, 0) = 1$$

$$V(s_2) = 0$$

$$V(s_3) = 0$$

$$V(s_4) = 0$$

(maybe)

$$S_1 = 0.5 \cdot (1 + 0.5 \cdot 0) + 0.5 \cdot (0.5 + 0.5 \cdot 0.5) = 0.375$$

$$S_0 = 0.5 \cdot (1 + 0.5 \cdot 0.375) + 0.5 \cdot (0 + 0.5 \cdot 0) = 0.59375$$

$$V(s) = \max(R(a) + \alpha \cdot V(s'))$$

Question # 14 Type: Free answer, Maximum score: 5, Score:0

Předpokládejme deterministický gridworld, kde se robot může pohybovat jen doleva (L) nebo doprava (R); pohyb nic nestojí a robot získá odměnu 4 pokud opustí herní plochu; diskontní míra (discount factor) je 1.

Předpokládejme, že iničiální value funkce je

2	1	0	1	0
---	---	---	---	---

.

- (a) Jaké jsou hodnoty value funkce po dvou krocích value iteration algoritmu?
- (b) Jaké strategie by se měl držet agent s value funkcí z bodu (a) ve stavu nejvíce vlevo a proč?
- (c) Jak byste změnili tuto strategii aby fungovala lépe při praktické aplikaci v robotice?

Answer:

a)
přidám si krajní stavy
inicialni value funkce: 4 | 2 1 0 1 0 | 4
po prvním kroku: 4 | 4 2 1 0 4 | 4
po druhém kroku: 4 | 4 4 2 4 | 4

b) **just go left**
vzhledem k tomu, že není žádná penalizace (negativní reward) za pohyb, může agent chodit po herní ploše libovolný počet kroků a poté z něj odejít a získat reward 4

c)
přidal bych negativní reward za pohyb

Teacher's comment:

b) ano, ale to není odpověď na otázku

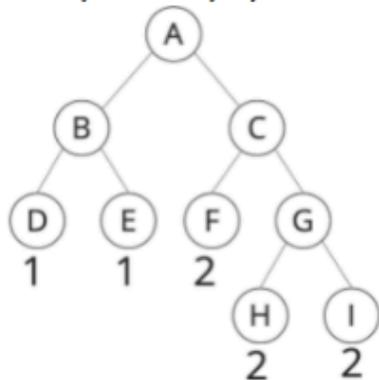
Question # 18 Type: Single answer, Maximum score: 1

CSP prohledávání sématicky odpovídá prohledávání

- A** do šířky
- B** do hloubky
- C** do hloubky s iterativním prodlužováním horizontu (iterative deepening)
- Clear answer**

Question # 17 Type: Free answer, Maximum score: 6, Score:0.5

Uvažujte následující jednohráčovou hru (čísla pod listy značí odměnu hráče v daném stavu):



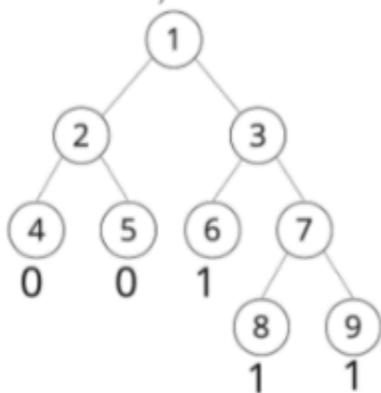
Uvažujme algoritmus UCT s explorační konstantou $c = 2$ a předpokládejme, že při rovnosti hodnot algoritmus preferuje levou akci (neplatí pro simulaci).

1. Specifikujte postavenou část stromu po 5 iteracích algoritmu.
2. Ve které iteraci UCT algoritmu se garantovaně navštíví uzel D? Uveďte pořadí iterace i postup výpočtu.

UCT = UCB on Trees :-D

Question # 21 Type: Free answer, Maximum score: 6, Score:2

Uvažujte následující jednohráčovou hru (čísla pod listy značí odměnu hráče v daném stavu):



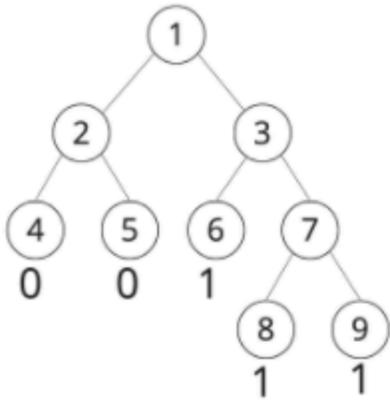
Kolik iterací musí algoritmus UCT s explorační konstantou 2 vykonat, aby bylo zaručené, že navštíví stav 4? Napište konkrétní číslo a pak jeho zdůvodnění.

