

# ÚVOD DO UMĚLÉ INTELIGENCE (CVIČENÍ 8)

**Simona Ondrčková**

# 5. DOMÁCÍ ÚKOL

Napište hráče pro hledání min.

Víte velikost tabulky, počet min sousedních políček a pravděpodobnost miny na každém políčku.

Napište funkci, která vrátí na které políčko chcete jít.

Cílem je mít lepší úspěšnost než triviální hráč.

Přesné zadání: <https://gitlab.mff.cuni.cz/finkj1am/introai>

# 6. DOMÁCÍ ÚKOL

Na marsu je robot, který se má dostat na základnu.

Robot neví kde je, a nefungují mu navigační systémy a většina senzorů.

Fungují mu motory a musí se pohybovat ve 4 směrech. Vždy ví svou relativní pozici k tomu kde začal.

Funguje mu binární čidlo, které vrací true/false na základě tmavosti/světlosti pozice.

Má mapu, ze které pro každou pozici dokáže určit stupeň šedi.

Cíle je dostat se na základnu, ale nesmí vyjet z mapy a má omezený počet kroků.

Napište program, který v každém kroku dostane hodnotu z čidla a zvolí směr kterým má robot jít.

Přesné zadání: <https://gitlab.mff.cuni.cz/finkj1am/introai>



# BOSS FIGHT

# MARKOVŮV ROZHODOVACÍ PROCES

Sekvenční rozhodovací problém v úplně pozorovatelném stochastickém prostředí s Markovským přechodovým modelem a aditivní funkcí užitku.

Robot, jež se snaží dostat do cíle.

Úplně pozorovatelné prostředí (vím kde jsem).

Mohu si zvolit akce (jakým směrem pojedu), ale nemám jistotu, že akce se povede.

Stavy mají odměnu (ocenění).

Nemůžeme si vytvořit plán, protože si nejsem jisti kam dojedeme. Vytvoříme si tedy strategii (policy)  $\pi(s)$ .

# BELLMANOVA ROVNICE UŽITKU

$$U(s) = R(s) + \gamma \max_a \sum_{s'} P(s'|s, a) * U(s')$$

$P(s'|s, a)$  je pravděpodobnost, že se ze stavu  $s$  akcí  $a$  dostanu do stavu  $s'$ .

$R(s)$  odměna při návštěvě (i opakované) stavu  $s$ .

$U(s)$  maximální očekávaný užitek ze stavu  $s$  do cíle.

Předpokládáme, že  $U(s)$  ze stavu  $s$  nezávisí na tom jak jsme se do  $s$  dostali.

# BELLMANŮV UPDATE

$$U_{i+1}(s) = R(s) + \gamma \max_a \sum_{s'} P(s'|s, a) * U_i(s')$$

Tato rovnice konverguje k  $U^*(s)$ , což je řešení Belmmanových rovnic.

Algoritmus iterace hodnot (value iteration):

$$U_1(s) = R(s)$$

$$U_{i+1}(s) = R(s) + \gamma \max_a \sum_{s'} P(s'|s, a) * U_i(s')$$

Nejvhodnější strategie (optimal policy) je pak dána jako:

$$\pi(s) = \operatorname{argmax}_a \sum_{s'} P(s'|s, a) * U^*(s')$$

# ROBOT Z PŘEDNÁŠKY

Pro akci pohyb nahoru se robot s pravděpodobností 0.8 posune nahoru a s pravděpodobností 0.1 doleva a 0.1 doprava. Pokud se nemůže posunout kam chce tak zůstane na místě.

