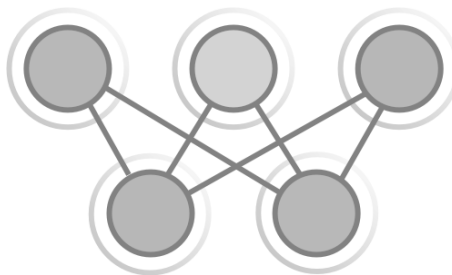



Zvonimir Pavlić  
Tomislav Lugarić  
Goran Narančić

Fakultet elektrotehnike i računarstva  
Zavod za elektroniku, mikroelektroniku, računalne i inteligentne sustave

# Strojno učenje

## Podržano učenje



- 
- podržano učenje – uvodna razmatranja
  - Q - učenje
  - proširenja osnovnih koncepata
  - primjeri



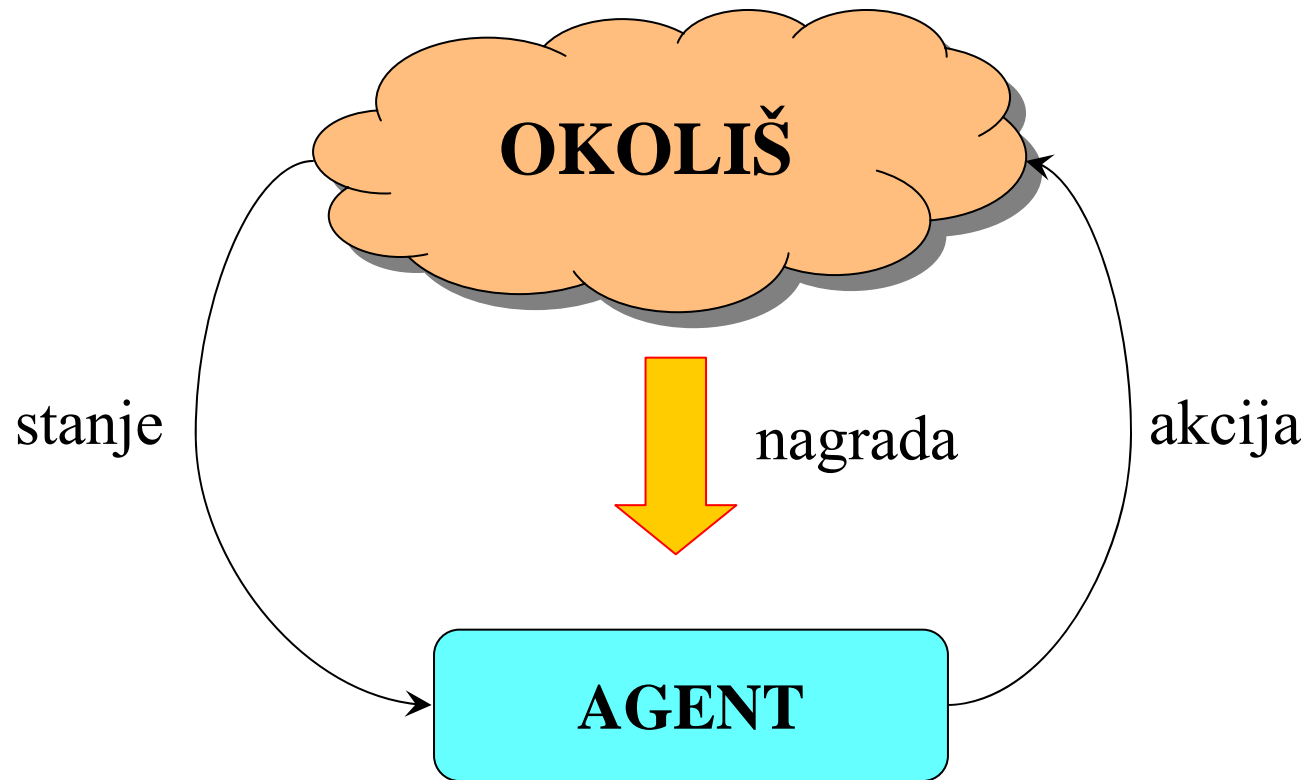
# Podržano učenje

## uvodna razmatranja

Zvonimir Pavlić

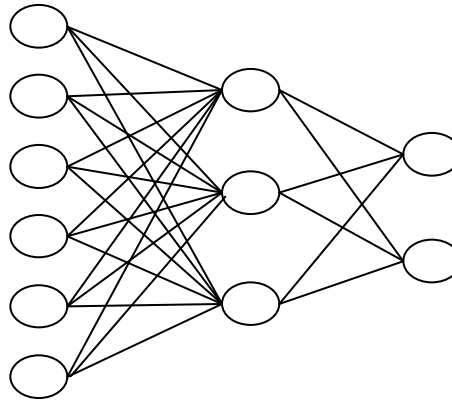


- nenadzirano učenje – ograničeno područje primjene
- nadzirano učenje – potreba učitelja
- **podržano učenje** (engl. *reinforcement learning*) – procjenitelj (engl. *critic*) – ne govori unaprijed što raditi, nego daje odgovarajuće “nagrade” ili “kazne” na kraju niza akcija



- Problem priznanja zasluge (engl *credit assignment*)

- problem priznanja zasluge (engl. *credit assignment*)
  - kako odrediti koje su akcije odgovorne za konačan ishod
  - istovjetan problem kao u neuronskim mrežama



