Décision dans l'incertain

Cours 9: Décisions séquentielles - PDM

Stéphane Airiau

Université Paris-Dauphine

Processus de décision markovien

Cours 9: Décisions séquentielles - PDM

Stéphane Airiau

Université Paris-Dauphine

Généralisation de expectimax

Avec les arbres de décisions, on a vu un premier type de décision séquentielle dans l'incertain.

On va maintenant voir une généralisation.

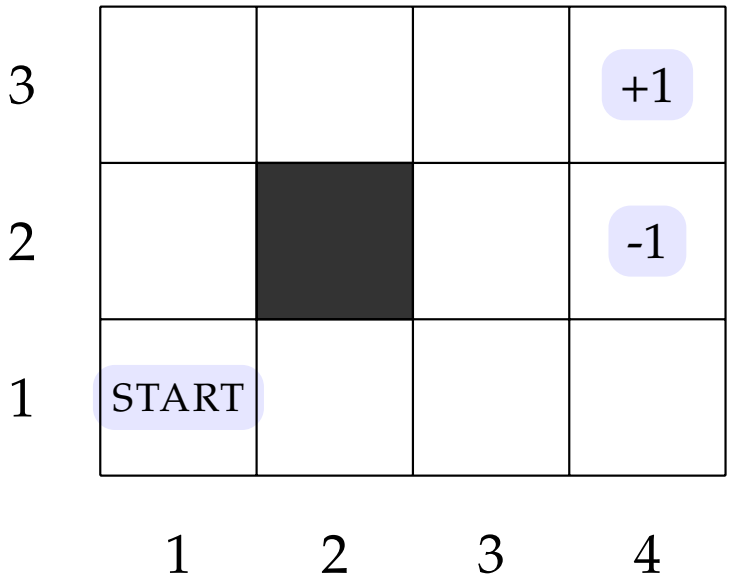
Exemple canonique : un robot se déplace dans une grille.

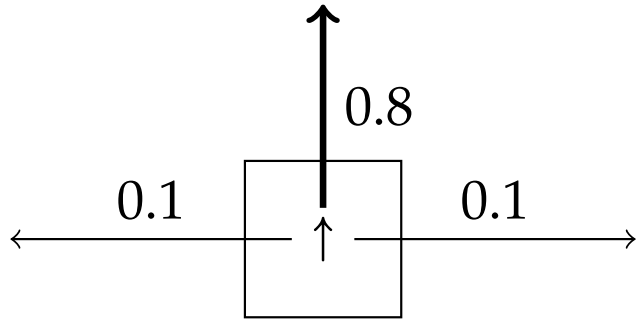
- Les murs bloquent le déplacement de l'agent. Il ne connait pas a priori la position de tous les murs
- Il peut tomber dans des trous, il ne connait pas la position de chaque trou ou s'ils existent
- le robot est un peu plus réaliste : ses actions peuvent échouer (les roues peuvent patiner, ou non)
  - on estime que l'action D mène bien le robot dans la case au D,
  - mais dans 10% des cas, il dévie vers la gauche
  - mais dans 10% des cas, il dévie vers la droite
  - Si $D = Nord$, il a 80% de chance d'aller dans la case au nord, 10% d'aller dans la case à l'est, 10% d'aller dans la case à l'est.

Exemple : gridworld

- l'agent paie une pénalité pour chaque déplacement (il dépense de l'energie)
- certaines cases peuvent contenir une grosse récompense

cf Opportunity et Curiosity sur Mars...





exemple action vers le haut

Définition (Processus décisionnel de Markov)

Un *Processus décisionnel de Markov* est un tuple $\langle S, A, T, R, \gamma \rangle$ où

- S est un ensemble fini d'états
- A est un ensemble fini d'actions
- T est une matrice de transition
  $T^a_{ss'} = \mathbb{P}\left[S_{t+1} = s' \mid S_t = s, A_t = a\right]$ probabilité d'arriver dans l'état s' à l'instant +1 quand on a pris l'action a dans l'état s à l'instant t
- R est le vecteur de récompenses
  $R^a_s = \mathbb{E}\left[R_{t+1} \mid S_t = s, A_t = a\right]$ valeur moyenne obtenue après avoir pris l'action a dans l'état s
- un ensemble d'état initial
- parfois un ensemble d'états terminaux

Hypothèse de Markov

Si on exécute l'action a dans l'état s :

- On obtient une récompense r
- On arrive dans un état s'

En principe, r et s' peuvent dépendre de tout l'historique !

