Exemplu

Presupunem ca avem mediul din Figura 1.

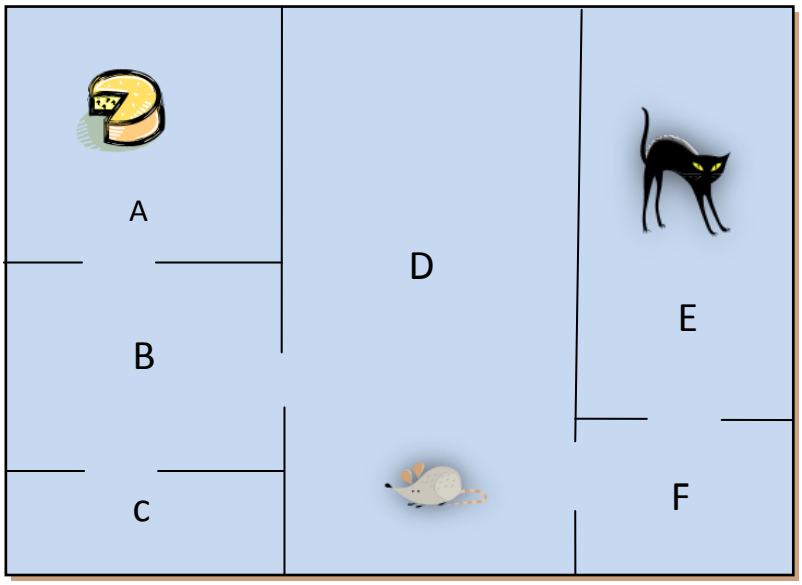

Figura 1. Mediul in care actioneaza agentul soarece

Exista asadar 6 incaperi etichetate de la A la F. Agentul care va trebui sa invete din acest mediu este soarecele, care se afla in imagine in incaperea D. In A se gaseste o bucata de cascaval, iar in E se gaseste o pisica. Soarecele este pus in diverse incaperi si trebuie sa se adapteze in asa fel incat sa ajunga la cascaval si, evident, sa nu ajunga la pisica. Cascavalul si pisica nu isi pot schimba incaperile de care apartin, dar soarecele da.

O stare in acest exemplu se identifica cu una din celulele din figura. Starea curenta este incaperea in care se gaseste soarecele, adica D in figura. Actiunile posibile se refera la deplasarea soarecelui in una din incaperile adiacente celei in care se afla. In figura, actiunile posibile ale soarecelui sunt deplasarea in celula B, respectiv in F. Putem modela figura de mai sus prin intermediul unui graf astfel:

