

# Üzleti Intelligencia

## 2. Előadás: Bevezetés a megerősítéses tanulásba

Kuknyó Dániel

Budapesti Gazdasági Egyetem

2023/24

1.félév

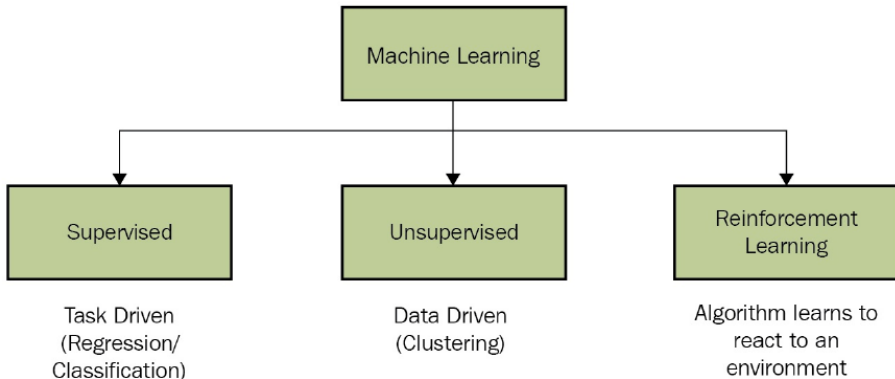
- 1 Bevezetés
- 2 Markov döntési folyamatok
- 3 Értékfüggvények
- 4 Bellman szabályok
- 5 Politika javítása

- 1 Bevezetés
- 2 Markov döntési folyamatok
- 3 Értékfüggvények
- 4 Bellman szabályok
- 5 Politika javítása

# A gépi tanulás fő területei

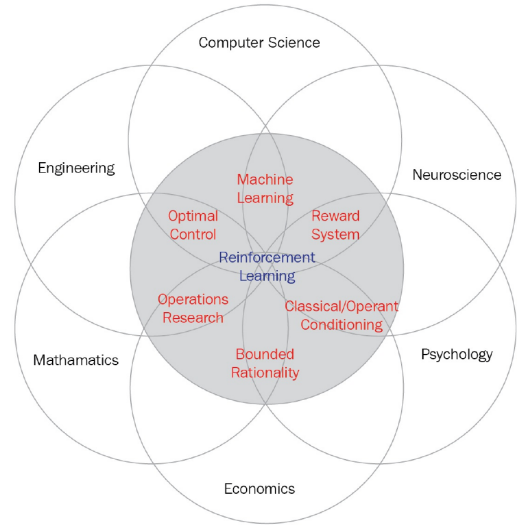
A három fő típus, ahova be lehet sorolni a gépi tanulási algoritmusokat:

- Felügyelt tanítás
- Felügyelet nélküli tanítás
- Megerősítéses tanulás



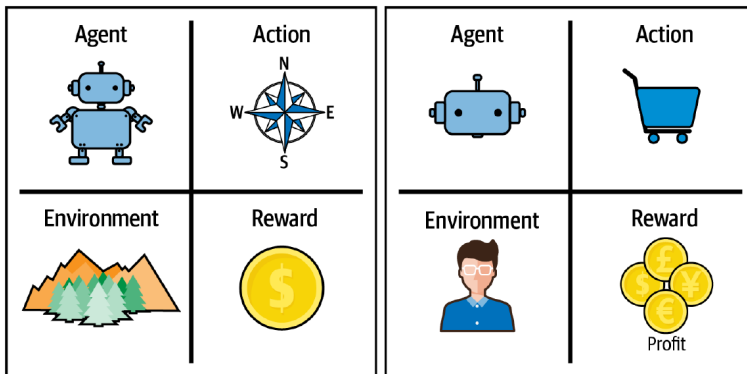
# Hol vagyunk?

A megerősítéssel tanulás számos tudományterület együttese.  
A fő elképzelése az, hogy az emberi illetve evolúcióból ismerős módszerekkel tanítson helyzetekhez alkalmazkodni tudó, intelligens modelleket valamilyen módszertan alapján.