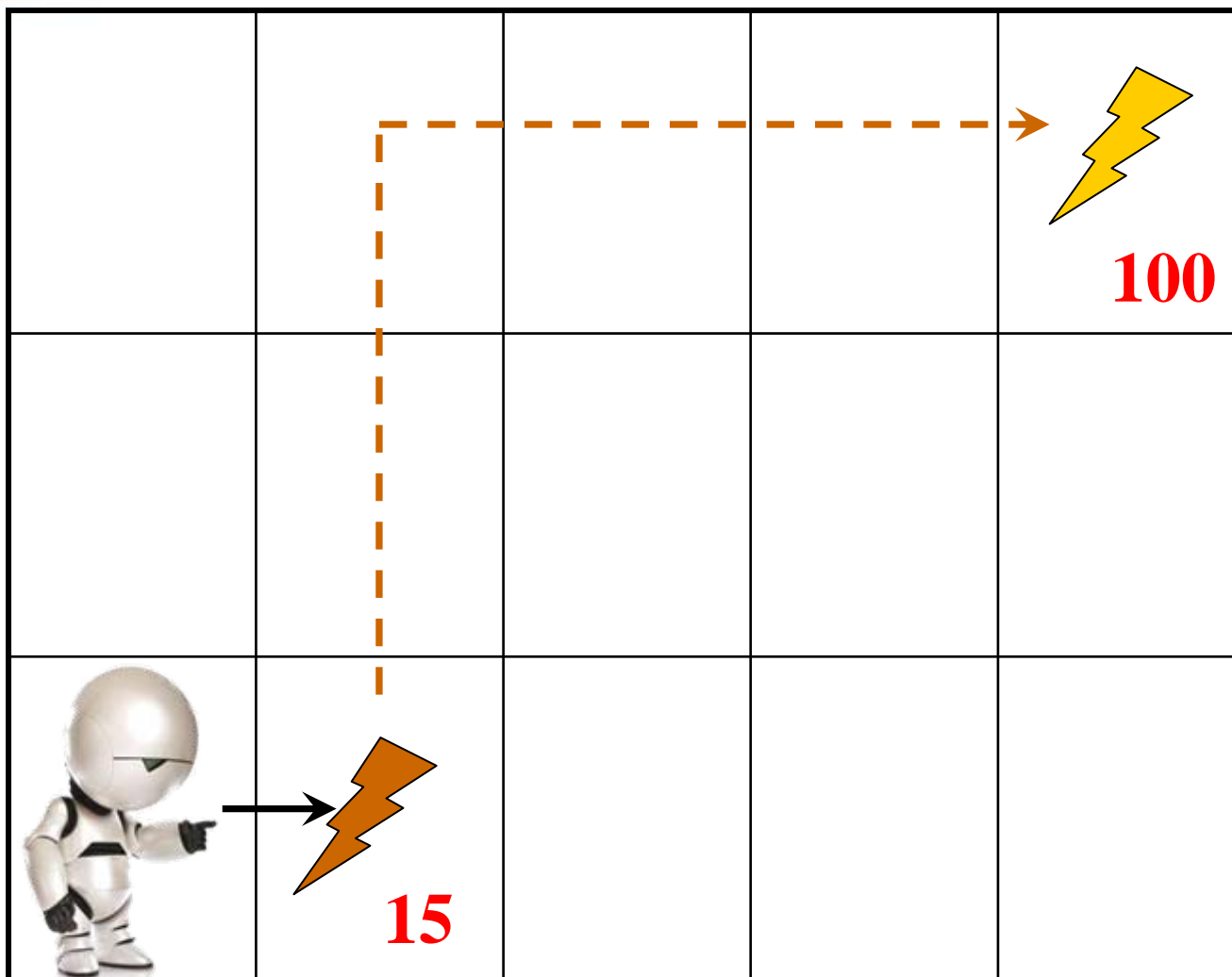
- Minsky i Papert – zaustavili istraživanje neuronskih mreža na dvadesetak godina

- odgođene nagrade
  - priznanje zasluga
- istraživanje
  - agent utječe na izbor primjera za učenje:
    - istraživanje nepoznatih stanja u potrazi za informacijama
    - iskorištavanje poznatih stanja
- cjeloživotno učenje (engl. *life-long learning*)
  - učenje različitih zadataka u istom okolišu
- djelomično vidljiva stanja
  - očitavanja senzora nedovoljna za cjelovitu sliku


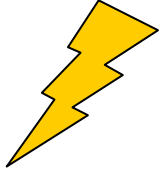







# Specifičnosti podržanog učenja - istraživanje



# Specifičnosti podržanog učenja – cjeloživotno učenje

				 <b>100</b>
				
				

# Formalna definicija problema učenja – pojmovi (1)

- skup stanja (engl. *state*) –  $s \in S$
- skup akcija (engl. *action*) –  $a \in A$
- strategija (engl. *policy*) –  $\pi : S \rightarrow A$
- dobrota stanja  $s$  (engl. *value*) –  $V(s)$
- dobrota akcije  $a$  u stanju  $s$  (engl. *quality*) -  $Q(s,a)$
- nagrada (engl. *reward*) -  $r(s)$  ili  $r(s,a)$
- koeficijent umanjenja nagrade –  $\gamma$
- dobrota strategije odlučivanja, počevši iz stanja  $s$

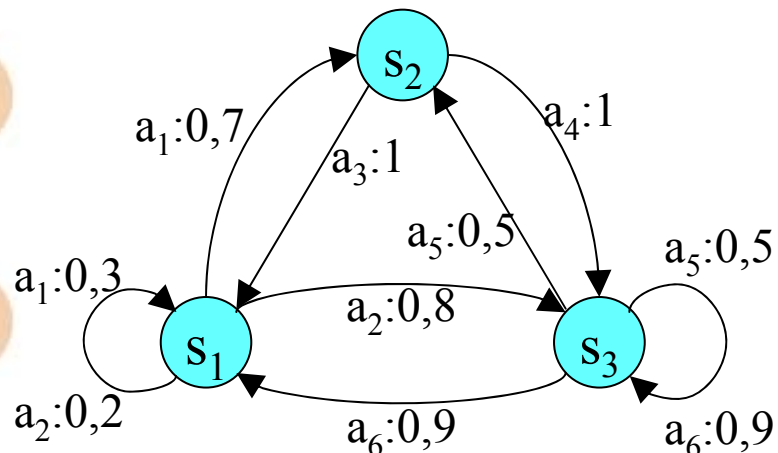
$$V^{\pi}(s) = \sum_{i=0}^{\infty} \gamma^i r_{t+i}$$

## Formalna definicija problema učenja – pojmovi (2)

- optimalna strategija (\* - znak za optimalno):

$$\pi^* = \arg \max_{\pi} V^{\pi}(s), (\forall s)$$

- često se koristi dobrota najbolje strategije:  $V^{\pi^*}(s)$
- Markovljev proces odlučivanja (MDP)



prijelazi:  $s' = \delta(s, a)$

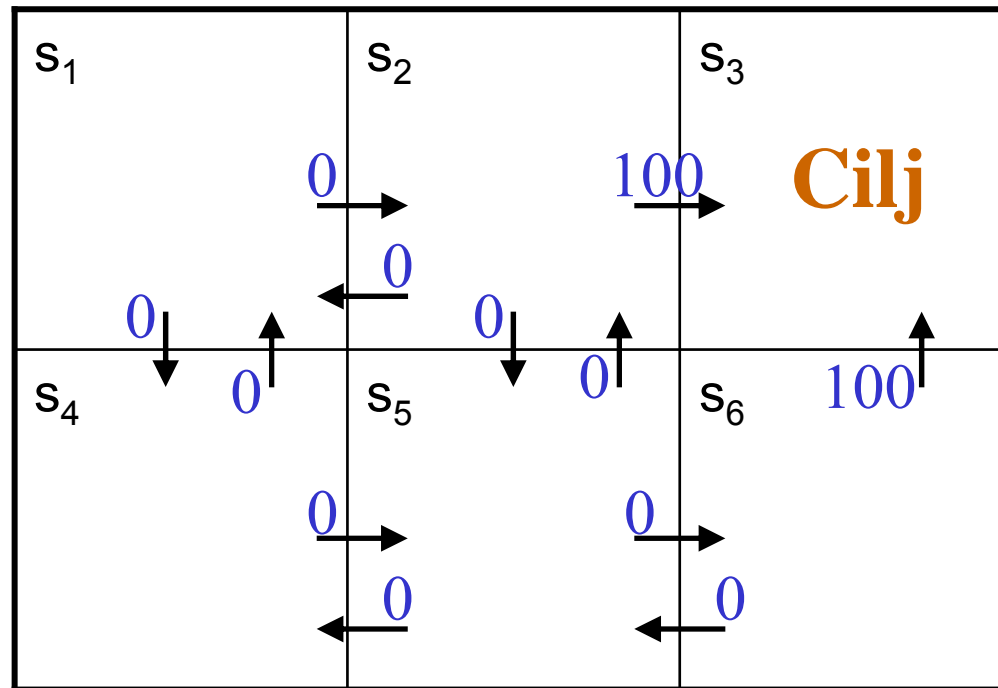
nagrade:  $r(s, a)$

- funkcije  $\delta$  i  $r$  ovise samo o trenutnom stanju  $s$  i akciji  $a$

- učenje s **unaprijed poznatim modelom** svijeta
  - odgođena nagrada
  - rješenje – dinamičko programiranje
  - izuzetno rijetki slučajevi
- učenje s **iterativnim upoznavanjem svijeta** (engl. *temporal difference*)
  - svijet nije u potpunosti unaprijed poznat
  - kompromis između istraživanja i ponovnog iskorištenja već poznatih stanja
  - pokriva većinu problema

