CHAPITRE 4

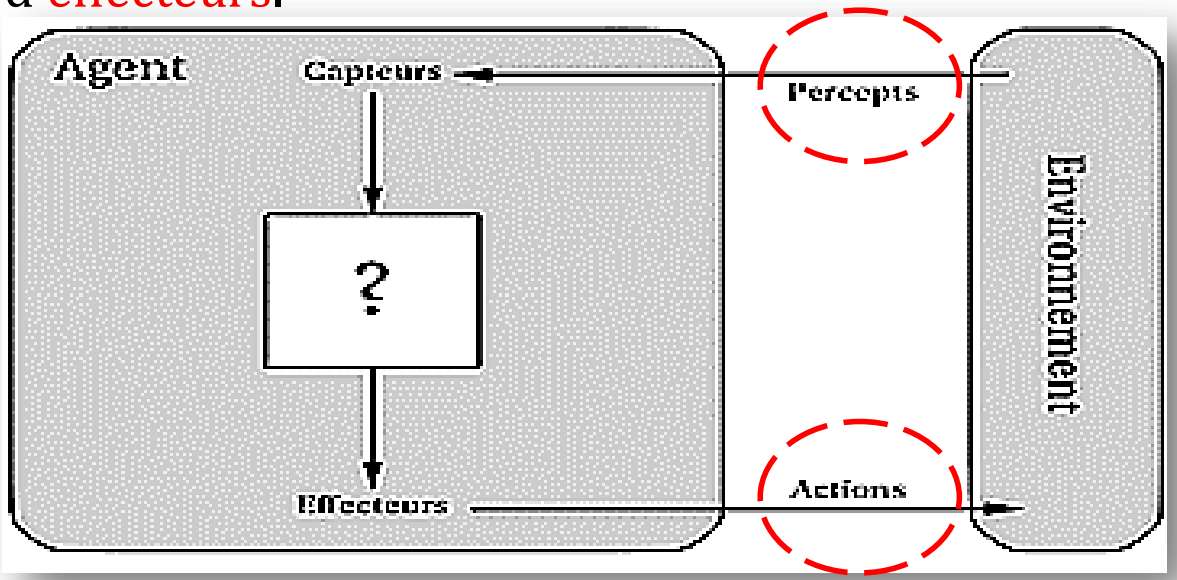
Apprentissage par Renforcement

A Goal Directed Learning from Interaction

1. Rappel sur les Agents Intelligents

- ✓ **PEAS**

Un agent est Tout ce qui peut être vu comme percevant son environnement au travers de capteurs et agissant sur cet environnement au travers d'effecteurs.



Environnement :

- Univers dans lequel l'agent évolue et effectue ses tâches...
- Tout ce qui est en dehors du contrôle absolu de l'agent ..

Fonction agent:

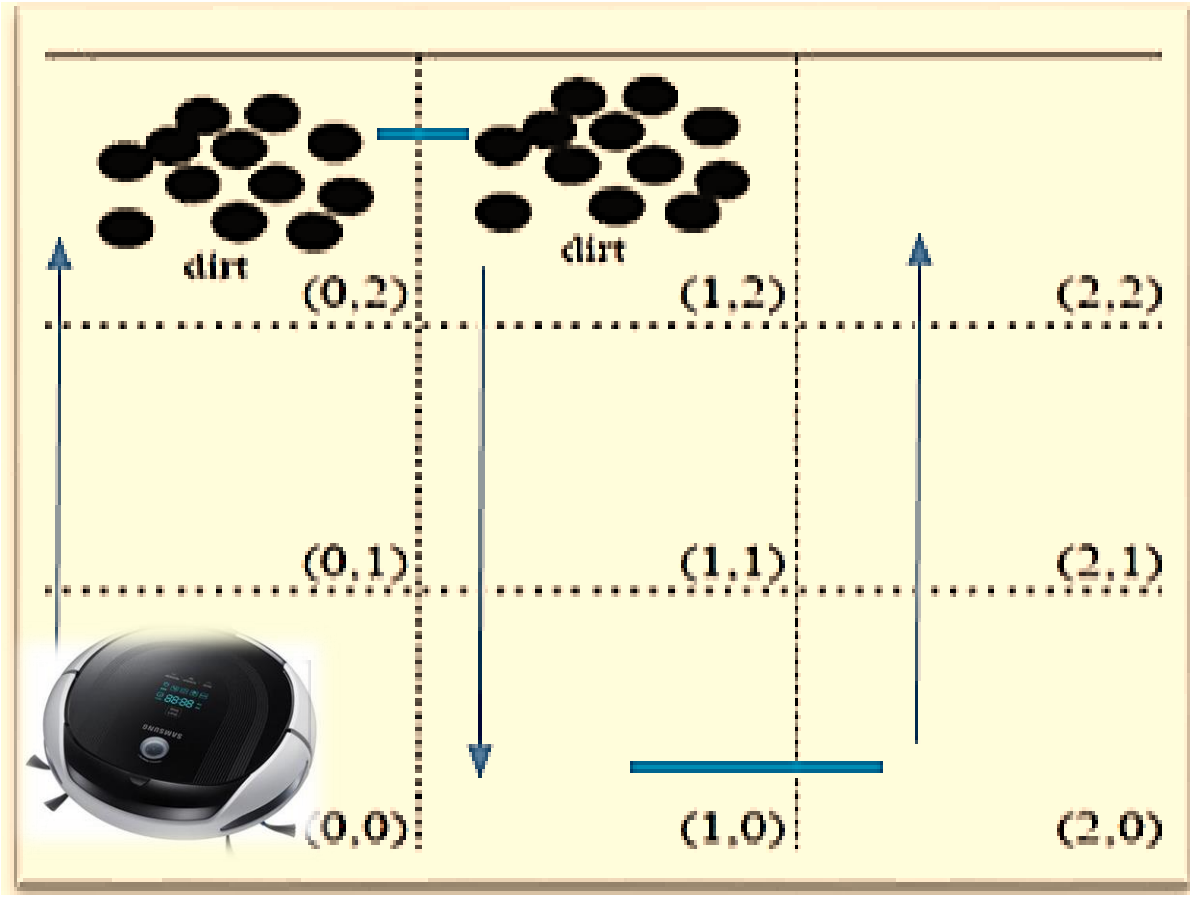
- Détermine comment associer aux états des actions **??**

L'environnement de la tâche se défini par la combinaison **PEAS**:

- ***Mesures de Performance*** :quelles sont les qualités désirables pour notre agent? Selon quelle mesure peut on dire que l'agent a accomplit sa tâche avec succès ? …
- ***Environnement***: avec quoi interagit l'agent? l'environnement est à l'origine des percepts et subit un changement suite aux actions de l'agent….
- ***Effecteurs*** : dispositifs via lesquels l'agent modifie **l'état de son environnement** …
- ***Capteurs*** : dispositifs via lesquels l'agent perçoit **l'état de son environnement** …

PEAS : Robot chauffeur de Taxi ?

