

## B4B36ZUI – Základy umělé inteligence / Introduction to Artificial Intelligence – 13. 6. 2018

O1	O2	O3	O4	O5	Total (50)

**Instrukce:** Na vypracování máte 150 min, můžete použít vlastní poznámky v podobě ručně popsaného listu A4. Použití počítače nebo mobilního telefonu není povoleno. V otázkách true/false zakroužkujte jednu z možností. Pokud si odpovědi nejste jistí, zdůvodněte ji.

### Question 1 (10 points) Search

Mějme zaplavenou oblast reprezentovanou obdélníkovou mřížkou  $X \times Y$  políčkami označenými jako  $b_{xy}$ . V oblasti jsou lidé, kteří potřebují být zachráněni a transportováni do bezpečí (pole  $b_{\bullet\bullet}$ ), které je mimo mřížky  $X \times Y$  a je přístupné pouze z pole  $b_{00}$ . Na každém poli  $b_{xy}$  může být od 0 do  $n$  osob. Záchranáři mají jeden vrtulník, který může přenášet  $m$  lidí. Počáteční místo pro vrtulník je  $b_{\bullet\bullet}$  a můžete předpokládat, že vrtulník má nekonečné palivo. V jednom tahu se vrtulník může pohybovat z aktuálního pole do přilehlého pole v 8 směrech (nahoru, dolů, vlevo, vpravo, 4 diagonální pohyby) nebo naložit/vyložit 1 osobu. Cílem je najít nejkratší posloupnost, která přenesou všechny lidi do bezpečí (tj. všichni lidé jsou na poli  $b_{\bullet\bullet}$  a vrtulník je prázdný).

(a) (4 body) Zapište formální definici problému tak, aby mohl být vyhledávací algoritmus použitý k nalezení řešení. Zaznamenejte reprezentaci stavů a akcí.

(b) (3 body) Necht  $n_{ij}$  je počet lidí na poli  $b_{ij}$ , kteří nejsou ve vrtulníku. Jsou následující funkce přípustné heuristické funkce pro algoritmus  $A^*$  pro tento problém, nebo ne?

(YES/NO)  $h = \sum_{i \in X, j \in Y} n_{ij}$

(YES/NO)  $h = \sum_{i \in X, j \in Y} \frac{n_{ij}}{m}$

(YES/NO)  $h = \sum_{i \in X, j \in Y} \text{sgn}(n_{ij}) \cdot (i + j)$

(YES/NO)  $h = \sum_{i \in X, j \in Y} n_{ij} \cdot (i + j)$

(YES/NO)  $h = \sum_{i \in X, j \in Y} n_{ij} \cdot (\max\{i, j\})$

(YES/NO)  $h = \sum_{i \in X, j \in Y} [\text{sgn}(n_{ij}) \cdot (\max\{i, j\}) + n_{ij}]$

(c) (3 body) Označte následující obecná tvrzení jako pravdivá nebo nepravdivá:

(TRUE/FALSE)  $h(x) = 1$  pro všechny stavy  $x$  je přípustná heuristická funkce.

(TRUE/FALSE) Prohledávání do hloubky najde vždy optimální řešení i v nekonečném stavovém prostoru.

(TRUE/FALSE) Za předpokladu, že existuje řešení a všechny ceny za akce se rovnají 2, prohledávání do šířky vždy najde optimální řešení i v nekonečném stavovém prostoru.

**Question 2** (10 points) CSP

1. (2 body) Označte následující obecná tvrzení jako pravdivá nebo nepravdivá:

(TRUE/FALSE) AC-3 algoritmus je v průběhu CSP prohledávání spuštěn pouze jednou.

(TRUE/FALSE) Představte si, že algoritmus AC-3 doběhl na problému CSP, domény jsou prořezány, nicméně každá proměnná má stále neprázdnou doménu. Nyní musí existovat řešení tohoto CSP problému.

Představte si, že potřebujete vytvořit pracovní plán pro továrnu, která vyrábí automobily jednoho typu. K dokončení auta je třeba dokončit klíčové kroky  $S$  (např. dokončit hlavní kostru, nainstalovat motor atd.). Tyto kroky musí být dokončeny v určitém pořadí. Kroky jsou indexovány  $s_1, \dots, s_S$  a každý krok  $s_i$  má sadu přípravných kroků  $\{p_1^i, \dots, p_{k_i}^i\} \subseteq S$ , které musí být dokončeny před zahájením práce na kroku  $s_i$ . Dokončení každého kroku trvá  $t_{s_i}$  jednotek času.