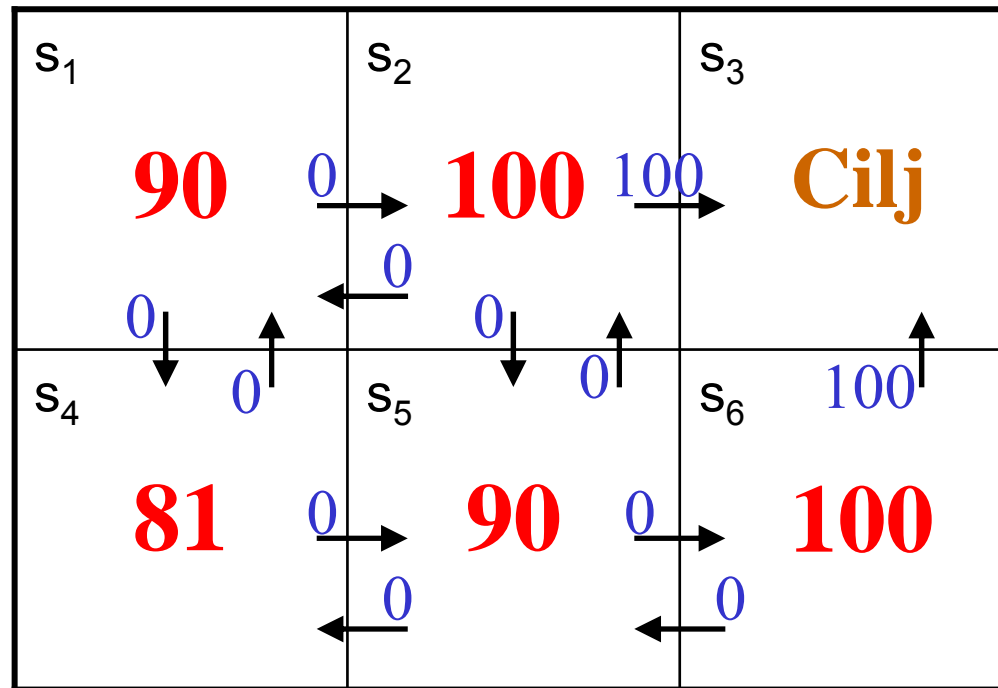
- iteracija po vrijednostima (engl. *value iteration*)
  - uči se određivanjem dobrote stanja
- iteracija po strategijama (engl. *policy iteration*)
  - uči se izravnim određivanjem strategija biranja akcija
  - Veća složenost jednog koraka nego kod iteracije vrijednosti, ali manji broj koraka
- najprikladnijim izborom pokazalo se određivanje dobrote para (*stanje, akcija*), pomoću funkcije  $Q(s,a)$