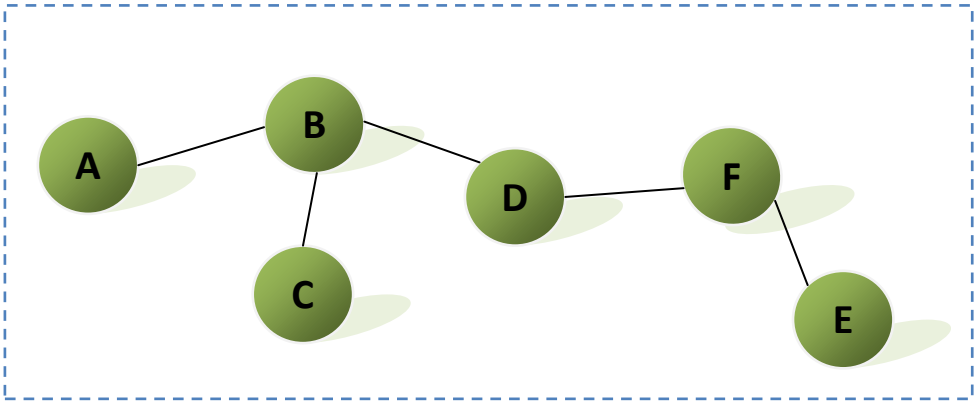

Figura 2. Reprezentarea conexiunilor dintre celule sub forma de graf

Soarecele poate fi pus in orice celula si va trebui sa ajunga in incaperea A si sa nu ajunga in E. Pentru a stabili acest scop, vom pune o recompensa de 100 pentru tranzitia de la celula B la cea cu cascaval, adica A, respectiv una de -100 pentru tranzitia de la F la incaperea cu pisica, adica E. Cum celelalte incaperi nu au conexiune directa cu A sau E, vor avea numai recompensa 0. Evident, se poate merge si din A in B, si din B in A, asadar vom avea cate un arc de la fiecare nod catre celalalt. Am adaugat cate un arc catre propria stare atat pentru A (cu recompensa 100), cat si pentru F (cu -100).

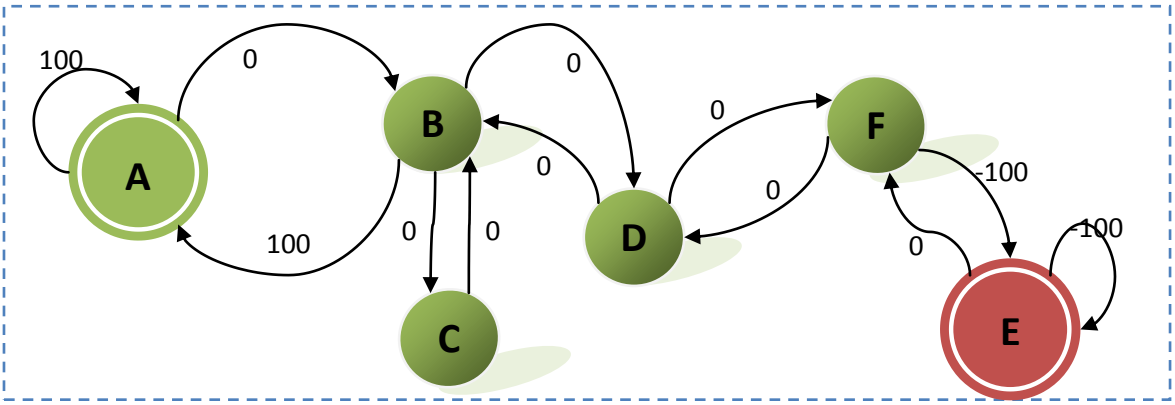

Figura 3. Graful care contine etichete cu valorile recompenselor

In exemplul ales, o stare se identifica cu o locatie, iar o actiune reprezinta mutarea agentului de la o stare la alta. In Figura 3, o stare se identifica cu un nod, iar o actiune cu o muchie. Daca presupunem ca agentul nostru se gaseste in starea D, el se poate deplasa in B pentru ca este conectata direct cu D, insa nu se poate muta in C pentru ca nu sunt concectate direct (nu exista muchie intre cele doua). Din B, agentul se poate deplasa in A, C sau inapoi in D, asadar sunt trei actiuni posibile din starea B.

Obtinem astfel o matrice a recompenselor R:

R =

		Actiuni					
		A	B	C	D	E	F
Stari	A	100	0	-	-	-	-
	B	100	-	0	0	-	-
	C	-	0	-	-	-	-
	D	-	0	-	-	0	-
	E	-	-	-	-	-100	0
	F	-	-	-	0	-100	-

De observat ca in matricea de mai sus starile sunt reprezentate pe linie, iar actiunile pe coloana: fiind in starea B – a doua linie a tabelului – actiunile posibile sunt A, C si D; valorile indicate in tabel sunt cele care apar si ca etichete pe muchiile care pleaca din B in Figura 3. Daca ne uitam in coloana etichetata cu B, avem actiunile care fac agentul sa intre in B, deci sunt alte valori: dupa cum se observa si in Figura 3, muchiile care intra in B au toate eticheta 0.