PEAS : Robot Aspirateur ?



1. Rappel sur les Agents Intelligents

- ✓ **PEAS**
- ✓ **Propriétés de l'environnement**

Propriété de l'environnement de la tâche

- Mono-agent
- Multi-agent (concurrentiel ou coopératif)

Multi-agent

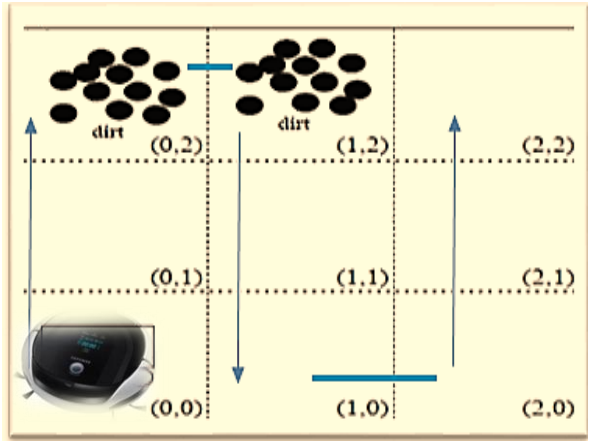
Mono-agent

Propriété de l'environnement de la tâche

- **Discret** : s'il existe un nombre fini de **percepts/états et d'actions** alors l'environnement est discret.
- **Continu**, Sinon .

Continu

Si on considère:
État : localisation, vitesse actuelle …
Actions : changer de vitesse, changer de direction



Discret

État : (S, P, D)
Actions : Aspirer , Avancer, Tourner

Propriété de l'environnement de la tâche

- **Déterministe** : l'état du monde à l'instant suivant est complètement déterminé par l'état courant et par l'action exécutée par l'agent.
- **Stochastique.**

Stochastique

Déterministe

État : (S, P, D)
Actions : Aspirer , Avancer, Tourner

Propriété de l'environnement de la tâche

- Dans un environnement **épisodique,** la vie de l'agent est divisée en **épisodes**. Ce qui se passe dans l'épisode ne dépend pas de ce qui s'est passé dans les épisodes précédents.
- **Séquentiel:** la décision courante est susceptible d'affecter les décisions futures.

Séquentiel

Épisodique

Jeu composé de plusieurs parties

Entièrement observable ?

- Russell & Norvig « *Un environnement de tâche est entièrement observable si, à chaque instant, les capteurs détectent tous les éléments nécessaires pour le choix de l'action ; Inversement s'il y a du bruit, de l'imprécision ou une information qui manquent concernant* ***l'état de l'environnement*** *alors l'environnement est dit partiellement observable.* »
- Wooldridge adopte le terme **accessible**: «*Un environnement accessible est un environnement dans lequel l'agent peut obtenir des informations complètes, précises et à jour sur* ***l'état de l'environnement*** **».**

Partiellement observable

Entièrement observable

Statique?

- Russell & Norvig : « *Si l'état de l'environnement ne change pas* ***durant la prise de décision*** *alors le système est dit statique … Il est dynamique sinon*.»
- Wooldridge: «*Un environnement statique est un environnement qui est supposé rester* ***inchangé sauf par l'exécution d'actions par l'agent****. Un environnement dynamique est un environnement sur lequel agissent d'autres processus et qui, par conséquent, change de manière indépendante de l'agent.*»

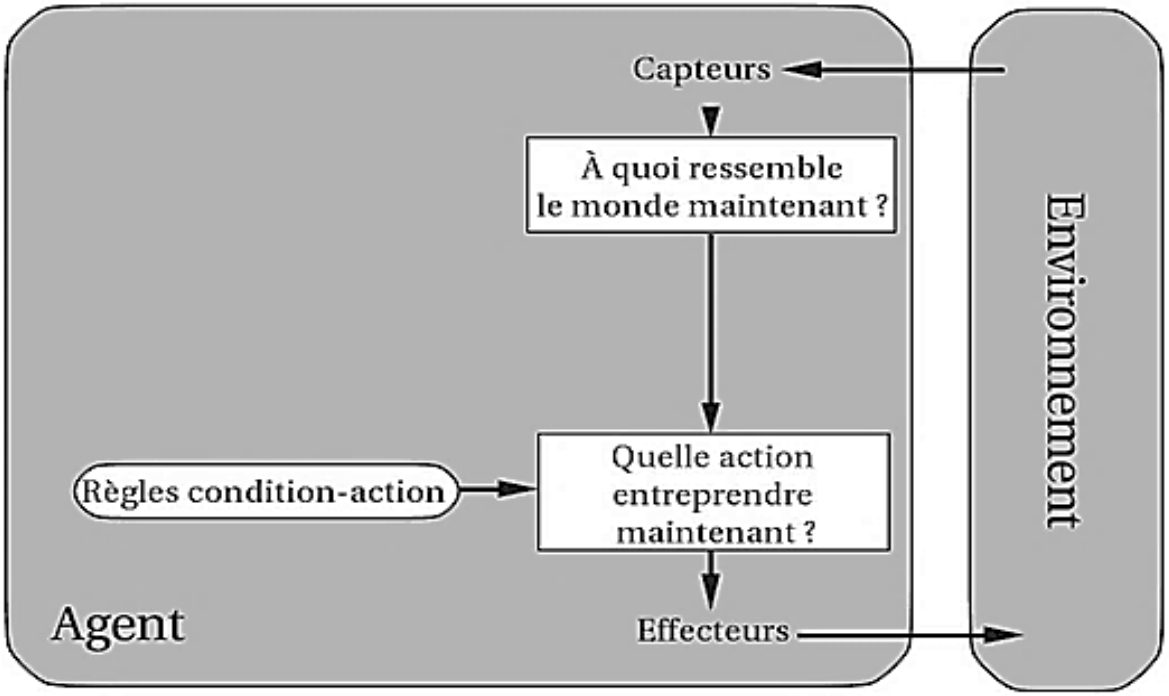
Dynamique

Statique

1. Rappel sur les Agents Intelligents

- ✓ **PEAS**
- ✓ **Propriétés de l'environnement**
- ✓ **Types des agents (selon la fonction agent)**