$$V^*(s), \gamma = 0,9$$



$$V(s) = \max_a r(s, a) + \gamma V^*(\delta(s, a))$$

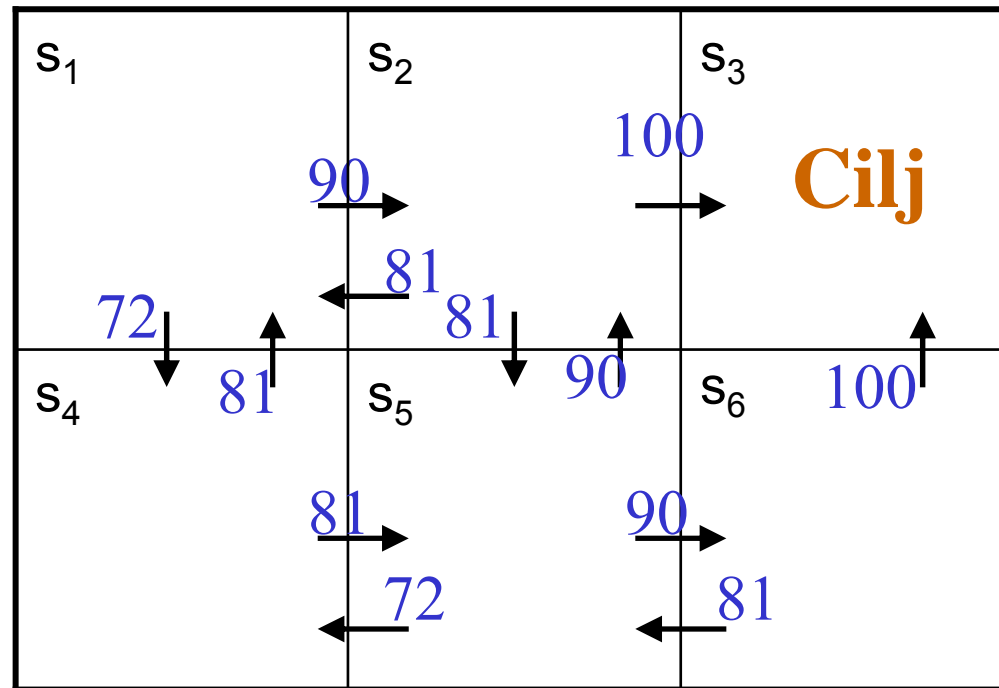
$$V^*(s), \gamma = 0,9$$



$$V(s) = \max_a r(s, a) + \gamma V^*(\delta(s, a))$$

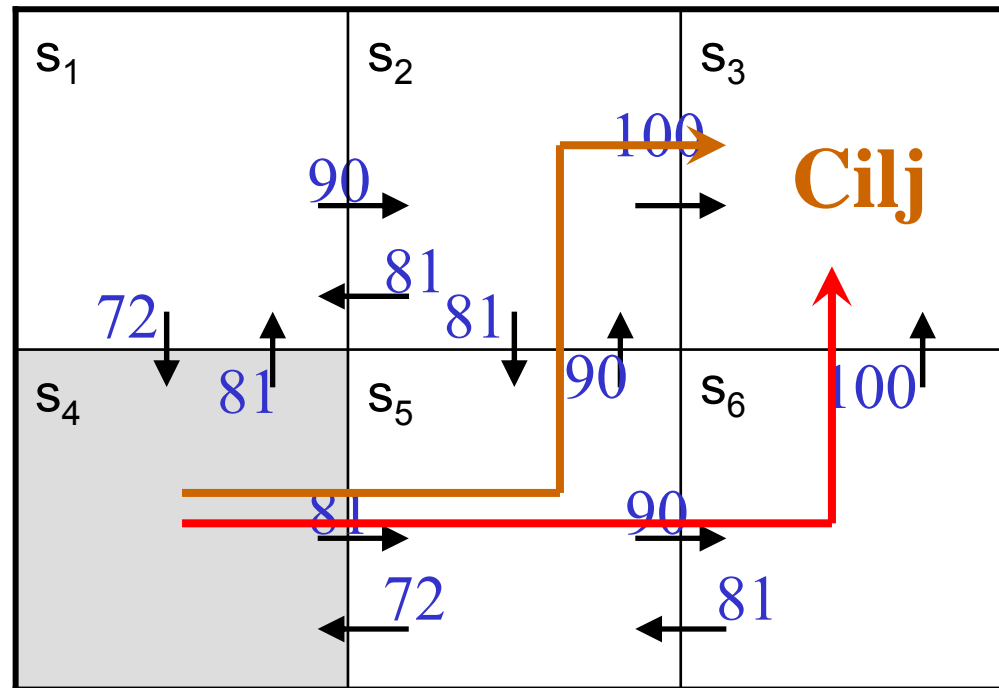


$$Q(s,a), \gamma=0,9$$



$$Q(s,a) = r(s,a) + \gamma \max_a Q(\delta(s,a), a')$$

# Primjer optimalne strategije



- početno stanje:  $s_4$
- ima više optimalnih strategija (nisu prikazane sve)

# Q učenje

Tomislav Lugarić

- metoda podržanog učenja
- učenje vrijednosti stanja i akcija – Q funkcija
- odrediti vrijednost akcije bez poznavanja kompletnog modela svijeta

# Automat sa K poluga (K-armed bandit)

- hipotetski automat nalik na one u kockarnici
- svaka poluga donosi određeni dobitak
- strategija izbora poluge
- pojednostavljeni problem Q - učenja

- potreba za učenjem optimalne strategije
- agent želi odabrati akciju koja maksimizira nagradu koju dobiva

$$\pi^*(s) = \arg \max_a \left[ r(s, a) + \gamma V^*(\delta(s, a)) \right]$$

- potrebno poznavati funkciju  $r(s, a)$  i sve prijelaze
- uvodi se Q vrijednost

$$Q(s, a) = r(s, a) + \gamma V^*(\delta(s, a))$$

- potrebno naučiti Q

- Za učenje potrebna rekurzivna definicija

$$Q(s, a) = r(s, a) + \gamma V^*(\delta(s, a))$$

$$V^*(s) = \max_a Q(s, a')$$

$$Q(s, a) = r(s, a) + \gamma \max_a Q(\delta(s, a), a')$$

- funkcija Q nepoznata – učenje aproksimacijom
  - $\hat{\cdot}$  - oznaka aproksimacije

$$\hat{Q}(s, a) \leftarrow r(s, a) + \gamma \max_{a'} \hat{Q}(s', a')$$

- dodatna mogućnost – stopa učenja

$$\hat{Q}(s, a) \leftarrow \eta(r(s, a) + \gamma \max_{a'} \hat{Q}(s', a')) + (1 - \eta)\hat{Q}(s, a)$$

$$0 < \eta \leq 1$$



- za svako stanje postavi vrijednost  $Q(s,a)$  na nulu
- postavi se u neko stanje  $s$
- ponavljaj beskonačno:
  - odaberi i izvrši akciju  $a$
  - primi nagradu  $r$
  - osvježi zapis  $Q(s,a)$  prema formuli:

$$Q(s, a) \leftarrow r(s, a) + \gamma \max_{a'} Q(s', a')$$

- uzmi stanje  $s'$  kao novo stanje  $s$

- za svako stanje postavi vrijednost  $Q(s,a)$  na nulu
- postavi se u neko stanje  $s$
- ponavljaj beskonačno:
  - odaberi i izvrši akciju  $a$
  - primi nagradu  $r$
  - osvježi zapis  $Q(s,a)$  prema formuli:

$$Q(s, a) \leftarrow \eta(r(s, a) + \gamma \max_{a'} Q(s', a')) + (1 - \eta)Q(s, a)$$

- uzmi stanje  $s'$  kao novo stanje  $s$ 
  - $\eta = \eta * k, 0 < k < 1$

- algoritam konvergira samo pod određenim uvjetima
  - 1) sustav je deterministički Markovljev proces odlučivanja
  - 2) nagrade su ograničene
  - 3) svaki par stanje-akcija se izvede beskonačno često
- Dokaz – Mitchell str. 381, teorem 13.1

# Način osvježavanja Q vrijednosti

- osvježavanje samo jednog koraka
  - sporije, osvježava se Q samo za jedan korak
- učenje unatrag
  - agent pamti kuda je prošao i osvježava sve Q vrijednosti po putu
- epizoda

- rizik od prevelike preferencije već nađenih puteva
- probabilistički pristup izboru akcije
- npr:

$$P(a_i | s) = \frac{k^{Q(s, a_i)}}{\sum_j k^{Q(s, a_j)}}$$

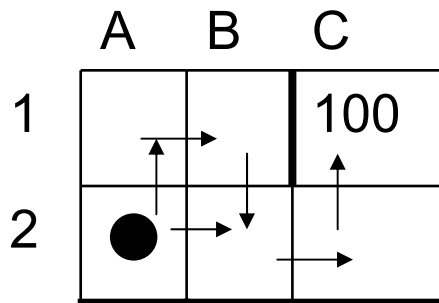
- ili:  $\epsilon$  određuje vjerojatnost da se akcija odabire nasumce
- $\epsilon$  – može se mijenjati (npr. smanjivati) tokom izvođenja