Answer:

Zprvu UCT nahodně vybírá 2 nebo 3, a pak vybírá jiné číslo, aby mohl pokračovat na další hloubku. Tady máme 2 iterace. Pak máme aktualizovat skóre 2 a 3: pro 2 dostaneme 0/1, pro 3 - 1/1. Pak UCT ve "selection phase" vybere 3, protože má vyšší skóre. Opakujeme algoritmus pro uzly 6 a 7 a dostaneme:
3: 3/3
6: 1/1
7: 1/1
Pak UCT opakuje se v uzlech 8 a 9 a jen tady může explorační konstanta 2 vyvolat návštěvu uzlu 4.

Question # 21 Type: Free answer, Maximum score: 6, Score:0

Uvažujte následující jednohráčovou hru (čísla pod listy značí odměnu hráče v daném stavu):



Kolik iterací musí algoritmus UCT s explorační 2 vykonat, aby bylo zaručené, že navštíví stav 4? Napište konkrétní číslo a pak jeho zdůvodnění.

Answer:

Question # 22 Type: Free answer, Maximum score: 1, Score:1

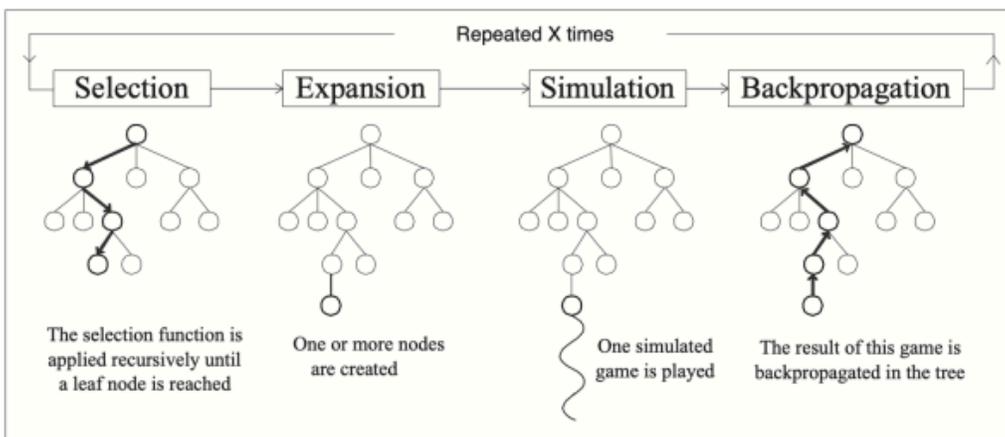
Setříděte fáze Monte Carlo prohledávání stromu podle počtu různých stavů hry, které jsou během každé fáze typicky navštíveny od nejmenšího.

Answer:

Monte Carlo Tree Search

Idea:

- 1 Instead of evaluation function, use random roll-outs (simulations) of the rest of the game
- 2 Store detailed statistics only in relevant parts of the game tree



(Image from Chaslot et al. 2007)

Question # 22 Type: Single answer, Maximum score: 1

Který selekční algoritmus by jste použili v Monte Carlo prohledávání stromu s velkým (případně nekonečným) počtem akcí?

UCT -> score = reward/total_selections + c(ln(times_this_one_visited)/total_selections^{-0,5})

Question # 22 Type: Free answer, Maximum score: 4, Score:0.5

Uvažujme problém plánování pohybu robotické ruky v prostoru (např. při manipulaci s předměty) s cílem vykonat co možná nejefektivnější pohyb bez nechtěné kolize s předmětem.

1. Navrhněte formální reprezentaci, kterou by bylo možné použít pro prohledávání prostoru.
2. Navrhněte netriviální přípustnou heuristiku a zdůvodněte proč je přípustná.