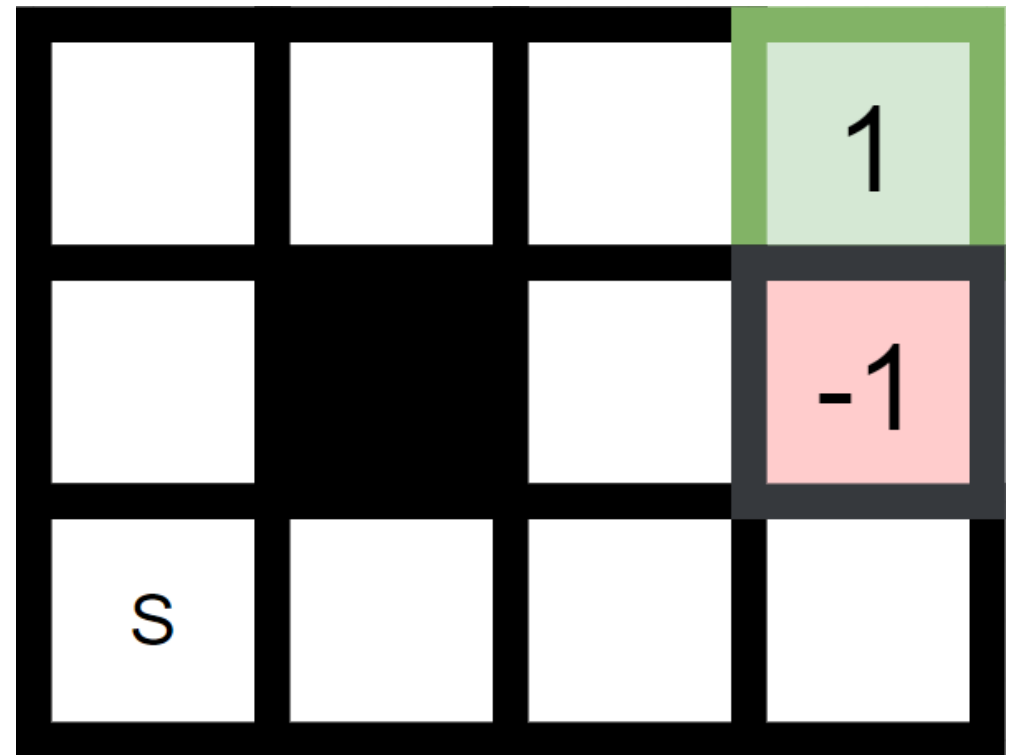
Z políček s hodnotou  $R(s)=1/-1$  nejde odejít.

Pro všechny políčka bez hodnot je  $R(s)=0$ .

Vypočítejte iterativně užitekovou funkci.

Slevový faktor  $\gamma = 0.9$

$$U_{i+1}(s) = R(s) + \gamma \max_a \sum_{s'} P(s'|s, a) * U_i(s')$$





# ROBOT Z PŘEDNÁŠKY

**Iterace 1:**  $(U_{i+1}(s) = R(s) + \gamma \max_a \sum_{s'} P(s'|s, a) * U_i(s'))$

**Stav (3,3):**  $U(s) = 0 + 0.9 \max(0.8 * 1 + 0.1 * 0 + 0.1 * 0, 0.8 * 0 + 0.1 * 0 + 0.1 * 1, 0.8 * 0 + 0.1 * 0 + 0.1 * 0) = 0.9 * 0.8 = 0.72$

Ostatní stavy jsou 0.

Vypočítejte iteraci 2.

		0.72	1
			-1
s			

# ROBOT Z PŘEDNÁŠKY

		0.72	1
			-1
s			

Iterace 2:

$$\text{Stav (3,3): } U(s) = 0 + 0.9 \max(0.8 * 1 + 0.1 * 0.72 + 0.1 * 0; 0.8 * 0 + 0.1 * 0.72 + 0.1 * 0, 0.8 * 0 + 0.1 * 1 + 0.1 * 0) = 0.9 * (0.8 + 0.072) = 0.78$$

$$\text{Stav (3,2): } U(s) = 0 + 0.9 \max(0.8 * 0.72 + 0.1 * -1 + 0.1 * 0, 0.8 * -1 + 0.1 * 0.72 + 0.1 * 0, 0.8 * 0 + 0.1 * -1 + 0.1 * 0, 0.8 * 0 + 0.1 * 0.72 + 0.1 * 0) = 0.9 * (0.8 * 0.72 - 0.1) = 0.43$$

$$\text{Stav(2,3)= } U(s) = 0 + 0.9 \max(0.8 * 0.72 + 0.1 * 0 + 0.1 * 0, 0.8 * 0 + 0.2 * 0, 0.8 * 0 + 0.1 * 0.72 + 0.1 * 0) = 0.9 * (0.8 * 0.72) = 0,52$$

Ostatní 0.