$$0 < \epsilon < 1$$

• Vidjeti priloženo datoteku s animacijama

## Primjer Q učenja

- Skup akcija:  $\leftarrow, \uparrow, \rightarrow, \downarrow$
- $\gamma = 0.9$



$$Q(s, a) \leftarrow r(s, a) + \gamma \max_{a'} Q(s', a')$$

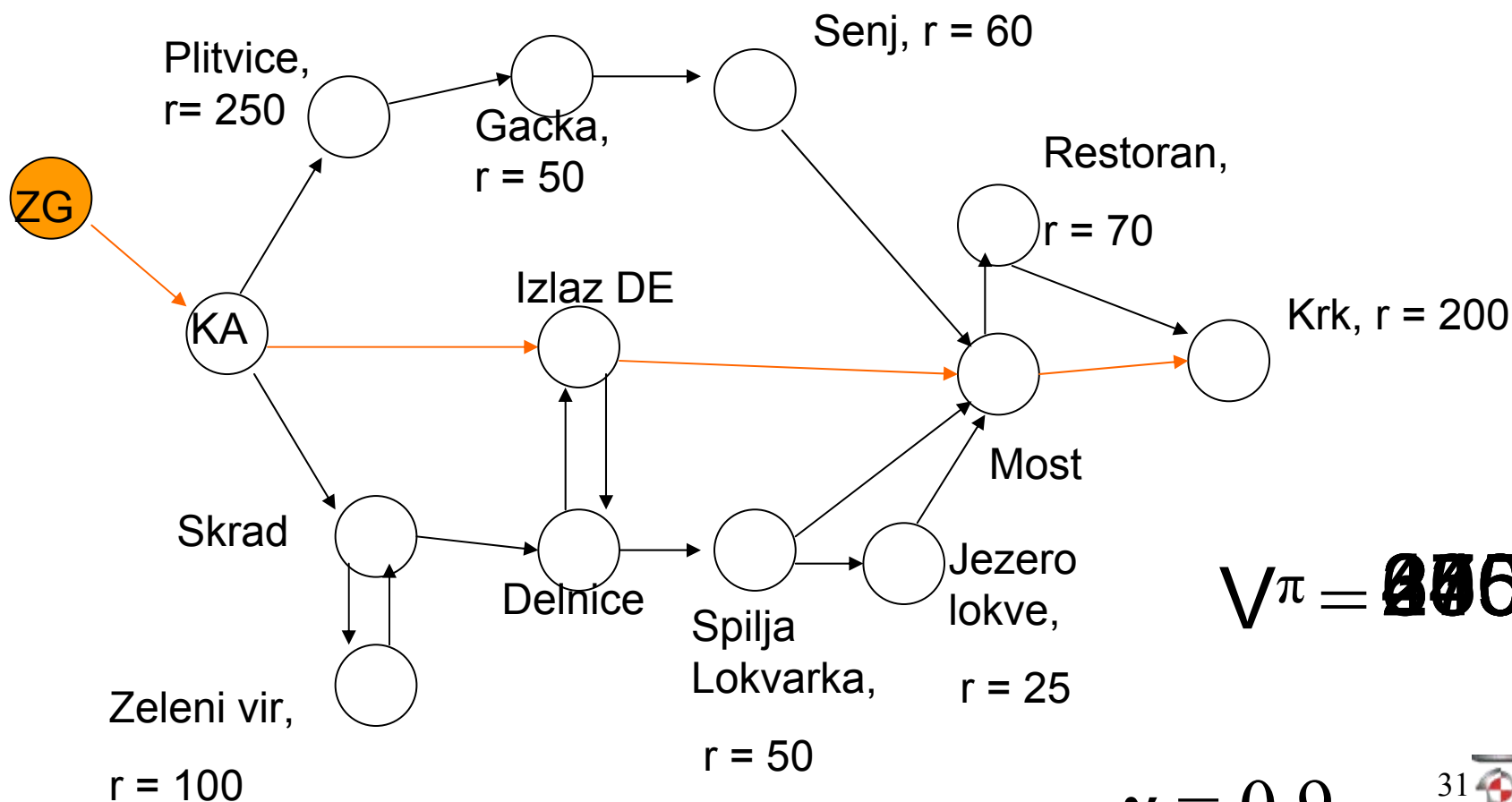
$$Q(s, a) \leftarrow \eta(r(s, a) + \gamma \max_{a'} Q(s', a')) + (1 - \eta)Q(s, a)$$

- Osvježenje jednog koraka

- Učenje unatrag

$Q(A1, \downarrow)$	69
$Q(A1, \rightarrow)$	73
$Q(A2, \uparrow)$	66
$Q(A2, \rightarrow)$	81
$Q(B1, \leftarrow)$	66
$Q(B1, \downarrow)$	81
$Q(B1, \rightarrow)$	zabranjeno
$Q(B2, \leftarrow)$	0
$Q(B2, \uparrow)$	73
$Q(B2, \rightarrow)$	90 * $\gamma =$
$Q(C1, \downarrow)$	90
$Q(C1, \leftarrow)$	0
$Q(C2, \uparrow)$	100
$Q(C2, \leftarrow)$	81

- Primjer iz realnog svijeta:
- problem: idemo na more
- $r$  = sreća zbog obištenih lokaliteta i pojedene hrane



$$V^{\pi} = 306$$

$$\gamma = 0.9$$

# Nedeterminističke nagrade i akcije

- u stvarnom svijetu: neprecizni senzori, igre s bacanjem kocke...
- funkcija  $V$  – očekivanje umjesto točnog broja

$$V^{\pi}(s_t) = E \left[ \sum_{t=0}^{\infty} \gamma^i r_{t+1} \right]$$

$$Q(s, a) = r(s, a) + \gamma \max_a Q(\delta(s, a), a')$$

$$Q(s, a) = E[r(s, a)] + \gamma \sum_{s'} P(s' | s, a) \max_{a'} Q(s', a')$$





# Proširenja osnovnih konceptata

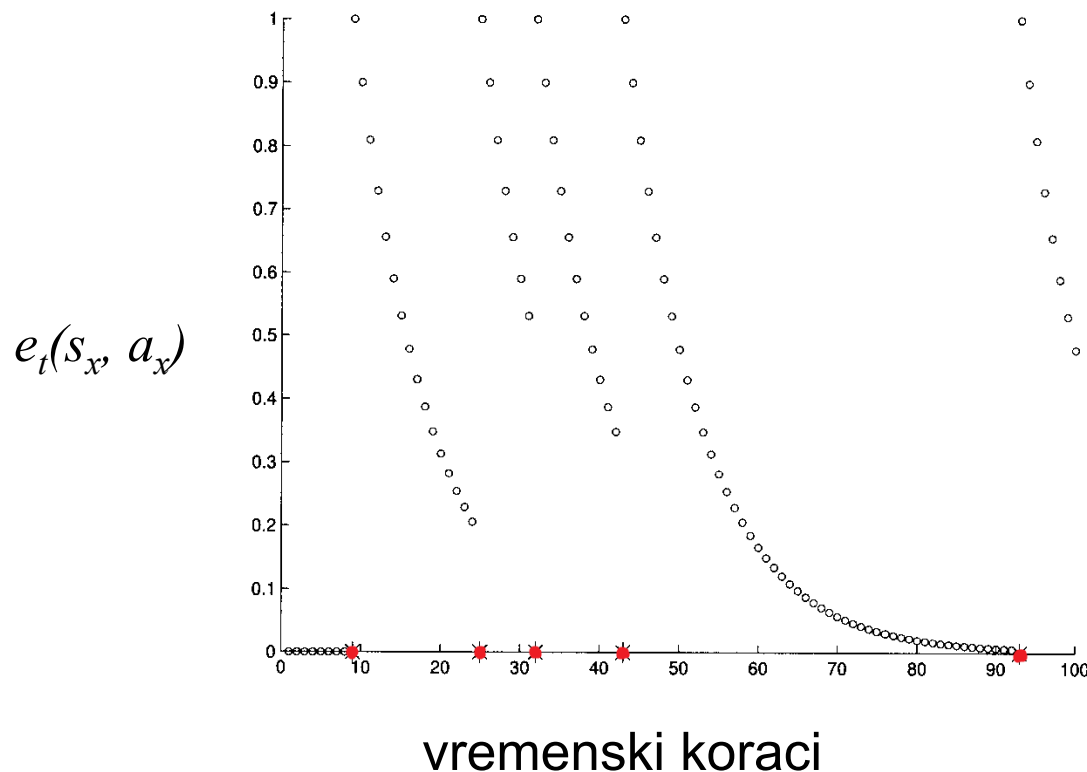
Goran Narančić

- pretpostavke dosad:
  - agent vidi stvarno stanje svijeta
  - nedeterministički jedino ishod akcije *a priori*
- agentova opažanja nisu nužno jednaka pravom stanju:
  - uvodi se vjerojatnost stanja  $p(s_i | o_i)$
  - označava vjerojatnost da za opažanje  $o_i$  svijet se nalazi u stanju  $s_i$
- uvodimo novi pojam: MDP s djelomično vidljivim stanjima