Agent réflexe simple

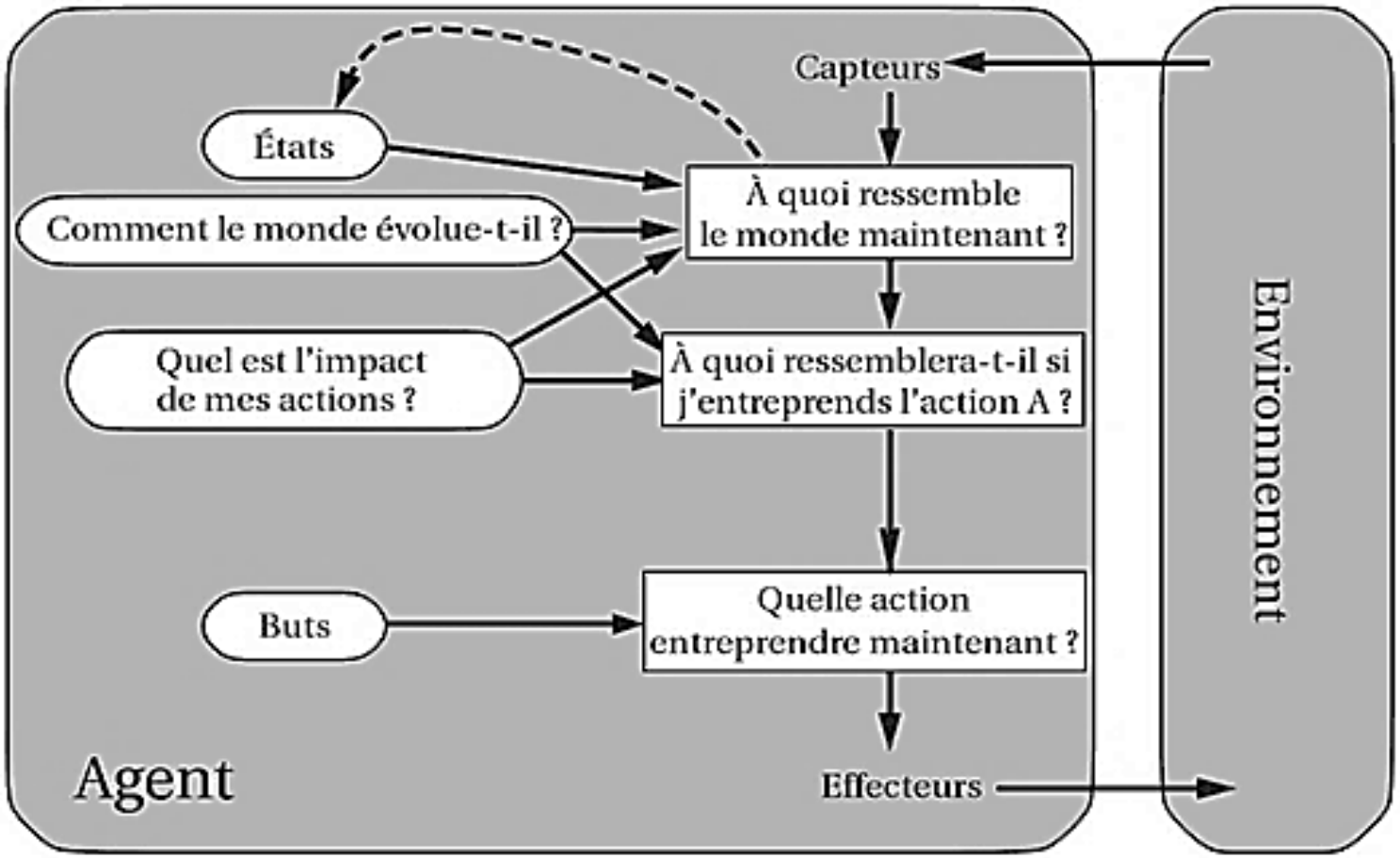


Le **thermostat d'un chauffage** est muni d'un capteur pour détecter la température ambiante. Imaginons les règles de fonctionnement:

Si la Température est trop basse Alors allumer le chauffage

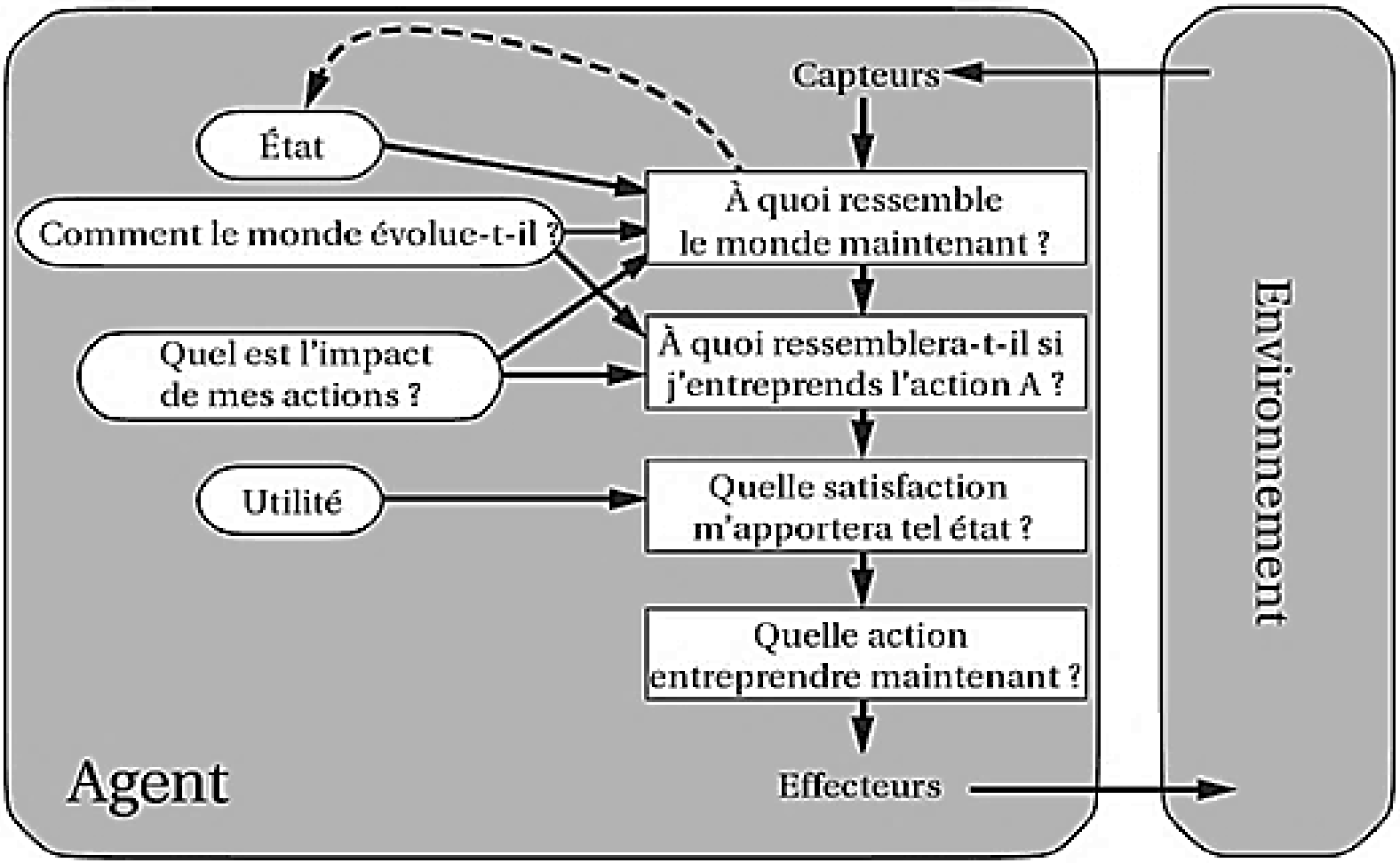
Si la Température est bonne Alors éteindre le chauffage