Spočítejte iteraci 3.

# ROBOT Z PŘEDNÁŠKY

	0.52	0.78	1
		0.43	-1
s			

Iterace 3:

$$\text{Stav (3,3): } U(s) = 0 + 0.9 \max(0.8 * 1 + 0.1 * 0.78 + 0.1 * 0.43, \dots) = 0.9 * (0.8 + 0.078 + 0.043) = 0.83$$

$$\text{Stav (3,2): } U(s) = 0 + 0.9 \max(0.8 * 0.78 + 0.1 * -1 + 0.1 * 0.43, \dots) = 0.9(0.624 - 0.1 + 0.043) = 0.51$$

$$\text{Stav(2,3)= } U(s) = 0 + 0.9 \max(0.8 * 0.78 + 0.2 * 0.52, \dots) = 0.9 * (0.624 + 0.104) = 0.66$$

$$\text{Stav (3,1)= } U(s) = 0 + 0.9 \max(0.8 * 0.43 + 0.1 * 0 + 0.1 * 0, \dots) = 0.9 * (0.344) = 0.31$$

$$\text{Stav(1,3)= } U(s) = 0 + 0.9 \max(0.8 * 0.52 + 0.1 * 0 + 0.1 * 0, \dots) = 0.9 * (0.416) = 0.37$$

$$\text{Kdy přestat? } |U_{k+1} - U_k| < \frac{1-\gamma}{\gamma}$$

# ROBOT Z PŘEDNÁŠKY

0.812	0.868	0.918	+1
0.762		0.660	-1
0.705	0.655	0.611	0.388

Obrázek z přednášky

Jak se rozhodne co bude robot dělat? (Policy update)

$$\pi(s) = \operatorname{argmax}_a \sum_{s'} P(s'|s, a) * U^*(s')$$

Jsem na pozici 1,1 jakou zvolím akci?

# ITERACE STRATEGIE

0.812	0.868	0.918	+1
0.762		0.660	-1
0.705	0.655	0.611	0.388

Obrázek z přednášky

V některých případech je jasné, kam jít. Jeden krok je výrazně lepší jak ostatní. Potom je vhodné tudy jít.

Algoritmus modified policy iteration, používá zjednodušenou verzi Belmmanovi rovnice:

$$U_{i+1}(s) = R(s) + \gamma \sum_{s'} P(s'|s, \pi_i(s)) * U_i(s')$$

$$\pi_{i+1}(s) = \operatorname{argmax}_a \sum_{s'} P(s'|s, a) * U_i(s')$$

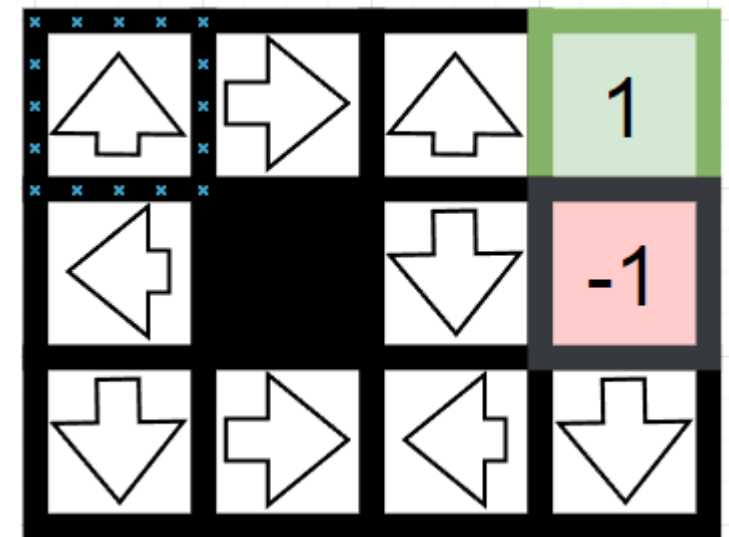
Vypočítejte U pro bod 3,2  $\pi_i$  a  $U_{i+1}$ .