- sljedeće stanje u svijetu ovisi o stvarnom trenutnom stanju i agentovoj akciji
- dva stanja svijeta mogu rezultirati u jednakom opažanju od strane agenta
- Markovljevo svojstvo ne vrijedi za opažanja – sljedeće opažanje stanja ne ovisi isključivo o trenutnom opažanju i akciji
- potrebno uzeti u obzir putanju agenta (akcije i opažanja)
- dodatne akcije agenta – služe samo za skupljanje opažanja

- uvođenje internog stanja agenta: **stanje vjerovanja**  $b_t$  (engl. *belief state*)
- predstavlja agentovu procjenu trenutnog stanja
- agent koristi **procjenitelj stanja** (engl. *state estimator*) da osvježi stanje vjerovanja  $b_{t+1}$  na temelju:
  - trenutnog vjerovanja  $b_t$
  - zadnje akcije  $a_t$
  - trenutnog opažanja  $o_{t+1}$
- strategija odlučivanja  $\pi$  i svi algoritmi se koriste stanjem vjerovanja  $b_t$  umjesto stvarnog stanja  $s_t$

- engl. *eligibility traces*
- služe za određivanje starosti pojedine Q vrijednosti
- koriste se za smanjivanje utjecaja starih vrijednosti



- zvjezdice na x-osi: agent je izveo akciju  $a_x$  u stanju  $s_x$
- Alpaydin 2004, str. 386, slika 16.7

- algoritam Q učenja je dio skupine algoritama učenja iterativnim upoznavanjem svijeta (engl. *temporal difference*)
- ideja: smanjivati razliku između agentovih aproksimacija koje se izvedu u različitim trenucima
- dosad se gledalo jedan korak unaprijed – zašto ne gledati dva, tri ili više koraka unaprijed?
  - $Q^{(n)}(s_t, a_t) = r_t + \gamma r_{t+1} + \dots + \gamma^{(n-1)} r_{t+n-1} + \gamma^n \max Q(s_{t+n}, a)$

- Sutton (1988) uvodi generalnu metodu TD( $\lambda$ )
- konstanta  $\lambda$  ( $0 \leq \lambda < 1$ ) se koristi za spajanje procjena  $Q^{(n)}(s_t, a_t)$  za različite udaljenosti (različite  $n$ )
- Tesauro (1995) koristi jedan TD( $\lambda$ ) algoritam za izgradnju programa TD-Gammon za igranje igre Backgammon
- TD-Gammon igra na razini vrhunskog eksperta
- učenje algoritma: 1,5 milijuna odigranih partija protiv samog sebe

- svi algoritmi i pristupi dosad su se temeljili na izgradnji tablica koje sadrže vrijednost  $Q(s, a)$
- nauče se samo parovi stanja i akcija koji su isprobani tijekom učenja
- $Q$  vrijednosti za parove  $(s, a)$  koji nisu bili isprobani su ostali nepromijeni (0 ili nasumični broj, ovisi o inicijalnoj postavci)
- dobro učenje primjera – ali vrlo loša generalizacija!
- potpuno neprikladno za primjenu u aplikacijama za korištenje u stvarnom svijetu

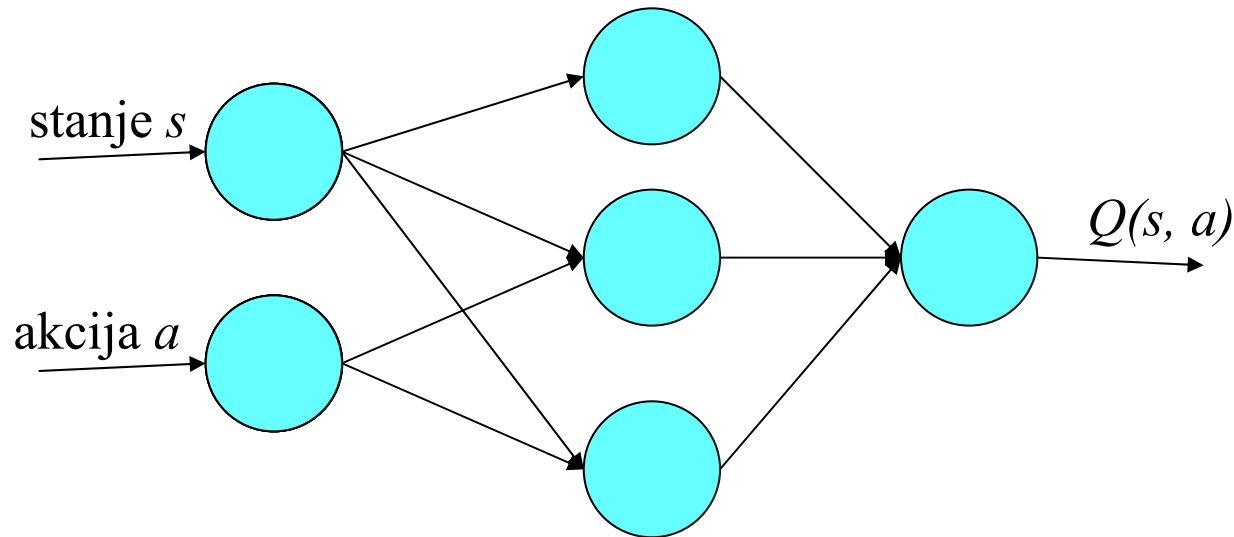


- moguća rješenja: proći kroz sve moguće parove (s, a)
  - obični “štreberski” algoritam
  - nepraktično za svjetove s velikim brojem stanja i akcija
  - nemoguće za kontinuirane svjetove
  - beskorisno za uvođenje u nove situacije
- metode gradijentnog spusta
  - linearne – teoretski zanimljive, dobri praktični rezultati
  - nelinearne – npr. neuronske mreže

- neuronske mreže korištene umjesto tablica za određivanje  $Q(s, a)$  vrijednosti
- učenje algoritmom širenja pogreške unazad (engl. *backpropagation*)
- problem: učenje vrijednosti za jedan par može poremetiti izlaz za neki drugi par
- TD-Gammon koristi neuronske mreže