Agent basé sur les Buts



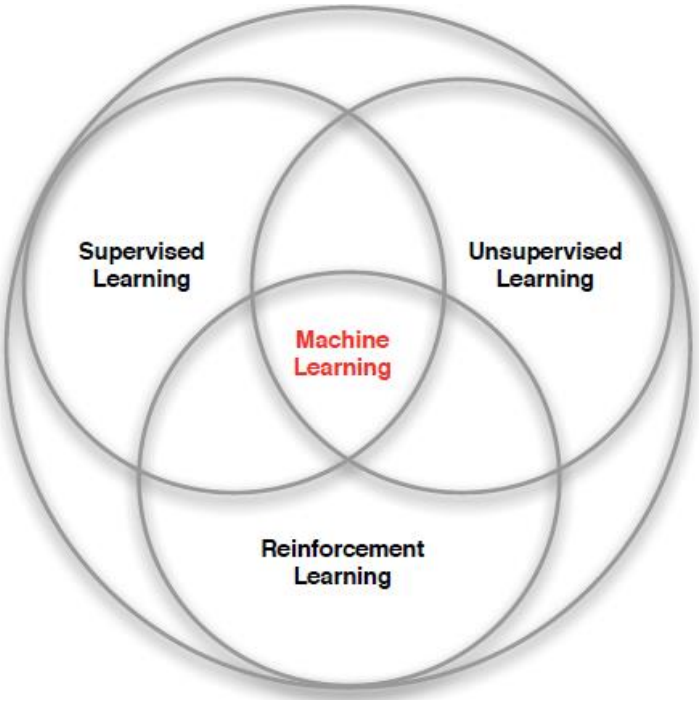
Un agent d'apprentissage par renforcement est guidé par un objectif

Agent basé sur l'utilité



Un agent d'apprentissage par renforcement est basé sur l'utilité

2. Introduction à l'apprentissage par renforcement



Applications (Quand ?)

Apprendre à résoudre des **problèmes complexes** :

1. Il nous est impossible d'obtenir des exemples de comportement souhaité qui soient à la fois corrects et représentatifs **de toutes les situations** dans lesquelles l'agent doit agir. Par exemple:
   - ✓Faire apprendre à un programme de jouer aux échecs
   - ✓Faire apprendre un robot humanoïde de marcher
2. En revanche, il est possible d'indiquer à l'agent s'il a gagné ou perdu:
   - ✓ gagner ou non la partie du jeu *(signal après une séquence d'actions)*
   - ✓ ne pas tomber, éviter des obstacles , avancer *(signal après chaque action)*

Applications (Exemples)

- Jeux de société, Jeux de carte, Jeux vidéos
- Santé, Par exemple:
  - Régimes de traitement dynamiques (maladies chronique & prise en charge des cas critiques)
  - Diagnostic automatique (à base de données médicales structurées & non structurées)
- Robotique , Par exemple:
  - Véhicules autonomes (Piloter un hélicoptère, Conduire une voiture)
  - Contrôle de robots manipulateurs comme les Robots chirurgicaux
- Traitement du langage naturel , Par exemple:
  - Résumé automatique
  - Systèmes de questions/réponses automatiques
- Finance/Gérer un portefeuille d'investissement, Energie/ Contrôler une centrale électrique , Réseaux de télécoms/Routage , etc.

Origine : Apprentissage par «essai et erreur»

- Études sur la psychologie de l'apprentissage chez l'animal.
- Edward Lee Thorndike (1911) , **Loi de l'effet** :

  « La probabilité qu'une action soit répétée augmente quand elle est suivie d'une conséquence agréable ou satisfaisante »

Autrement :

*« Lorsqu'un comportement est **récompensé**, il a plus de chance de se reproduire».*

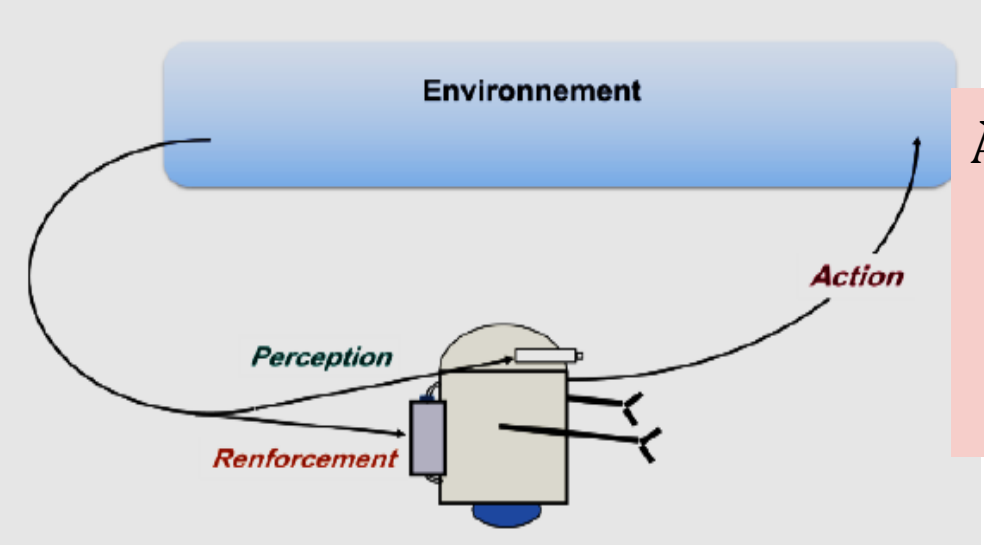
Une définition de l' apprentissage par renforcement