Construim in continuare matricea Q a calitatii care va reprezenta memoria agentului. Poate fi initializata in mod aleator, caz in care presupunem ca agentul este foarte debusolat, sau poate contine numai valori de zero, adica agentul nu stie nimic. Apoi, pe masura ce agentul invata, toate cunostintele se depun in matricea Q. La fel ca la matricea R, vom considera starile pe linie si actiunile pe coloana; o vom popula cu valori nule.

Q =

		Actiuni					
		A	B	C	D	E	F
Stari	A	0	0	0	0	0	0
	B	0	0	0	0	0	0
	C	0	0	0	0	0	0
	D	0	0	0	0	0	0
	E	0	0	0	0	0	0
	F	0	0	0	0	0	0

Algoritmul de invatare aplicat in continuare este cel de mai jos:

1. **Se initializeaza Q(s, a) in mod aleator sau cu 0**
2. **Repeta**
    1. **Initializeaza starea s**
    2. **Repeta**
        1. **Alege actiunea a in functie de strategia aleasa (ε-greedy sau Softmax)**
        2. **Executa actiunea a, observa r si s'**
        3. **$Q(s, a) = Q(s, a) + \alpha[R(s, a) + \gamma \max_{a'} Q(s', a') - Q(s, a)]$**
        4. **s = s'**
    3. **Pana cand s este stare terminala**
3. **Pana cand se intalneste conditia de oprire (un numar de iteratii)**

Stabilim valori pentru cei doi parametri ai algoritmului rata de invatare **α = 0.6** si factorul de reducere **γ = 0.8**. Daca factorul de reducere este aproape de 0, agentul va considera numai recompensa imediata; daca este aproape de 1, recompensele mai indepartata vor avea pondere mai mare, dorind sa amane astfel castigul.

Folosim strategia ε-greedy, cu **ε = 0.2** – probabilitatea ca o actiune sa fie aleasa in mod aleator.

Derularea experimentului

Presupunem ca agentul este asezat initial in celula D. Sunt doua posibilitati de deplasare din aceasta incapere, catre B si catre F; cum ambele au aceeasi valoare pentru recompensa, adica 0, una este aleasa in mod aleator, fie aceasta B. Executam actiunea, deci soarecele se afla in celula B. Urmatorul pas al algoritmului (2.2.3) este de a actualiza valoarea lui Q(D, B).

$$Q(\text{stare}, \text{actiune}) = Q(\text{stare}, \text{actiune}) + \alpha[R(\text{stare}, \text{actiune}) + \gamma \max_{a'} Q(\text{stare}', \text{actiune}') - Q(\text{stare}, \text{actiune})]$$
$$Q(D, B) = Q(D, B) + 0.6 * [0 + 0.8 * \max\{Q(B, A), Q(B, C), Q(B, D)\} - Q(D, B)]$$

Cum matricea Q este in intregime egala cu 0, Q(D, B) ramane nula. Revenim la linia 2.2.1 din algoritm si alegem actiunea care presupune mutarea catre A, presupunand ca nu s-a generat un

numar mai mic decat ε = 0.2, deci actiunea cu recompensa cea mai buna a fost aleasa. Ne deplasam in A, apoi calculam Q(B, A).

$$Q(stare, actiune) = Q(stare, actiune) + \alpha[R(stare, actiune) + \gamma max_{a'} Q(stare', actiune') - Q(stare, actiune)]$$
$$Q(B, A) = Q(B, A) + 0.6 * [100 + 0.8 * \max\{Q(A, A), Q(A, B)\} - Q(B, A)] = 0 + 0.6 * 100 + 0 - 0 = 60$$

Deoarece ne aflam in una din cele doua stari finale, se incheie aceasta etapa, adica ajungem la pasul 2.2.3 al algoritmului si va trebui sa ne intoarcem la pasul 2.1, adica va trebui sa introducem din nou agentul in una din celule, la intamplare. In acest moment, matricea Q arata astfel:

Q =

		Actiuni					
		A	B	C	D	E	F
Stari	A	0	0	0	0	0	0
	B	60	0	0	0	0	0
	C	0	0	0	0	0	0
	D	0	0	0	0	0	0
	E	0	0	0	0	0	0
	F	0	0	0	0	0	0

Se initializeaza agentul soarece in celula C. Are o singura posibilitate de deplasare de aici, in B. Calculam Q(C, B).

$$Q(stare, actiune) = Q(stare, actiune) + \alpha[R(stare, actiune) + \gamma max_{a'} Q(stare', actiune') - Q(stare, actiune)]$$
$$Q(C, B) = Q(C, B) + 0.6 * [0 + 0.8 * \max\{Q(B, A), Q(B, C), Q(B, D)\} - Q(C, B)] = 0.6 * (0.8 * 60) \approx 29$$

Din B, posibilitatile de deplasare sunt A, C si D. Se genereaza un numar real aleator intre 0 si 1 si se verifica daca este mai mic decat ε = 0.2. Presupunem ca nu este nici de aceasta data, deci se alege actiunea cea mai profitabila din punct de vedere al recompensei, adica A. Se face deplasarea si se calculeaza Q(B, A).

$$Q(stare, actiune) = Q(stare, actiune) + \alpha[R(stare, actiune) + \gamma max_{a'} Q(stare', actiune') - Q(stare, actiune)]$$
$$Q(B, A) = Q(B, A) + 0.6 * [100 + 0.8 * \max\{Q(A, A), Q(A, B)\} - Q(B, A)] = 60 + 0.6 * [100 - 60] = 84$$

A este stare finala, prin urmare si aceasta etapa se incheie. Matricea Q actualizata se modifica in:

Q =

		Actiuni					
		A	B	C	D	E	F
Stari	A	0	0	0	0	0	0
	B	84	0	0	0	0	0
	C	0	29	0	0	0	0
	D	0	0	0	0	0	0
	E	0	0	0	0	0	0
	F	0	0	0	0	0	0

Presupunem agentul initializata acum in celula F. Are doua posibilitati sa se mute, catre D sau catre E. Din nou se genereaza un numar intre 0 si 1 si se verifica daca este mai mic decat ε.

Presupunem ca avem noroc si nu este mai mic de 0.2, asadar se alege actiunea care ne duce catre o stare cu o recompensa mai buna, adica spre celula D (care are 0, fata de E care avea -100). Calculam in continuare Q(F, D):

$$Q(\text{stare}, \text{actiune}) = Q(\text{stare}, \text{actiune}) + \alpha[R(\text{stare}, \text{actiune}) + \gamma \max_{a'} Q(\text{stare'}, \text{actiune'}) - Q(\text{stare}, \text{actiune})]$$
$$Q(F, D) = Q(F, D) + 0.6 * [0 + 0.8 * \max\{Q(D, B), Q(D, F)\} - Q(F, D)] = 0$$

Din D posibilitatile sunt de a merge in B sau inapoi in F. Ambele au recompense nule, se alege la intamplare si de aceasta data se intampla sa fie selectionat F. Se face deplasarea inapoi si se calculeaza Q(D, F).

$$Q(\text{stare}, \text{actiune}) = Q(\text{stare}, \text{actiune}) + \alpha[R(\text{stare}, \text{actiune}) + \gamma \max_{a'} Q(\text{stare'}, \text{actiune'}) - Q(\text{stare}, \text{actiune})]$$
$$Q(D, F) = Q(D, F) + 0.6 * [0 + 0.8 * \max\{Q(F, D), Q(F, E)\} - Q(D, F)] = 0$$

Avem optiunile de deplasare catre D sau E. Este generat din nou un numar in intervalul [0, 1], presupunem ca este ales 0.1 care este mai mic decat ε = 0.2, asadar se alege mutarea la celula E. Calculam Q(F, E).