V továrně jsou k dispozici rekonfigurovatelné výrobní linky  $K$  (předpokládejme  $K < S$ ), které lze použít pro dokončení libovolného kroku  $s_i$ . Pokud linka  $k_j$  dokončí nějaký krok  $s_i$  (např. dokončení hlavního skeletu vozu), linka může být přímo použita pro dokončení stejného kroku pro další auto nebo linka musí být znovu nakonfigurována pro další krok  $s_j$ . Rekonfigurace linky trvá pevnou časovou dobu  $r$  a lze ji použít znovu až po této době. Pro splnění všech kroků je dán pevný horizont  $T < K$ .

Uvažte tři možné formalizace CSP:

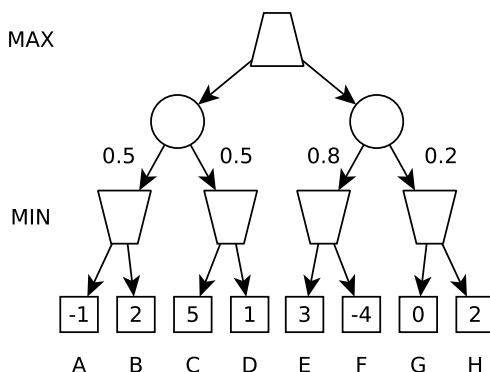
- (a) každý krok  $s_i$  odpovídá proměnné
- (b) konfigurace všech linek pro daný časový krok odpovídá jedné proměnné
- (c) každá linka odpovídá jedné proměnné

(3 body) Uvedte příklad hodnot pro jednu proměnnou pro každou z výše uvedených formalizací.

(3 body) Předpokládejte první formalizaci. Formálně запиšte podmínku, že jsou splněny předpoklady pro krok  $s_i$  (tj. předchozí kroky byly dokončeny před zahájením práce na kroku  $s_i$ ).

(2 body) Která z těchto formalizací poskytuje nejnižší vyhledávací strom (má minimální výšku)? Jaká je hodnota této výšky?

**Question 3** (10 points) *Two-player games*



Zvažte hru se dvěma hráči s nulovým součtem zobrazenou na obrázku výše. Tato hra obsahuje stochastické události modelované náhodnými uzly (kruhy ve stromu hry) s pravděpodobnostmi napsanými nad akcemi vedoucími z těchto uzlů.

- (a) (1 bod) Spočítejte očekávanou hodnotu hry, pokud hráči hrají podle optimálních strategií.
- (b) (1 bod) Představte si modifikaci alfa-beta prořezávání pro hry s náhodnými uzly. Předpokládejme, že akce jsou vyhodnocovány zleva doprava a předpokládáme, že hodnoty užitku v listech jsou přiřazeny z intervalu  $[-\infty, \infty]$ . Které listy může tento modifikovaný alfa-beta algoritmus určitě přeskočit?
- (c) (1 bod) Jak by se situace změnila, kdyby byly hodnoty užitku v listech přiřazeny z intervalu  $[-4, 5]$ ? Které listy může tento modifikovaný alfa-beta algoritmus určitě přeskočit?

(d) (3 body) Označte následující tvrzení jako pravdivé nebo nepravdivé:

- (TRUE/FALSE) Alpha-beta prořezávání může navštívit méně uzlů než Negamax.
- (TRUE/FALSE) V algoritmu Negascout se rekurzivní volání **prvního** potomka v uzlu provádí s upraveným oknem  $[\alpha, \alpha + 1]$ .
- (TRUE/FALSE) Algoritmus Monte Carlo Tree Search řeší problém explorační/exploitační v tzv. selection phase.
- (e) (2 body) Uvažujme standardní alfa-beta prořezávání s upraveným počátečním alfa-beta intervalem nastaveným na  $[-\infty, x]$ . Algoritmus vrátí hodnotu  $y > x$ . Které z následujících tvrzení je pravdivé?
  - (TRUE/FALSE) Přesná hodnota hry není známa, ale je větší nebo rovna hodnotě  $x$ .
  - (TRUE/FALSE) Přesná hodnota hry není známa, ale je větší nebo rovna hodnotě  $y$ .
- (f) (2 body) Nyní se situace změní a algoritmus vrátí hodnotu  $y < x$ . Které z následujících tvrzení je pravdivé?
  - (TRUE/FALSE) Hodnota hry je nižší nebo stejná než  $x$ .
  - (TRUE/FALSE) Hodnota hry je rovná  $y$ .