# A megerősítéses tanulás

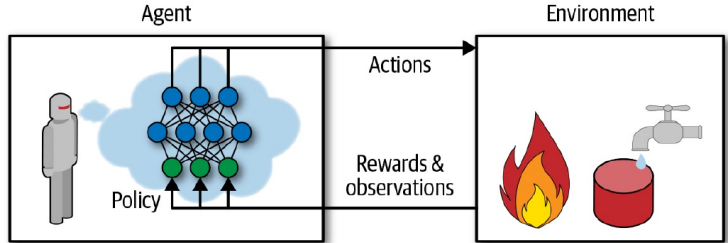
A megerősítéses tanuló modell célja, hogy a legjobb döntéseket hozza egymás után, egy adott kontextusban, hogy maximalizálja a sikert mérő értéket. A döntéshozó entitás próbákkal és hibákkal tanul. Nincs megadva, hogy milyen döntéseket hozzon, hanem ő maga tanulja meg azáltal, hogy kipróbálja azokat.



# A modellezés komponensei

## Ügynök (agent)

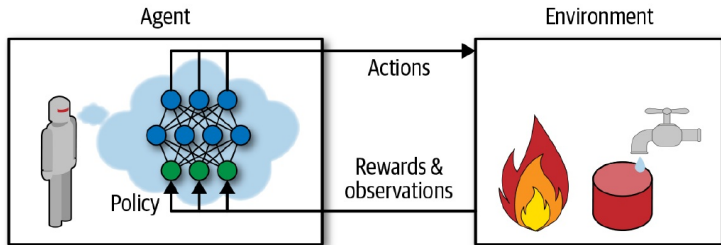
Az autonóm cselekvő, ami a feladat végrehajtására törekszik.



# A modellezés komponensei

## Környezet (environment)

Egy fekete doboz, amely az ügynök cselekvéseinek helyszíne.

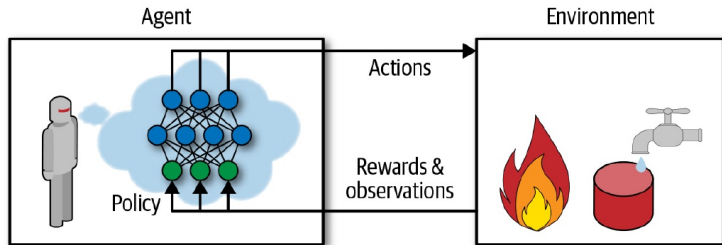




# A modellezés komponensei

## Idő (time)

RL folyamán az  
időlépések diszkrét:  
 $t \in 1, 2, 3, \dots$



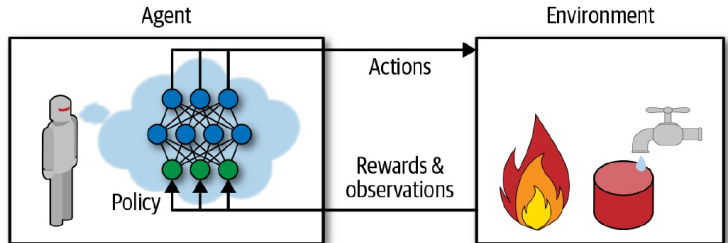
# A modellezés komponensei

## Állapot (state)

Az ügynök megfigyelése a környezetre vonatkozóan.

A környezetet leíró változók összessége.

Jelölés:  $s_t \in \mathcal{S}$

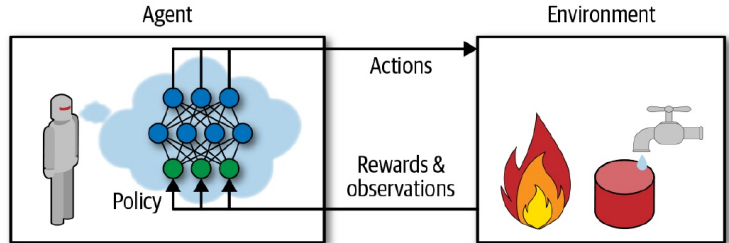


# A modellezés komponensei

## Jutalom (reward)

Az ügynök cselekvésének  
jóságát jelző skalár:

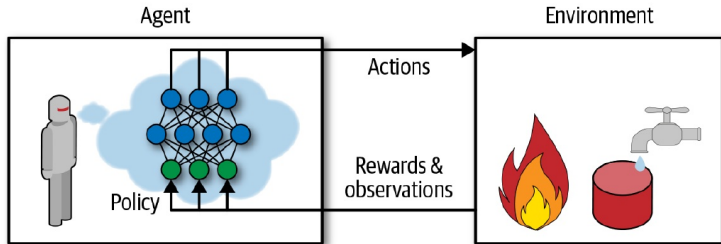
Jelölés:  $r_t \in \mathbb{R}$



# A modellezés komponensei

## Cselekvés (action)

Az ügynök által végrehajtott művelet, ami a környezetet befolyásolja.  
Jelölés:  $a_t \in A$



# A modellezés komponensei

## Politika (policy)

Egy állapot  $\rightarrow$  cselekvés leképezés. Az ügynök cselekvéseinek szabályait adja meg.

Jelölés:

- Determinisztikus:

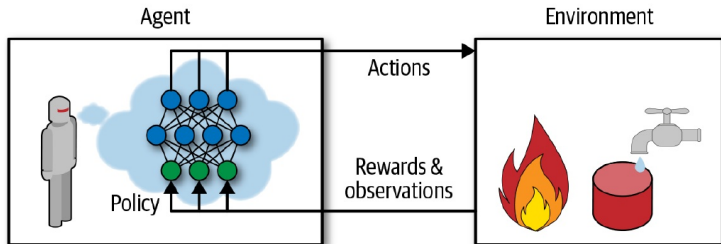
$$\pi \in S \rightarrow A$$

- Sztochasztikus:

$$\pi \in S \times A \rightarrow [0, 1]$$

Röviden:  $\pi(s, a)$

Vagy:  $\pi(a|s)$

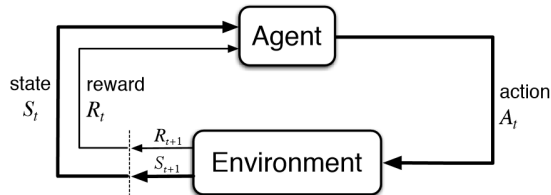


# Interakció a környezettel

- Az ügynök és a környezet egymásra hatnak. Az ügynök cselekszik, ennek hatására a környezet megváltozik. Az ügynök megfigyeli a környezetet, majd ismét cselekszik:

$$s_1 \rightarrow a_1 \rightarrow s_2 \rightarrow a_2 \rightarrow \dots \rightarrow s_t \rightarrow a_t$$

- A jutalom azonnali, és cselekvés-állapot párosért jár:  $R(s_t, a_t)$
- A környezet változását az átmeneti valószínűségek adják:  $P(s'|s_t, a_t)$
- Az ügynök célja a lehető legmagasabb jutalom összegyűjtése hosszú távon:  
 $E_\pi(r_1 + r_2 + r_3 + \dots) \rightarrow \max$

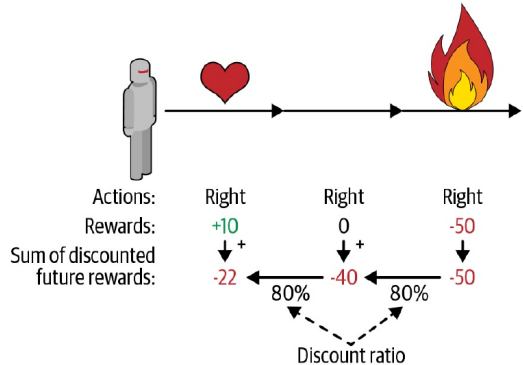


# A kredit hozzárendelési probléma

A megerősítéses tanulásban a jutalmak általában ritkák és késleltetettek.  
Például: ha az ügynök életben maradt 100 lépésen keresztül, és a 101. lépésben meghal, honnan tudjuk, melyik lépés volt érte a felelős?