Recherche par un **agent autonome** d'une **politique** de décision **optimale**. Ceci via des **interactions** de genre « **essai-erreur** » avec son **environnement** suite desquelles il est **récompensé** ou **pénalisé**. L'agent apprend la meilleure politique, qui est la séquence des actions qui **maximisent** sa **récompense totale**.

- ✓ Interaction
- ✓ Signal de renforcement
- ✓ Optimisation **(agent basé utilité)**
- ✓ Exploration (Essai …)
- ✓ …….

Interaction avec son environnement / problèmes en boucle fermée

L'agent apprenant est supposé en interaction avec son environnement via trois canaux distincts :



À chaque étape t, l'agent :

- ❑ Exécute une action a_t
- ❑ Reçoit une observation o_t
- ❑ Reçoit une récompense r_t

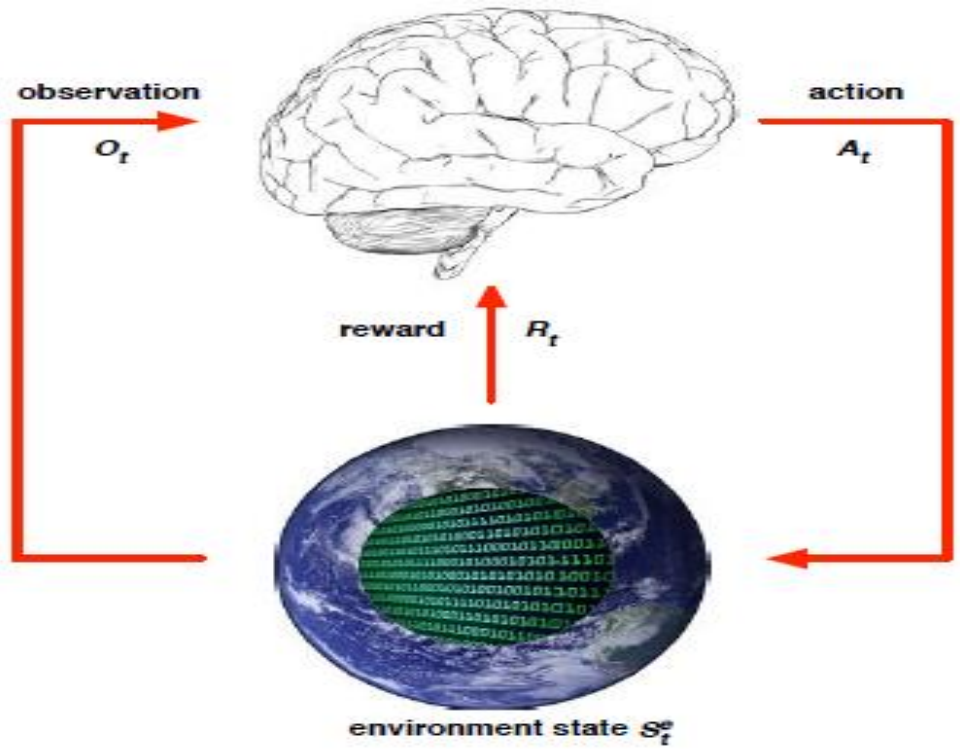
1. canal perceptif par lequel l'agent mesure **l'état** dans lequel il se trouve dans l'environnement ;
2. canal qui transmet à l'environnement **l'action** choisie par l'agent.
3. canal spécifique aux **signaux de renforcement** via lequel l'agent est informé sur la qualité de ses actions

Exemple : un Robot dans un labyrinthe.

Le robot peut se déplacer dans l'une des quatre directions de la boussole et doit effectuer une séquence de mouvements pour atteindre la sortie. Tant que le robot est dans le labyrinthe, il n'y a pas de retour et le robot essaie de nombreux mouvements jusqu'à ce qu'il atteigne la sortie et ce n'est qu'alors qu'il obtient une récompense.

- ✓Objectif : atteindre la sortie
- ✓Environnement: la labyrinthe
- ✓Agent: robot
- ✓État: position du robot dans la labyrinthe
- ✓Actions : se déplacer vers le Nord, Est, Ouest, Sud sans heurter les murs,
- ✓récompense : pas immédiatement mais retardée **(mais pas toujours)**

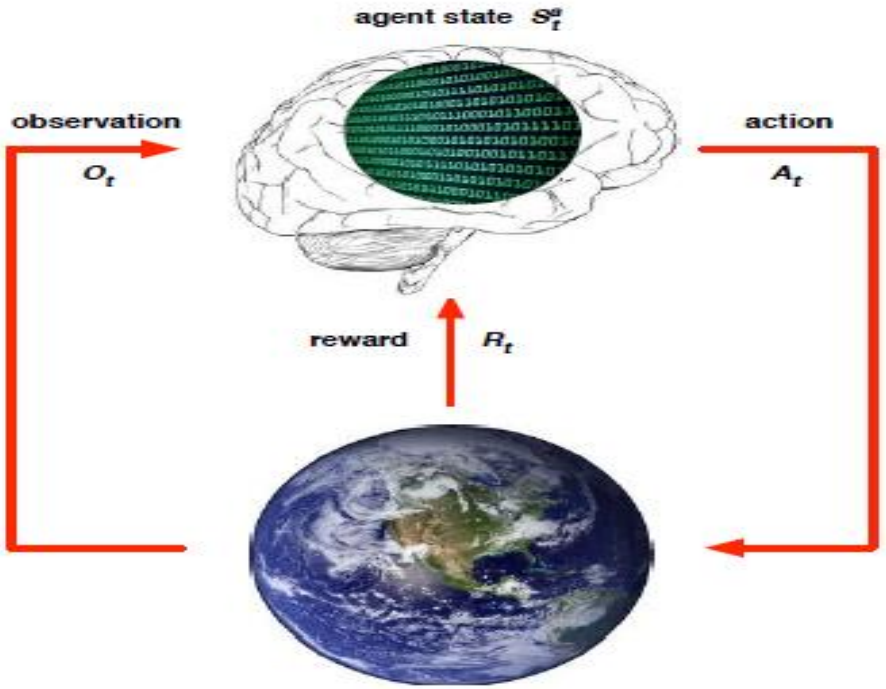
État de l'environnement S^e_t



Représentation privée à l' environnement

- N'importe quelle information qu'exploite l'environnement pour sélectionner le prochain observation/récompense
- Généralement caché/inconnu pour l'agent
- Même si connu, peut contenir des informations inappropriées/inutiles