# ROBOT Z PŘEDNÁŠKY

Vypočítejte užitkovou funkci a dle ní upravte strategii.

$$U_{i+1}(s) = R(s) + \gamma \sum_{s'} P(s'|s, \pi_i(s)) * U_i(s')$$

$$\pi_{i+1}(s) = \operatorname{argmax}_a \sum_{s'} P(s'|s, a) * U_i(s')$$



# ROBOT Z PŘEDNÁŠKY

Vypočítejte užitkovou funkci a dle ní upravte strategii.

$$U(3,3)=0.9*(0.8*0+0.1*1+0.1*0)=0.09$$

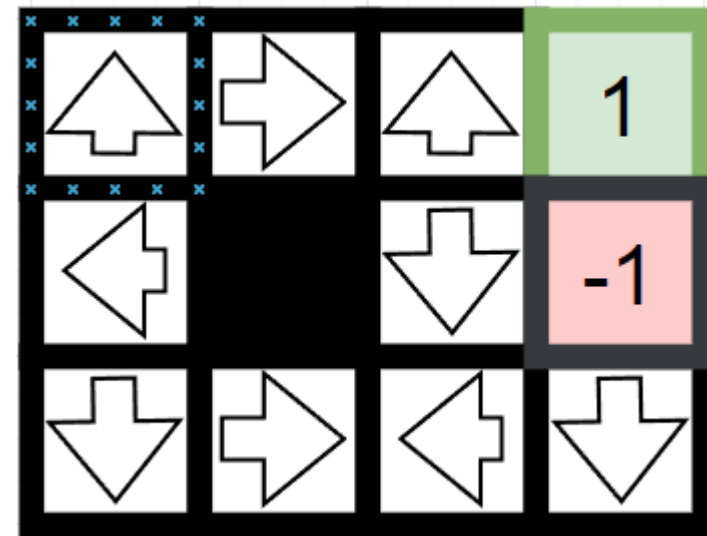
$$U(3,2)=0.9*(0.8*0+0.1*-1+0.1*0)=-0.09$$

Stav 3,2: šipka se změní na nahoru.

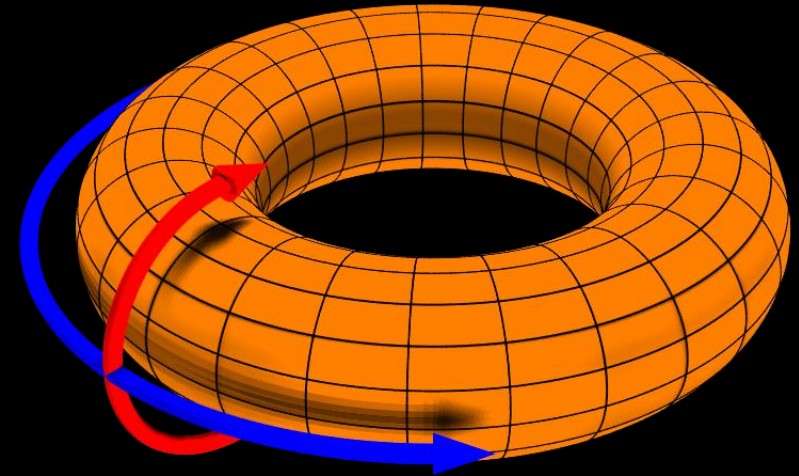
Stav 3,3: šipka se změní doprava

Stav 3,1?

Dalším krokem je znovu vypočítat U...



# 7. DOMÁCÍ ÚKOL



Obrázek z wikipedia.org

Na marsu je robot, který se má dostat na základnu.

Robot nepřistál přímo na základně musí tam dojet. Má rozbité motory a často jede jinam jako chce.

Funguje mu ale lokalizace, takže ví kde je.

Každý krok ho stojí energii. Pro každou pozici je známo kolik energie stojí jí projít.

Najděte nejkratší cestu na základnu.

Nelze předem spočítat optimální cestu

V každé pozici počítejte nejlepší pokyn, který může robot zvolit.

Přesné zadání: <https://gitlab.mff.cuni.cz/finkj1am/introai>