A probléma megoldására a tanulás egy diszkont rátát ( $\gamma$ ) alkalmaz.

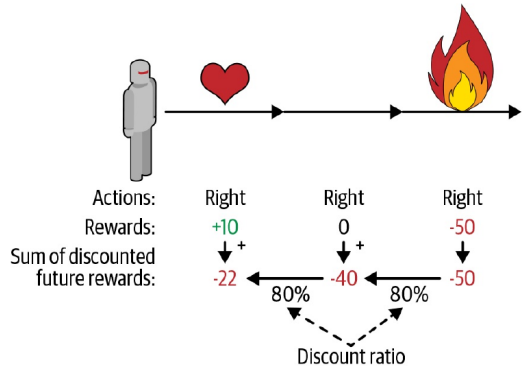
A diszkont ráta megadja a jövőbeli jutalmak jelenbeli értékét. Valamely  $r$  jutalom értéke  $k$  időlépés után  $\gamma^{k-1}$ .



# A kredit hozzárendelési probléma

Ha az ügynök háromszor egymás után jobbra megy, és +10 jutalmat kap az első lépés után, 0-t a második lépés után, és végül -50-et a harmadik lépés után, akkor feltéve, hogy  $\gamma = 0.8$  diszkontálási tényezőt használ, az első lépés hozama  $10 + \gamma 0 + \gamma^2(-50) = -22$  lesz.

Ha a diszkontálási tényező közel van a 0-hoz, akkor a jövőbeli jutalmak nem számítanak sokat az azonnali jutalmakhoz képest. Ha viszont a diszkontálási tényező közel van 1-hez, akkor a jutalmak a jövőben majdnem ugyanannyit számítanak, mint az azonnali jutalmak.





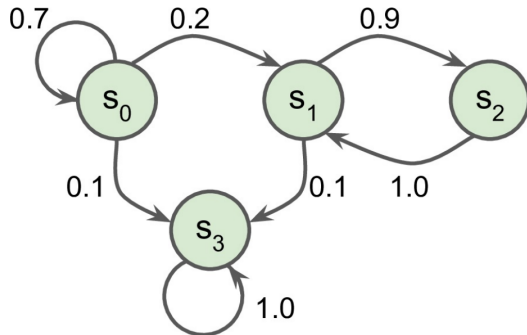
- 1 Bevezetés
- 2 Markov döntési folyamatok
- 3 Értékfüggvények
- 4 Bellman szabályok
- 5 Politika javítása

# Markov láncok

## Markov lánc

Memória nélküli sztochasztikus folyamat fix számosságú állapottal, amely véletlenszerűen vált állapotot minden lépésben.

Az átmeneti valószínűség az aktuális állapotból ( $s$ ) a következő állapotba ( $s'$ ) előre meghatározott, és csak az  $(s, s')$  pároson múlik, múltbeli állapotokon nem.

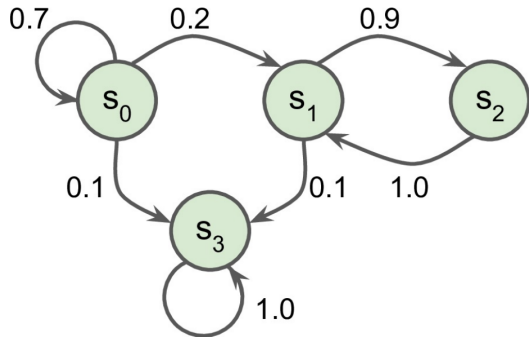


# Markov láncok

## Markov tulajdonság

Az ügynök nem nyerhet semmit azáltal, hogy ismeri az előző állapotokat.