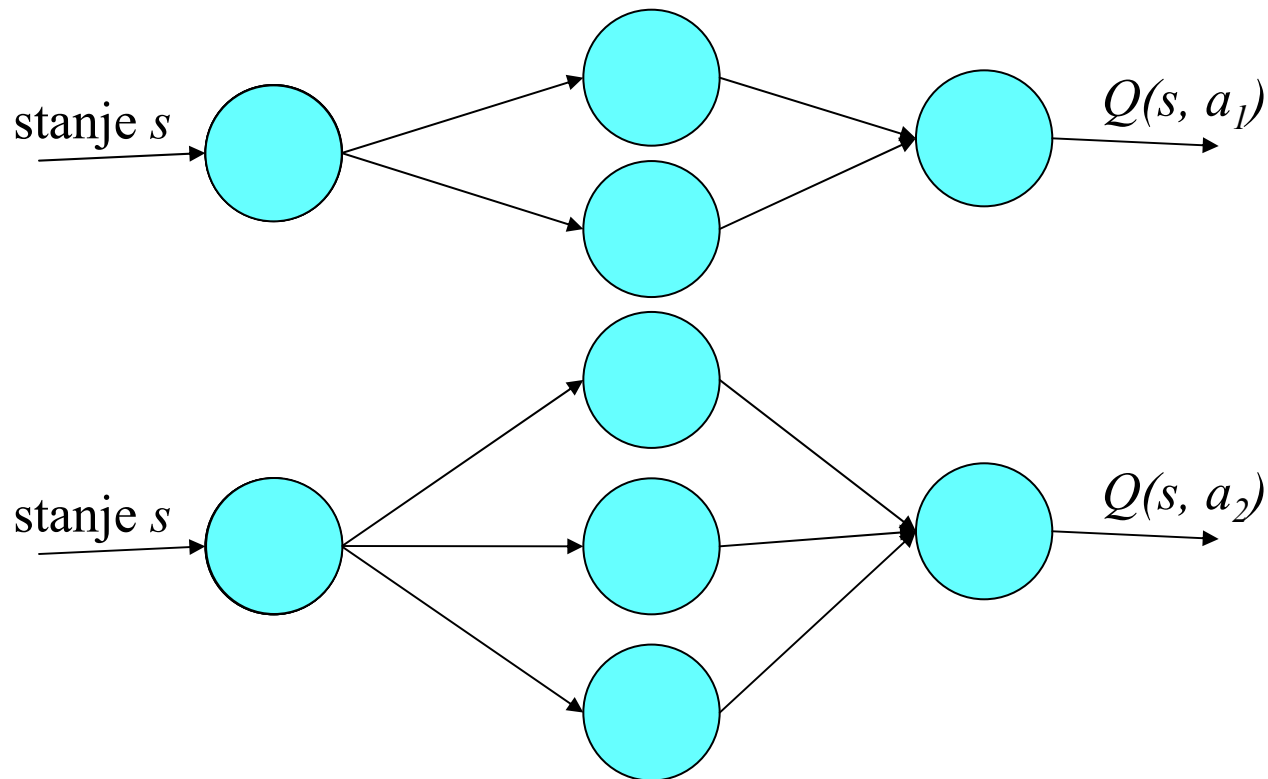
## Primjeri korištenja UNN (1)

- neuronska mreža prima stanje  $s$  i akciju  $a$ , a njen izlaz je vrijednost  $Q(s, a)$
- napomena: oblik mreža je samo primjer



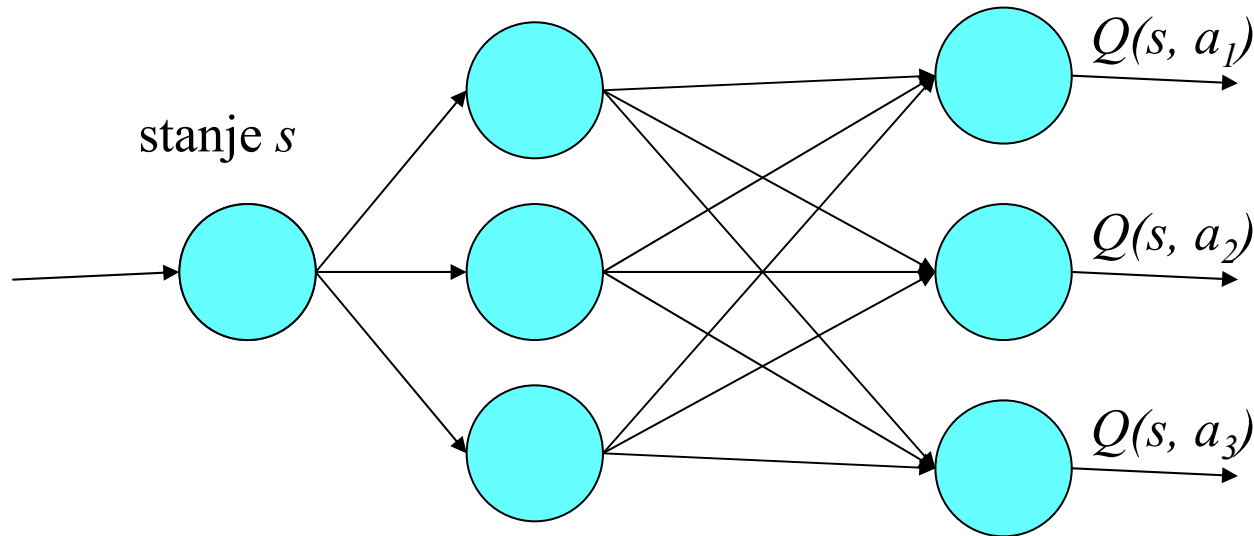
## Primjeri korištenja UNN (2)

- po jedna mreža za svaku akciju; mreža prima stanje  $s$  kao ulaz i vraća  $Q(s, a)$
- nešto uspješnija alternativa od prethodne



## Primjeri korištenja UNN (3)

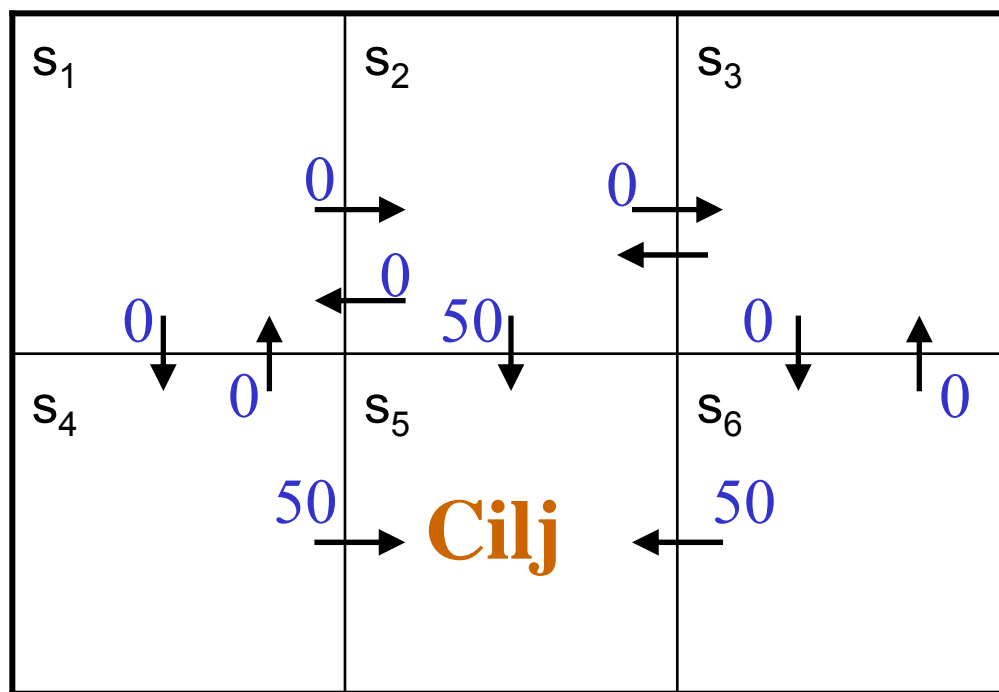
- mreža prima samo stanje  $s$ , a izlazi su joj vrijednosti  $Q(s, a)$  - uvijek vraća izlaze za svaku moguću akciju  $a$