Définition (Etat de Markov)

Un état S_t est dit de **Markov** ssi

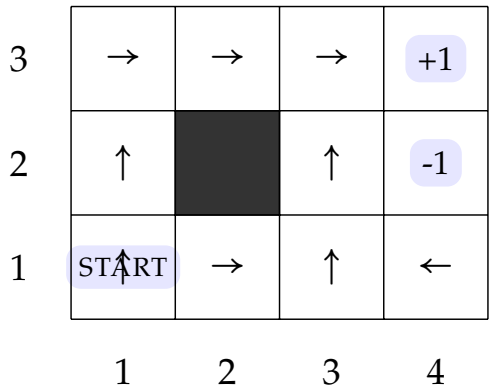
$$\mathbb{P}[S_{t+1}|S_t] = \mathbb{P}[S_{t+1}|S_1, \ldots, S_t]$$

- "Etant donné le présent, le futur est indépendant du passé"
- Une fois que l'état est connu, on peut effacer son historique !
- *ex :* aux échecs, l'état du jeu ne dépend pas de l'historique des coups !

Les composants d'un agent : politique

C'est ce qui gouverne le comportement de l'agent

- **politique déterministe** : La fonction associe à chaque état **une action** $\pi : S \mapsto A$



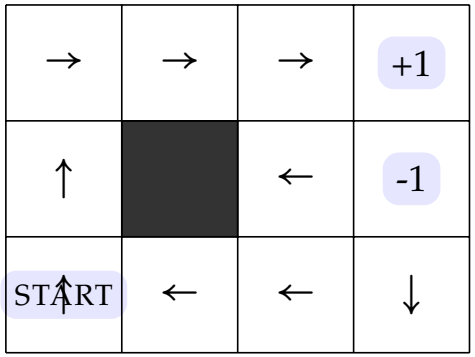
politique optimale pour un pénalité de 0.03 par déplacement

- **politique stochastique** : une distribution de probabilité sur les actions possibles

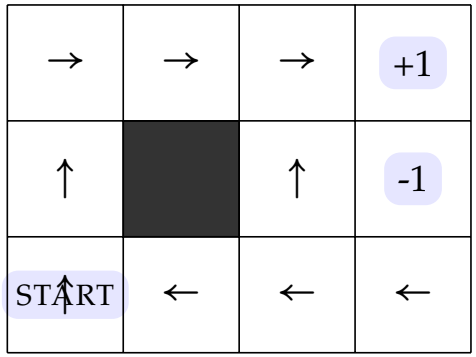
$$\pi : S \mapsto \Delta(A)$$

où $\Delta(N)$ désigne une distribution de probabilité sur l'ensemble (fini) N.
$p \in \Delta(N)$ ssi $\forall i \in N, p(i) \in [0,1]$ et $\sum_{i \in N} p(i) = 1$

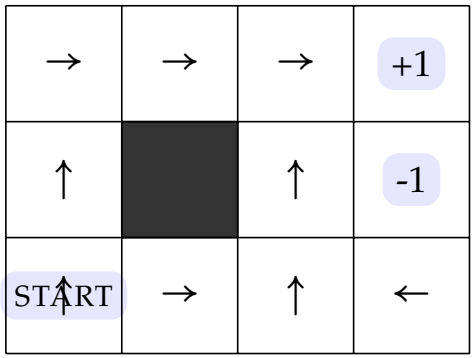
Politiques optimale



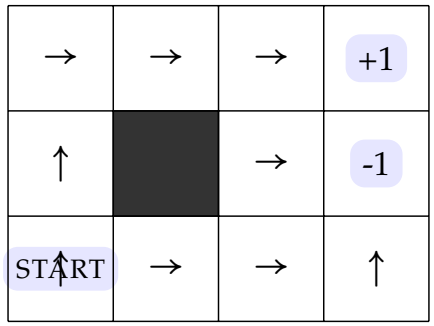
pénalité de 0.01 par déplacement
aucun risque : on fait le tour!