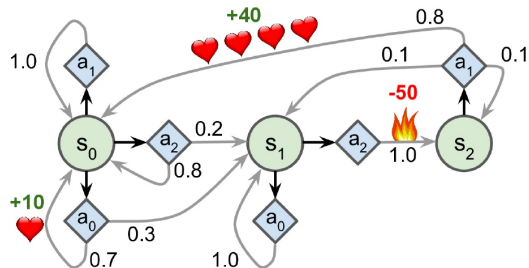
- Miért fontos ez?
- Milyen példákat lehet mondani ilyen játékokra?



# Markov döntési folyamatok

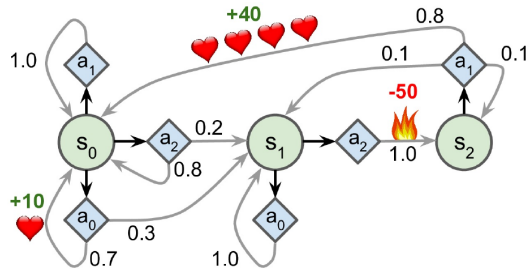
## Markov döntési folyamat (MDP)

Az ügynök minden lépésben választhat egy cselekvés közül. A környezet állapot átmeneti valószínűsége a következő állapotba a választott cselekvésen fog múlni. Ezenkívül némelyik állapot átmenetek jutalommal (pozitív vagy negatív) járnak.



# Markov döntési folyamatok

Ha az ügynök az  $s_0$  állapotban kezd, választhat az  $a_0$ ,  $a_1$  vagy  $a_2$  cselekvések között. Ha az  $a_1$  cselekvést választja, biztosan az  $s_0$  állapotban marad jutalom nélkül. De ha az  $a_0$  cselekvést választja, akkor 70% esélye van arra, hogy +10 jutalmat kapjon és az  $s_0$  állapotban maradjon. Előbb-utóbb az  $s_1$  állapotba fog megérkezni. Itt csak két lehetséges cselekvés van:  $a_0$  és  $a_2$ . Az  $a_0$  cselekvéssel ugyanabban az állapotban marad, vagy továbblép az  $s_2$  állapotba és -50 jutalmat kap. Az  $s_2$  állapotban csak az  $a_1$  cselekvést teheti meg, amely visszavezeti az  $s_0$  állapotba, +40 jutalommal.



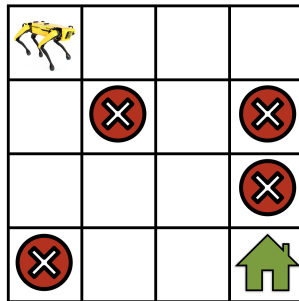
# A Markov döntési folyamat megoldása

A probléma felírása:

- Az ügynök  $s_0$  kezdőállapotból indul.
- Minden cselekvését  $\pi$  politika határozza meg.
- A környezet az állapot és a cselekvés alapján ad jutalmat:  
 $s_{t+1} \sim P(s_t, a_t); r_{t+1} \sim R(s_t, a_t)$
- A politika optimális, ha a kumulált diszkontált jutalma (**hozama**) maximális:

$$\begin{aligned} G_t &= (r_1 + \gamma r_2 + \gamma^2 r_3 + \dots) = \\ &= \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \rightarrow \max \end{aligned}$$

- A cél az **optimális politika** megtalálása.



- 1 Bevezetés
- 2 Markov döntési folyamatok
- 3 Értékfüggvények
- 4 Bellman szabályok
- 5 Politika javítása

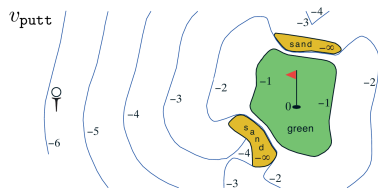
# Állapot-érték függvény

Az értékfüggvények az állapotok függvényei amik megadják, hogy mennyire jó az ügynöknek, hogy egy adott állapotban áll.