Answer:

1)

{úhelkloubu1, rotacekloubu1, délkapaže1, ... pro další paže a klouby}

tímto bychom byli schopni vždy zjistit přesnou polohu jednotlivých paží a kloubů

za tuto otázku jsem dostala 3 body a napsala jsem že muze to byt MDP s reprezentaci polohy a ostatnich objektu a o heuristice jsem rekla ze muzeme to delat podle pythagorove vety (edited)

Question # 22 Type: Free answer, Maximum score: 4, Score:1

Uvažujme problém nalezení nejrychlejšího spojení v MHD za předpokladu, že můžete použít různé způsoby dopravy (tramvaj, autobus, pěší přesun mezi zastávky do vzdálenosti 10 minut, etc.)

1. Navrhněte formální reprezentaci, kterou by bylo možné použít pro prohledávání prostoru.
2. Navrhněte netriviální přípustnou heuristiku a zdůvodněte proč je přípustná.

Answer:

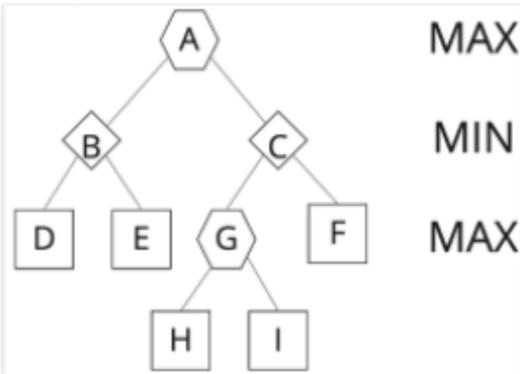
Question # 23 Type: Free answer, Maximum score: 1, Score:0

Vypište tři metody, založené na různém principu, které podporují exploraci při zpětnovazebném učení.

Answer:

UCB, epsilon greedy,

Uvažujte dvouhřáčkovou hru na obrázku:



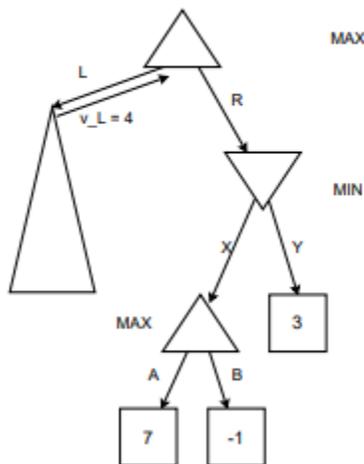
Napište utility hodnoty pro listy stromu (uzly D, E, H, I, F) tak, aby algoritmus Alpha-Beta nic neprořezal a algoritmus Negascout prořezal některé z uzlů.

5 6 7 8 4 - nega skipne I

5 4 7 -1 3 - nega skipne I

NegaScout vs. Alpha-Beta

NegaScout can evaluate some nodes multiple times. At the same time, it can never evaluate more different histories and can be 10 – 20% faster than Alpha-Beta pruning.



Assume that after evaluating subtree L , the value v_L equals to 4. The evaluation of R starts with $[\alpha_R, \beta_R] = [4, 5] = [\alpha_{RX}, \beta_{RX}]$. When the algorithm evaluates A , it updates α_{RX} to 7. Since β_{RX} has been manually set to 5, it holds that $\beta_{RX} < \alpha_{RX}$ and the algorithm prunes out evaluation of action B .

Note that this would not happen in Alpha-Beta pruning.

Question # 16 Type: Multiple answers, Maximum score: 1, Score:0

Která z následujících tvrzení jsou pravdivá?

- A Algoritmus Monte Carlo Tree Search staví vybalancovaný (z hlediska teorie grafů) herní strom.
- B Negascout může navštívit některé uzly v herním stromě opakovaně. Correct answer
- C Alpha-beta prořezávání vždy navštíví méně uzlů herního stromu než minimax.

Question # 12 Type: Multiple answers, Maximum score: 3, Score:0

Vyznačte která z následujících tvrzení jsou pravdivé.

- A Prohledávání do šířky (BFS) vždy expanduje méně uzlů než prohledávání do hloubky (DFS).
- B A* musí používat konzistentní heuristiku, aby našel optimální řešení.
- C Za předpokladu, že existuje řešení a všechny akce mají striktně pozitivní cenu (cost), iterativní prohledávání do hloubky IDA* s $h(x) = 0$ pro všechny stavy vždy najde optimální řešení i v nekonečném stavovém prostoru. Correct answer
- D Počet nesprávně umístěných čísel je přípustnou heuristikou pro 8-puzzle problém. Correct answer
- E Manhattanská vzdálenost je přípustná heuristika pro plánování robota v bludišti (na mřížce) pokud se robot může pohybovat všemi 8 směry.