État de l'agent $S^a{}_t$



Représentation interne de l'agent

- N'importe quelle information qu'exploite l'agent pour sélectionner une action
- Information utilisée par les algorithmes d'apprentissage par renforcement
- Fonction de l'historique (*séquence des observations, actions, récompenses*) :

$$S_t = f(H_t) \quad \text{et} \quad H_t = O_1, r_1, a_1, \ldots, a_{t-1}, O_t, r_t$$

- **Environnement complètement observable**

L'agent peut directement observer l'état de l'environnement . Ceci veut dire:

$$S_t{}^a = O_t = S_t{}^e$$

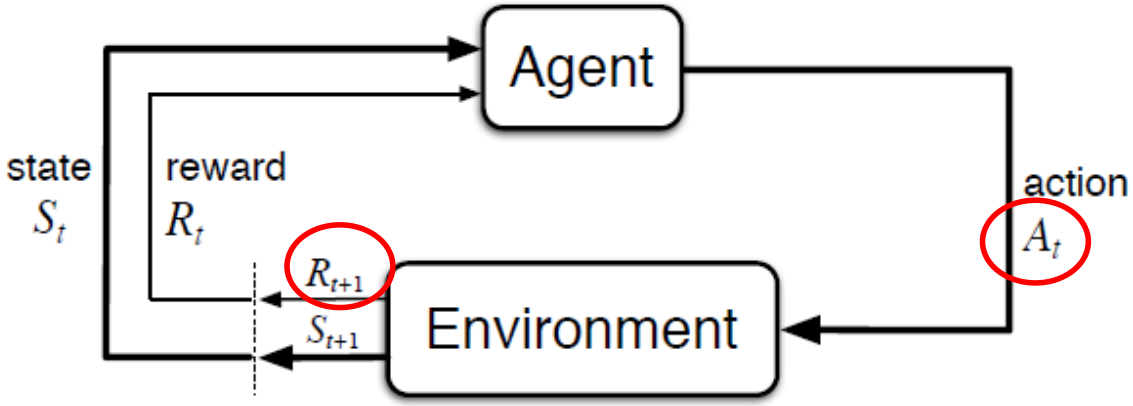
- **Environnement partiellement observable**

$$S_t{}^a \neq S_t{}^e$$

Récompense / pénalité

Hypothèse de la récompense :

« *Tout objectif peut être décrit par la* ***maximisation*** *de l'espérance des* ***récompenses*** *cumulées.* »



- Signal (valeur ***scalaire***) en provenance de l'environnement .
- Indique à quel point l'agent s'est **bien comporté** .
- L'agent n'est pas sensé réellement atteindre la récompense maximale. Il essaie plutôt d'**augmenter la somme** des récompenses qu'il reçoit.

Récompense / pénalité

1. Faire apprendre à un robot à s'échapper d'un labyrinthe rapidement
   - ✓ ***-1*** *pour chaque étape de temps passée dans le labyrinthe*
2. Faire marcher un robot humanoïde
   - ✓ ***+ r*** *pour le mouvement vers l'avant*
   - ✓ ***- r*** *pour tomber*
3. Apprendre à jouer aux échecs
   - ✓ *+1 pour gagner*
   - ✓ *-1 pour perdre*
   - ✓ *0 pour toutes les positions non terminales.*

Retour ou récompense totale G_t

- Si l'interaction de l'agent avec son environnement est épisodique et T est l'étape terminale , alors:

$$G_t = R_{t+1} + R_{t+2} + \cdots + \quad R_T$$

- Si la tâche est continue (autrement $T = \infty$) , il faut introduire un **facteur d'actualisation**. $0 \leq \gamma \leq 1$. L'idée est qu'une récompense r disponible dans n étapes correspond à une récompense $\gamma^n r$ disponible immédiatement.

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

- **Exemple numérique:** Considérer une récompense de **50** ; Calculer sa valeur actualisée si k=1,2,3,4,10,20 pour γ=0.8 et pour γ=0.1

Donc ..

0,8	1	2	3	4	10	20
	40	32	25,6	20,48	5,3687091	0,5764608
0,1	5	0,5	0,05	0,005	5E-09	5E-19

✓plus γ est proche de 0 : plus on s'intéresse aux récompenses immédiates.

✓Aussi, lorsque k tend vers ∞ , les futures récompenses tendent vers 0.

Dilemme : Exploration vs Exploitation

- **Exploitation:** Pour obtenir le maximum de récompense, un agent d'apprentissage par renforcement doit privilégier les actions qu'il a essayées dans le passé et qu'il a trouvées efficaces pour produire une récompense.
- **Exploration:** Mais pour découvrir de telles actions, il doit essayer des actions qu'il n'a pas sélectionnées auparavant.