## Állapot-érték függvény (value)

Egy  $s$  állapot állapot-értéke ( $v_\pi(s)$ ) valamely  $\pi$  politika szerint a várható hozam, ha az ügynök  $s$  állapotból indul, és utána  $\pi$  szerint hozza döntéseit:

$$\begin{aligned} v_\pi(s) &= E_\pi [G_t | S_t = s] = \\ &= E_\pi \left[ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid S_t = s \right] \end{aligned}$$

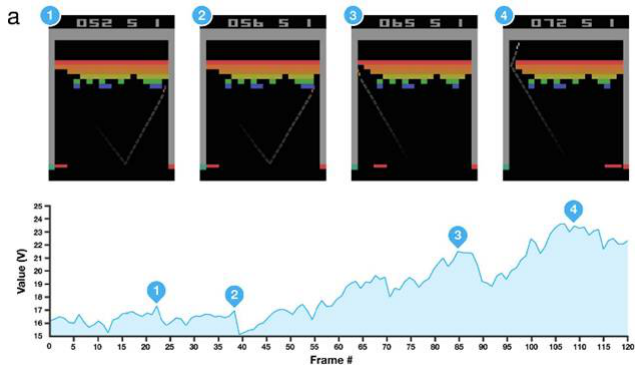


Példa: golfozásban a *putter* a kis hatótávú golfütőre vonatkozik. A diagramon azt látjuk, mennyire jó egy adott pozícióból ütni annak az ügynöknek, aki csak a putter ütőt használja. A terminális állapotban az érték 0, és minél távolabb van tőle, annál inkább csökken az értéke. A homokon a politika  $-\infty$  értéket kap.



# Állapot-értékek a Breakout-ban

A Breakout egy retró Atari játék, amiben a cél az, hogy az ütővel a játékos leüsse az összes téglát. A diagramon az adott állapothoz tartozó állapot-érték látható. Amikor felkerül a labda az állapot-érték is magasabb, mert ott potenciálisan több téglát tud kiütni.



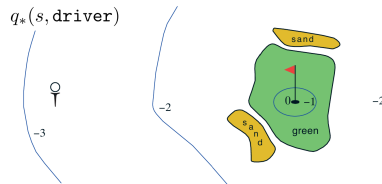
# Állapot-cselekvés minőség függvény

Hasonlóan az előzőhöz lehetséges definiálni egy adott állapot-cselekvés páros minőségének függvényét, amely megadja mennyire jó az ügynöknek, hogy egy adott állapotban áll, majd adott cselekvést hajt végre.

## Állapot-cselekvés minőség függvény (quality)

Egy  $(s, a)$  állapot-cselekvés páros minőség függvénye valamely  $\pi$  politika szerint a várható hozam, ha az ügynök  $s$  állapotból indul,  $a$  cselekvést hajtja végre, majd utána  $\pi$  szerint hozza döntéseit:

$$\begin{aligned} q_{\pi}(s, a) &= E_{\pi} [G_t | S_t = s, A_t = a] = \\ &= E_{\pi} \left[ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid S_t = s, A_t = a \right] \end{aligned}$$



Példa: golfozásban a *driver* a nagy hatótávú ütőre vonatkozik. Ebben az esetben a  $q(s, \text{driver})$  minőség függvény azt adja meg mennyire jövedelmező a játékosnak egy adott helyen állni, és onnan a driver ütőt választani a következő lövéshez.

- 1 Bevezetés
- 2 Markov döntési folyamatok
- 3 Értékfüggvények
- 4 Bellman szabályok**
- 5 Politika javítása

# Állapot-érték Bellman szabály

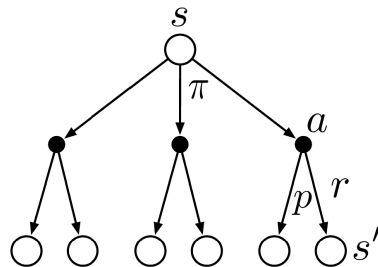
Az értékfüggvények egy alapvető tulajdonsága, hogy betartanak egy rekurzív kapcsolati rendszert.