$$Q(\text{stare}, \text{actiune}) = Q(\text{stare}, \text{actiune}) + \alpha[R(\text{stare}, \text{actiune}) + \gamma \max_{a'} Q(\text{stare'}, \text{actiune'}) - Q(\text{stare}, \text{actiune})]$$
$$Q(F, E) = Q(F, E) + 0.6 * [-100 + 0.8 * \max\{Q(E, E), Q(E, F)\} - Q(F, E)] = 0.6 * (-100) = -60$$

Se incheie si aceasta etapa nu prea norocoasa pentru agentul nostru, iar matricea Q va arata astfel:

Q =

		Actiuni					
		A	B	C	D	E	F
Stari	A	0	0	0	0	0	0
	B	84	0	0	0	0	0
	C	0	29	0	0	0	0
	D	0	0	0	0	0	0
	E	0	0	0	0	0	0
	F	0	0	0	0	-60	0

Avem o noua initializare a agentului in celula D. Avem posibilitatile de deplasare in B sau in F; se alege in mod aleator cu o distributie uniforma celula B. Calculam valoarea pentru Q(D, B).

$$Q(\text{stare}, \text{actiune}) = Q(\text{stare}, \text{actiune}) + \alpha[R(\text{stare}, \text{actiune}) + \gamma \max_{a'} Q(\text{stare'}, \text{actiune'}) - Q(\text{stare}, \text{actiune})]$$
$$Q(D, B) = Q(D, B) + 0.6 * [0 + 0.8 * \max\{Q(B, A), Q(B, C), Q(B, D)\} - Q(D, B)] = 0.6 * (0.8 * 84) \approx 40$$

Din B, avem trei posibilitati, A, C si D. Se genereaza un numar care nu este mai mic decat 0.2, deci se alege cea mai profitabila actiune, adica mutarea in celula A. Se actualizeaza si Q(B, A), dupa care se incheie si aceasta etapa.

$$Q(\text{stare}, \text{actiune}) = Q(\text{stare}, \text{actiune}) + \alpha[R(\text{stare}, \text{actiune}) + \gamma \max_{a'} Q(\text{stare'}, \text{actiune'}) - Q(\text{stare}, \text{actiune})]$$
$$Q(B, A) = Q(B, A) + 0.6 * [0 + 0.8 * \max\{Q(A, A), Q(A, B)\} - Q(B, A)] = 84 + 0.6 * (100 - 84) \approx 94$$

Experimentul poate continua pentru un numar mult mai mare de iteratii. Numai si dupa pasii urmati in acest exemplu, se obtine o matrice Q care este suficient de explicita pentru agentul soarece astfel incat sa ajunga la cascaval, indiferent in ce celula este initializat (exceptie, bineinteles incaperea E care contine pisica). Matricea Q obtinuta dupa iteratiile de pana acum este cea de mai jos:

Q =

		Actiuni					
		A	B	C	D	E	F
Stari	A	0	0	0	0	0	0
	B	94	0	0	0	0	0
	C	0	29	0	0	0	0
	D	0	40	0	0	0	0
	E	0	0	0	0	0	0
	F	0	0	0	0	-60	0

Aceasta poate fi interpretata prin graful din Figura 4. Cum foloseste agentul aceasta matrice? In acest punct am terminat cu algoritmul de invatare, iar agentul soarece se ghideaza numai dupa matricea Q sau echivalentul sau, graful din Figura 4. Din orice stare s-ar afla, el alege actiunea care il duce intr-o alta stare cu cea mai mare valoare.