# Primjeri

## 1. zadatak

Izračunati  $V(s)$  i  $Q(s, a)$  za zadani svijet. Primijeniti algoritam učenja s modelom.  $\gamma = 0,9$

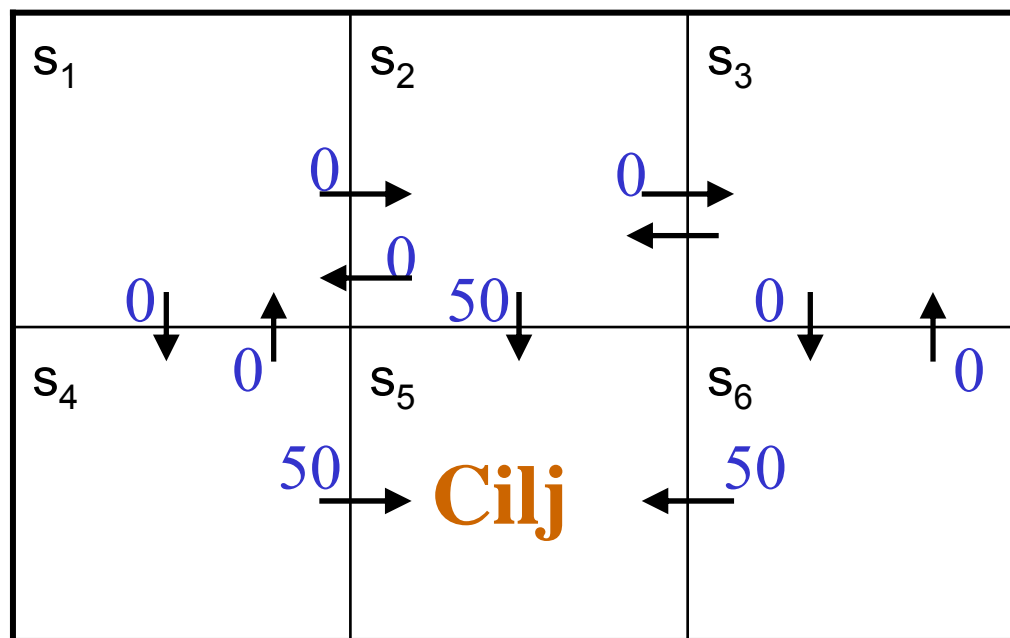


## 2. zadatak

Robot je pušten u isti svijet kao u prethodnom zadatku iz stanja  $s_1$ . Kretao se redom stanjima:  $s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow s_6 \rightarrow s_5$ . Provedi dvije iteracije učenja nad zadanom epizodom:

- učenje unaprijed (osvježavanje zadnjeg stanja)
- učenjem unazad

$\gamma = 0,9$





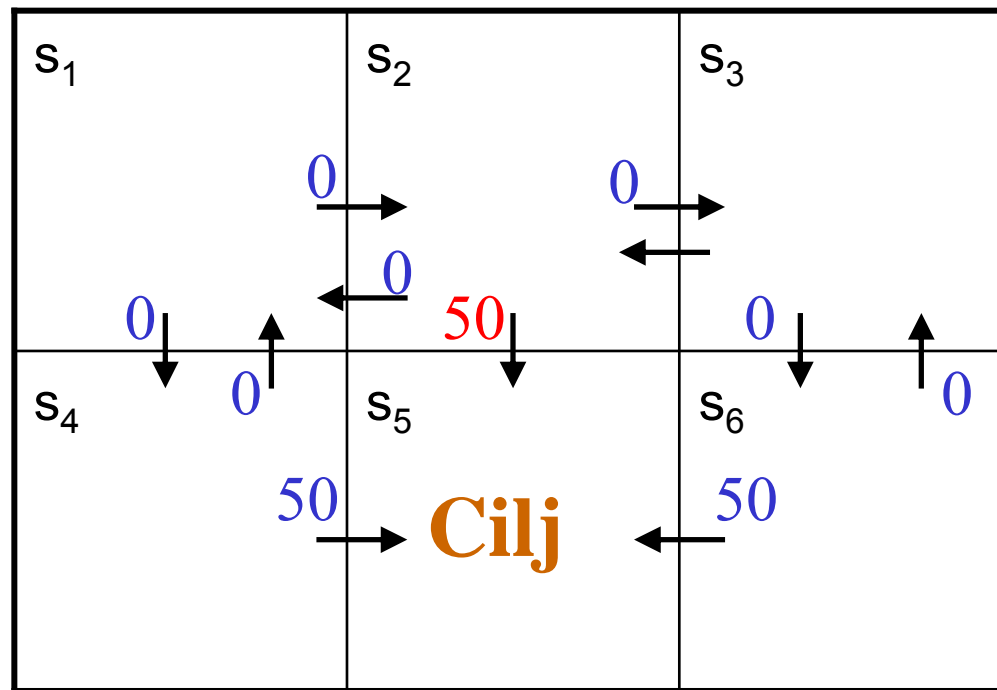
### 3. zadatak

Planiramo robota pustiti u jednostavan svijet s dva moguća apsorpcijska (ciljna) stanja  $G_1$  i  $G_2$ . Zadana nam je nagrada za prijelaz u stanje  $G_1$   $r(G_1, a) = 100$ , te  $\gamma = 0,9$ . Koliko minimalno mora biti  $r(G_2, a)$  da bi robot mogao naučiti da zaobiđe  $G_1$  i ode u  $G_2$ ?

Start			G1		G2

## Diskusijski primjer: odabir optimalne strategije

Imamo jednostavan svijet s zadanim vrijednostima  $Q(s,a)$ .  
Koja je optimalna strategija? Postoje li druge optimalne strategije?  
 $\gamma = 0,9$  a)  $Q(s_2, \downarrow) = 50$ ; b) a)  $Q(s_2, \downarrow) = 40$



- obvezatna literarura:
  - **T. M. Mitchell, Machine learning**, 1997.; poglavlje 13: Reinforcement learning
  - **E. Alpaydin – Introduction to machine learning**, 2004.; poglavlje 16: Reinforcement leaning
- ostala korištena literatura:
  - Sutton & Barto – Reinforcement learning, an introduction
  - T. Hrkać – IMAS, predavanja, Fer 2009.