pénalité de 0.02 par déplacement
petit risque



pénalité de 0.04 par déplacement
on prend le chemin le plus court

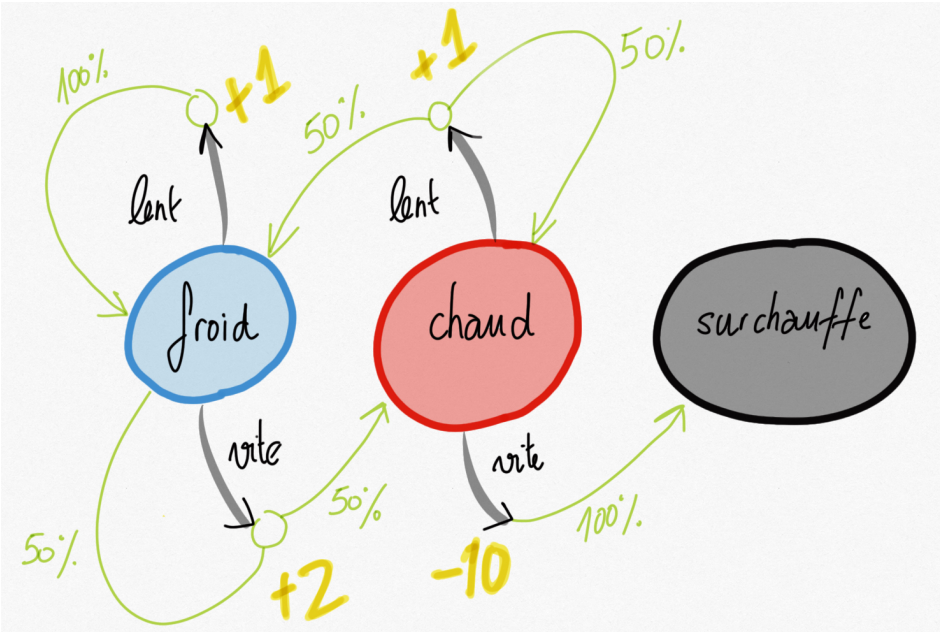


pénalité de 2 par déplacement
on prend des risques pour terminer au plus vite

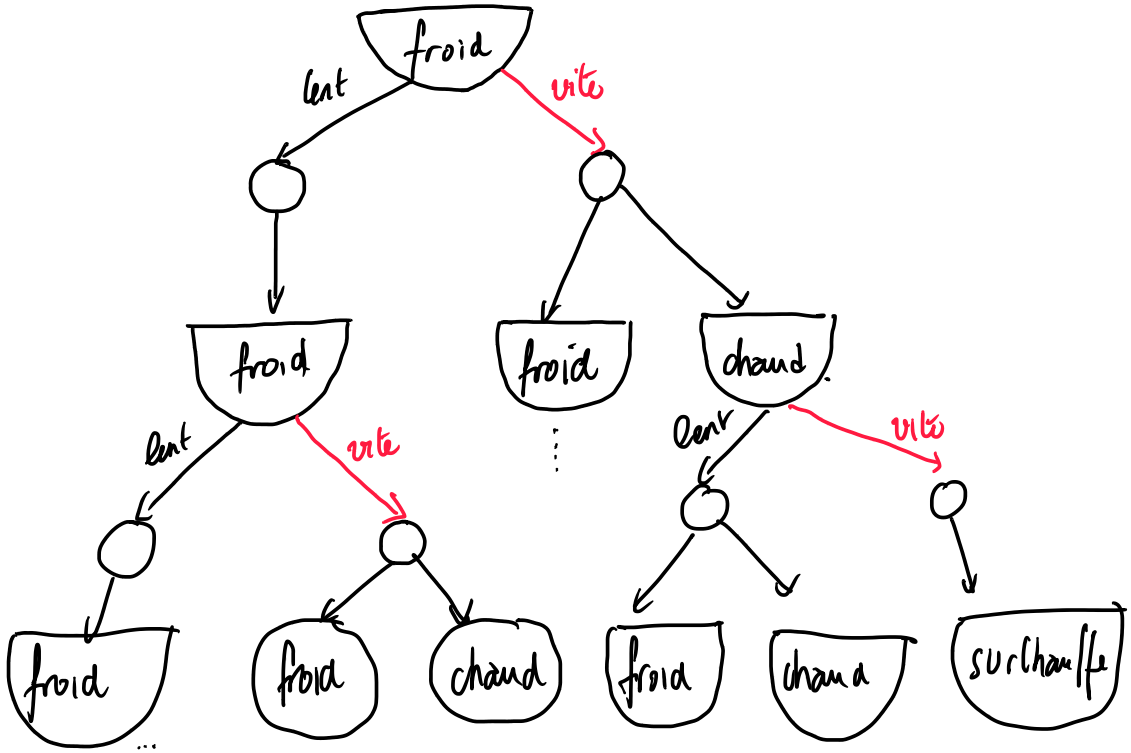
Exemple de la voiture de course

un robot voiture veut aller vite et loin !