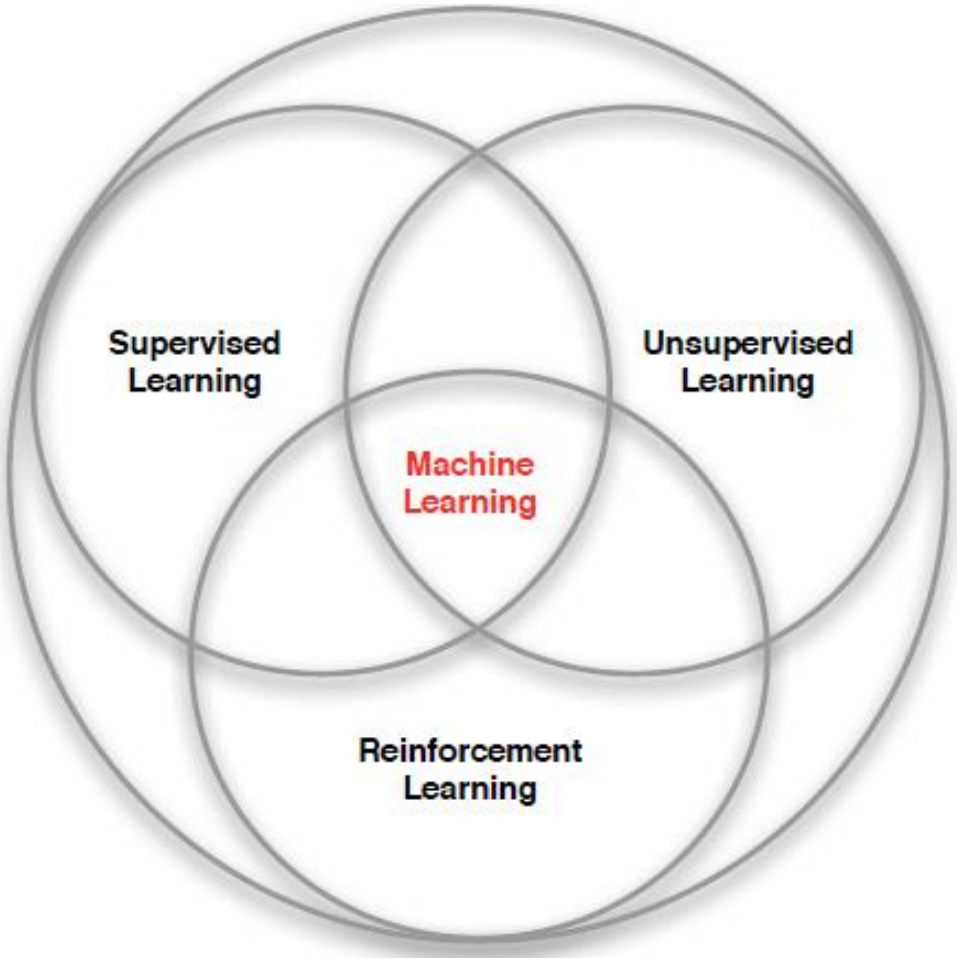
L'agent doit exploiter ce qu'il sait déjà pour obtenir une récompense, mais il doit également explorer afin de faire de meilleures sélections d'actions à l'avenir.

Exemples de règles de sélection d'actions

- **La règle ε–greedy:**
  - ✓Sélectionne souvent la meilleure action.
  - ✓Sélectionne uniformément une autre action dans A, avec une petite **probabilité ε.**
- **Règles Softmax**: la plus haute probabilité est attribuée à l'action gloutonne tandis que le reste des actions sont ordonnées selon leurs fonctions d'utilité Q_t. Une règle Softmax souvent utilisée est celle de ***Boltzmann*** où une action a est choisie à l'instant t avec la probabilité:

$$e^{Q_t(s,a)/\tau} / \sum_{b \in A} e^{Q_t(s,b)/\tau}$$

  - ✓Quand τ est **élevée** , l'agent parcourt **uniformément** toutes les actions.
  - ✓Quand τ est **petit**, les **meilleures** actions sont **favorisées**.


Supervised
Learning
Unsupervised
Learning
Machine
Learning
Reinforcement
Learning

Apprentissage supervisé :

- Apprentissage par un enseignant (Superviseur)
- Ensemble d'exemples étiquetés sont disponibles (Situation / Action).
- L'objet de ce type d'apprentissage est que le système généralise ses réponses afin qu'il agisse correctement dans des situations non présentes dans l'ensemble d'apprentissage.

Apprentissage par Renforcement :

- Apprentissage par critique
- Après une séquence d'actions, il vient nous informer « à quel point nous avons bien fait dans **le passé**» (i.e. les retours du critique arrivent tardivement).

- l'Apprentissage par Renforcement est comme **l'Apprentissage non supervisé** concernant l'absence d'un superviseur
  - L'objectif en Apprentissage non supervisé est de trouver une structure cachée dans des collections de données non étiquetées.
  - L'objectif en Apprentissage par Renforcement est de maximiser un signal de récompense.

	Supervisé	Non supervisé	Par renforcement
Optimisation	Non	Non	Oui
Apprendre à base d'expérience	Oui	Oui	Oui
Généralisation	Oui	Oui	Oui
Conséquences retardées	Non	Non	Oui
Exploration	Non	Non	Oui

II. Processus de Décision Markovien (PDM)

❑ Hypothèse de Markov:

Le future ne dépend pas du passé étant donné le présent:

$$p(s_{t+1}|s_t, a_t) = p(s_{t+1}|h_t, a_t)$$

- ❑ Un problème d'apprentissage par renforcement qui vérifie la propriété de Markov est un PDM (**Processus de Décision Markovien**)
- ❑ On parle de ***PDM fini*** lorsque l'ensemble des états et des actions sont finis
- ❑ On parle de ***PDM partiellement observable*** lorsque l'état est partiellement observable.

PDM

- $\boldsymbol{S}$: Ensemble des états.

  Il peut exister un ensemble d'états finaux F . Quand l'agent atteint l'un de ces états, sa tâche est terminée. On parle alors de tâche épisodique, ou à horizon fini.

- **A**: Ensemble des actions.

  On note A(s) : l'ensemble des actions possibles dans l'état s.

- **Modèle de l'environnement (dynamique de l'environnement):** Prédire ce que l'environnement va faire dans l'instant prochain. c'est à dire :

  - Prédire l'état suivant (probabilité de transition d'état) :

$$p(s'|s,a) = \Pr\{\boldsymbol{S_{t+1}} = s'|S_t = s, A_t = a\}$$

  - Prédire la récompense suivante:

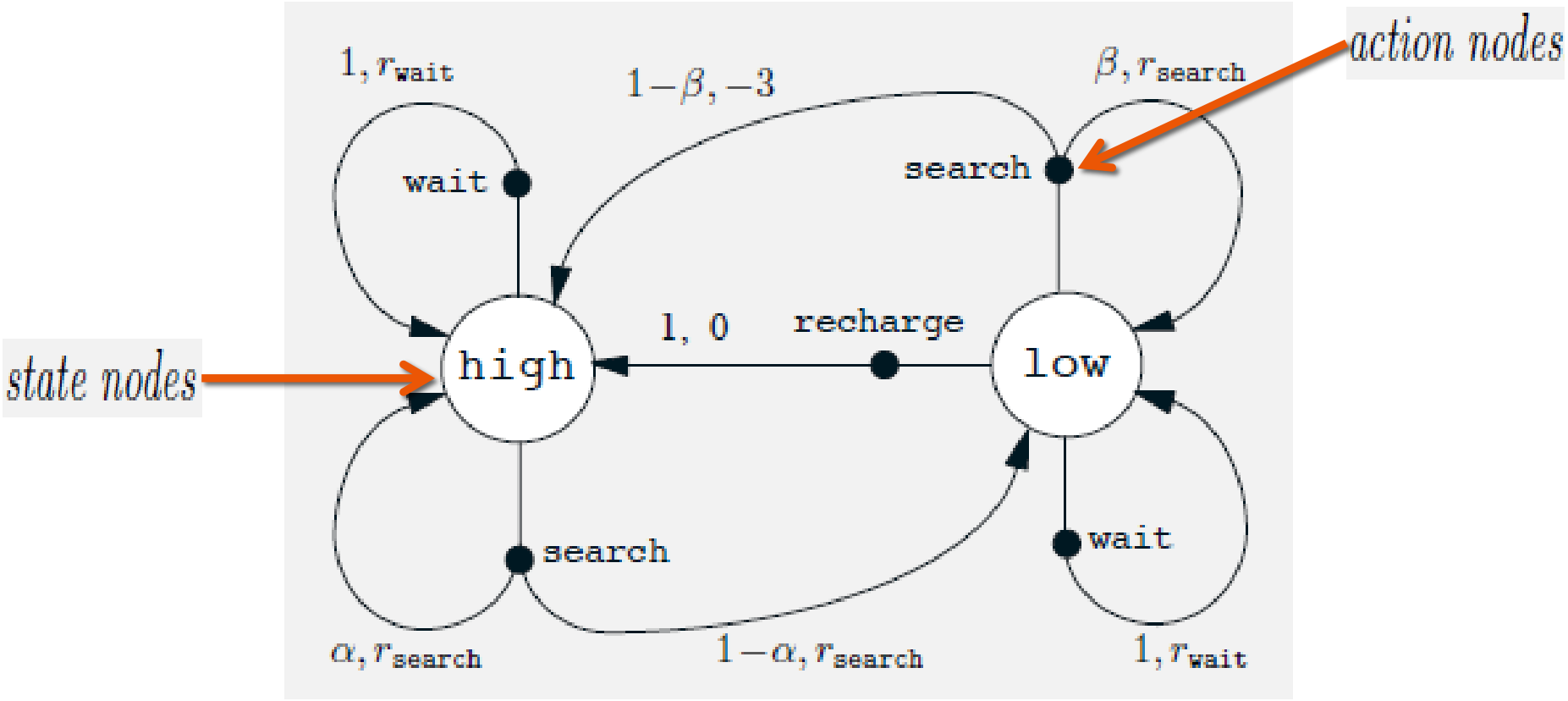
$$r(s,a,s') = E[\boldsymbol{R_{t+1}}|S_t = s, A_t = a, S_{t+1} = s']$$

Exemple : MDP du Robot de recyclage

- Un **robot** collecte des canettes de soda vides dans des bureaux [capteur, bras/pince, Sac embarqué]. Les décisions de haut niveau sur la façon de ***rechercher des boîtes*** sont prises par un **agent** d'apprentissage par renforcement. ***(Robot ≠ agent)***
- La meilleure façon de trouver des canettes est de les **rechercher** activement, mais cela épuise la batterie du robot, alors que l'**attente** ne le fait pas. Chaque fois que le robot recherche, il est possible que sa batterie s'épuise. Dans ce cas, le robot doit s'éteindre et attendre d'être secouru.
- Récupérer une cannette vide : Récompense positive (+1)

  Se décharger complètement (et donc être secouru) : Maximum de pénalité (-3)

Graphe de transition correspondant au MDP du Robot de recyclage



Politique (de décision) de l'agent

Une politique $\boldsymbol{\pi}$ définit le **comportement** de l'agent. Il s'agit d'un mapping de l'ensemble des états à l'ensemble des actions.

- Cas déterministe :

$$\boldsymbol{\pi}: \boldsymbol{S} \longrightarrow \boldsymbol{A}$$

$$\boldsymbol{s} \longrightarrow \boldsymbol{a}$$

- Cas stochastique :

$$\boldsymbol{\pi}: \boldsymbol{S} \times \boldsymbol{A} \longrightarrow [\mathbf{0}, \mathbf{1}]$$

$$(\boldsymbol{s}, \boldsymbol{a}) \longrightarrow \boldsymbol{\pi}(s, a) = \wp[A_t = a | S_t = \boldsymbol{s}]$$

Fonctions d'utilité

La majorité des algorithmes d'AR repose sur l'estimation des **fonctions d'utilité** liées aux **états** ou aux **actions** :

- $\boldsymbol{V_\pi(s)}$: Estimation de la récompense si l'agent se trouve dans l'état s et poursuit la politique π par la suite.

$$V_\pi(s) = E_\pi\ [\boldsymbol{G_t}|S_t = s] = E_\pi[\sum_{k=0}^{\infty} \gamma^k\, R_{t+k+1}\,|S_t = s]$$

- $\boldsymbol{q_\pi(s,a)}$ **:** Estimation de la récompense si l'agent se trouve à s , exécute l'action a et poursuit la politique π par la suite.

$$q_\pi(s,a) = E_\pi\ [G_t|S_t = s, A_t = a] = E_\pi\left[\sum_{k=0}^{\infty} \gamma^k\, R_{t+k+1}\middle| S_t = s, A_t = a\right]$$

Equation de Bellman

Définit la relation entre la valeur d'**un état** et les valeurs de ses états **successeurs** :

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

Exemple:

Soit le MDP du Robot de recyclage. Considérer que cet agent choisit ses actions de manière équiprobable.

Donner $V_\pi(s)$ ....

Fonctions d'utilité optimale

$$V_*(s) = max_\pi V_\pi(s) \quad \forall s \in S$$

$$= max_{a \in A(s)} \sum_{s',r} \boldsymbol{p(s', r|s, a)}[\boldsymbol{r} + \boldsymbol{\gamma V_*(s')}]$$

Exemple:

Considérer le MDP du Robot de recyclage.

Donner $\boldsymbol{V_*(s)}$ ....