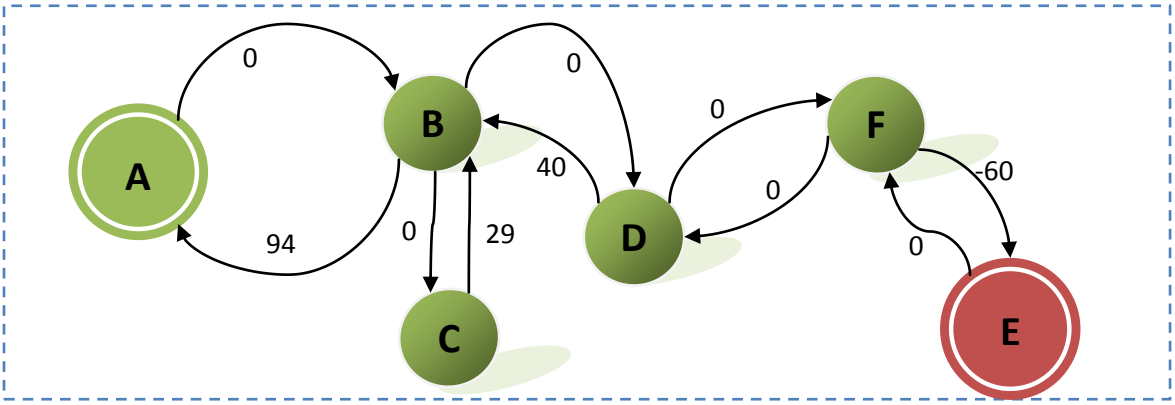

Figura 4. Graful care contine etichete cu valorile matricei de memorie Q

Pentru a fi si mai clara utilizarea matricei Q, agentul se ghideaza dupa algoritmul de mai jos.

1. **Starea in care este initializat agentul devine stare curenta.**
2. **Cat timp starea curenta este diferita de starea tinta**
   a. **De la starea curenta, se alege actiunea care produce valoarea maxima pentru Q.**
   b. **Starea curenta devine starea care a fost aleasa la pasul 2.**
   c. **Daca starea curenta nu este stare tinta, mergi la pasul 2**
3. **Sfarsit cat timp**

Presupunem ca este initializat agentul in celula D. Alege actiunea cu valoare maxima pentru Q, adica B (40 > 0, 0 corespunde actiunii care presupunea mutarea in F). Din B, se alege natural A (valoarea 94 este aleasa fata de 0 care ar fi fost mutarea catre C). Altfel, daca este initializat in C, se alege B, apoi A. Daca ar fi initializat in F, este aleasa actiunea cu valoarea 0, fata de cea cu valoarea -60, adica merge in D, apoi B si, in sfarsit, A.

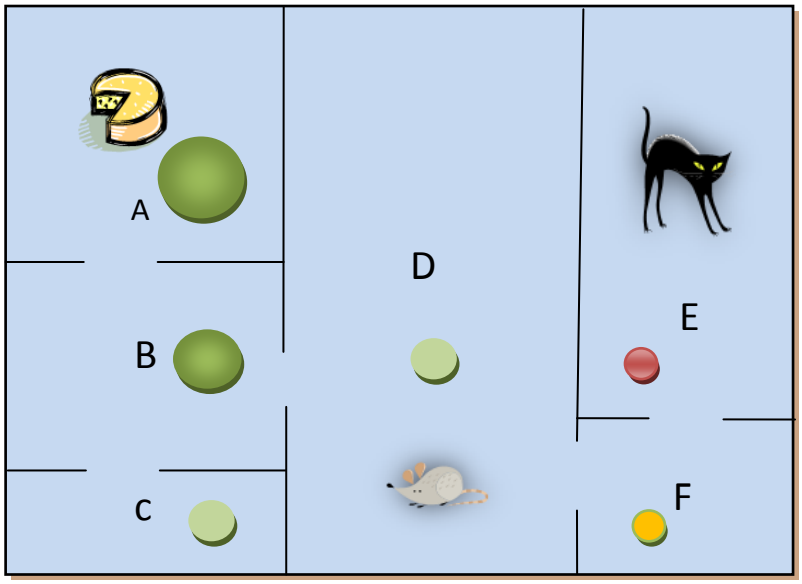

Figura de mai sus sugereaza care este tendinta agentului nostru, anume de a merge mai mult catre celule cu bule mari si verzi, si mai putin catre cele galbene sau rosii.