Minden  $\pi$  politikára és bármely  $s$  állapot esetén érvényes a következő konzisztencia kritérium  $s$  állapot és  $s'$  következő állapotai között:

## Állapot-érték Bellman szabály

$$v_{\pi}(s) = \sum_a \pi(a|s) \sum_{s', r} p(s', r|s, a) [r + \gamma v_{\pi}(s')]$$

*minden  $s \in S$  – re*



- $\pi(a|s)$ :  $a$  cselekvés valószínűsége  $s$  állapotból  $\pi$  politika szerint.
- $p(s', r|s, a)$ :  $s'$  következő állapot és  $r$  jutalom valószínűsége, ha adott  $s$  állapot és  $a$  cselekvés.

# Állapot-cselekvés minőség Bellman szabály

Hogyan lehet javítani egy  $\pi$  politikát?

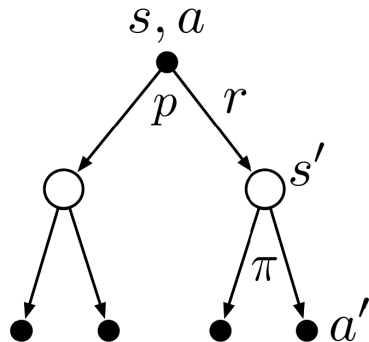
Azt tudjuk, hogy mennyire jövedelmező egy  $s$  állapotból  $\pi$ -t követni - ez a  $v_\pi(s)$ .

Érdemes lenne eltérni  $\pi$  politikától egy adott  $a$  cselekvést választva?

Ezt adja meg az állapot-cselekvés minőség függvény: mennyire jövedelmező egy ügynöknek  $s$  állapotból  $a$  cselekvést választani, majd utána  $\pi$  politikát követni:

## Állapot-cselekvés minőség Bellman szabály

$$q_\pi(s, a) = \sum_{s', r} p(s', r | s, a) [r + \gamma v_\pi(s')]$$



- $p(s', r | s, a)$ :  $s'$  következő állapot és  $r$  jutalom valószínűsége, ha adott  $s$  állapot és  $a$  cselekvés.

- 1 Bevezetés
- 2 Markov döntési folyamatok
- 3 Értékfüggvények
- 4 Bellman szabályok
- 5 Politika javítása

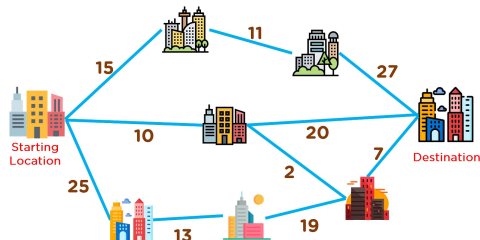
# Mohó ügynök

Hogyan válasszon cselekvést az ügynök?  
A legegyszerűbb cselekvés kiválasztási szabály, ha az ügynök mindig a számára elérhető legnagyobb értékű cselekvést választja. Ha több ilyen is van, tetszőlegesen választhat közöttük.

## Mohó cselekvés választás

$$A_t = \underset{a}{\operatorname{argmax}} Q_t(a)$$

- Mindig a mohó a legjobb megoldás?
- A legjobb megoldás mohó?



Melyik úton jutna el a mohó ügynök a kezdő városból a cél városba, ha a lehető legkevesebbet akarja költeni üzemanyagra?

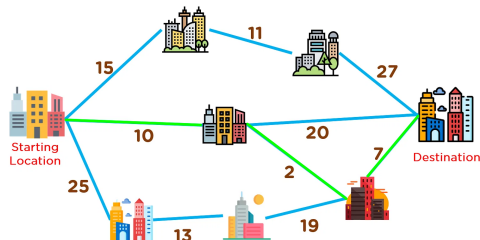
# Mohó ügynök

Hogyan válasszon cselekvést az ügynök?  
A legegyszerűbb cselekvés kiválasztási szabály, ha az ügynök mindig a számára elérhető legnagyobb értékű cselekvést választja. Ha több ilyen is van, tetszőlegesen választhat közöttük.

## Mohó cselekvés választás

$$A_t = \underset{a}{\operatorname{argmax}} Q_t(a)$$

- Mindig a mohó a legjobb megoldás?
- A legjobb megoldás mohó?



Melyik úton jutna el a mohó ügynök a kezdő városból a cél városba, ha a lehető legkevesebbet akarja költeni üzemanyagra?



# GridWorld

A következő példában egy egyszerű GridWorld játéknak láthatóak az állapot-értékei (jobb) és a mohó ügynök adott állapot-értékhez tartozó cselekvései. A játék célja, hogy az ügynök elérje valamelyik szürke zónát.

Véletlen politika értéke ( $v_k$ )

0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0

Mohó stratégia  $v_k$  szerint

	↕↕↕	↕↕↕	↕↕↕
↕↕↕	↕↕↕	↕↕↕	↕↕↕
↕↕↕	↕↕↕	↕↕↕	↕↕↕
↕↕↕	↕↕↕	↕↕↕	

$$k = 0$$

# GridWorld

A következő példában egy egyszerű GridWorld játéknak láthatóak az állapot-értékei (jobb) és a mohó ügynök adott állapot-értékhez tartozó cselekvései. A játék célja, hogy az ügynök elérje valamelyik szürke zónát.

Véletlen politika értéke ( $v_k$ )

0.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	0.0

Mohó stratégia  $v_k$  szerint

	←	↕	↕
↑	↕	↕	↕
↕	↕	↕	↓
↕	↕	→	

$$k = 1$$



# GridWorld

A következő példában egy egyszerű GridWorld játéknak láthatóak az állapot-értékei (jobb) és a mohó ügynök adott állapot-értékhez tartozó cselekvései. A játék célja, hogy az ügynök elérje valamelyik szürke zónát.

Véletlen politika értéke ( $v_k$ )

0.0	-1.7	-2.0	-2.0
-1.7	-2.0	-2.0	-2.0
-2.0	-2.0	-2.0	-1.7
-2.0	-2.0	-1.7	0.0

Mohó stratégia  $v_k$  szerint

	←	←	↕
↑	↖	↕	↓
↑	↕	↘	↓
↕	→	→	

$$k = 2$$

# GridWorld

A következő példában egy egyszerű GridWorld játéknak láthatóak az állapot-értékei (jobb) és a mohó ügynök adott állapot-értékhez tartozó cselekvései. A játék célja, hogy az ügynök elérje valamelyik szürke zónát.

Véletlen politika értéke ( $v_k$ )

0.0	-2.4	-2.9	-3.0
-2.4	-2.9	-3.0	-2.9
-2.9	-3.0	-2.9	-2.4
-3.0	-2.9	-2.4	0.0

Mohó stratégia  $v_k$  szerint

	←	←	↙
↑	↖	↙	↓
↑	↖	↘	↓
↖	→	→	

$$k = 3$$

# GridWorld

A következő példában egy egyszerű GridWorld játéknak láthatóak az állapot-értékei (jobb) és a mohó ügynök adott állapot-értékhez tartozó cselekvései. A játék célja, hogy az ügynök elérje valamelyik szürke zónát.

Véletlen politika értéke ( $v_k$ )

0.0	-6.1	-8.4	-9.0
-6.1	-7.7	-8.4	-8.4
-8.4	-8.4	-7.7	-6.1
-9.0	-8.4	-6.1	0.0

Mohó stratégia  $v_k$  szerint

	←	←	↙
↑	↖	↙	↓
↑	↗	↘	↓
↘	→	→	

$$k = 10$$

# GridWorld

A következő példában egy egyszerű GridWorld játéknak láthatóak az állapot-értékei (jobb) és a mohó ügynök adott állapot-értékhez tartozó cselekvései. A játék célja, hogy az ügynök elérje valamelyik szürke zónát.

Véletlen politika értéke ( $v_k$ )

0.0	-14.	-20.	-22.
-14.	-18.	-20.	-20.
-20.	-20.	-18.	-14.
-22.	-20.	-14.	0.0

Mohó stratégia  $v_k$  szerint

	←	←	↙
↑	↖	↙	↓
↑	↖	↘	↓
↖	→	→	

$$k = \infty$$