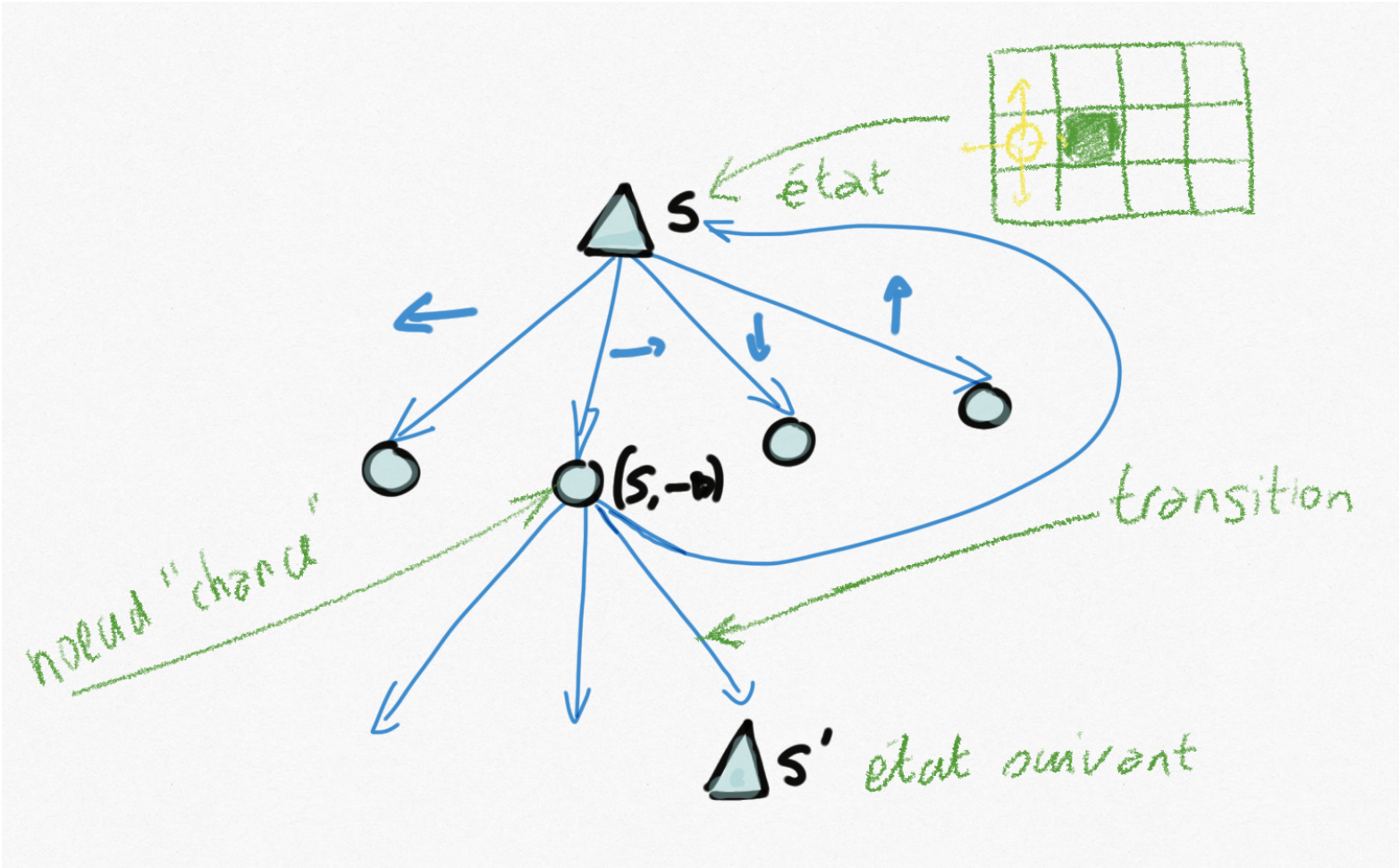
- trois état du moteur : normal, chaud, surchauffe
- deux actions : lent, vite
- en allant plus vite, on double la récompense
- +1 pour les action lent, +2 pour action vite, -10 pour atteindre un état surchauffe.
  la description ne suit pas exactement le modèle formel



Exemple de la voiture de course & expectimax



Exemple de la voiture de course & MDP



Avec les MDP, on ne va pas raisonner sur un arbre, mais sur un graphe.

Quel est notre but

On va obtenir une séquence de valeurs d'utilité. Que préfère-t-on ?

- maintenant, plus tard ? (ex $\langle 0,0,10 \rangle$ ou $\langle 8,2,0 \rangle$)
  vous préférez 10 dans 3 jours, ou 8 aujourd'hui, 2 demain, et 0 dans 3 jours ?

Quel est notre but

On va obtenir une séquence de valeurs d'utilité. Que préfère-t-on ?

- maintenant, plus tard ? (ex $\langle 0,0,10 \rangle$ ou $\langle 8,2,0 \rangle$)
  vous préférez 10 dans 3 jours, ou 8 aujourd'hui, 2 demain, et 0 dans 3 jours ?
- généralement, on préfère avoir plus d'utilité en tout, mais une distribution plus régulière peut être appréciable !

Quel est notre but

On va obtenir une séquence de valeurs d'utilité. Que préfère-t-on ?

- maintenant, plus tard ? (ex $\langle 0,0,10 \rangle$ ou $\langle 8,2,0 \rangle$)
  vous préférez 10 dans 3 jours, ou 8 aujourd'hui, 2 demain, et 0 dans 3 jours ?
- généralement, on préfère avoir plus d'utilité en tout, mais une distribution plus régulière peut être appréciable !
- Dans certains problèmes où il y a une **fin**, on pourra chercher à maximiser la somme des utilités

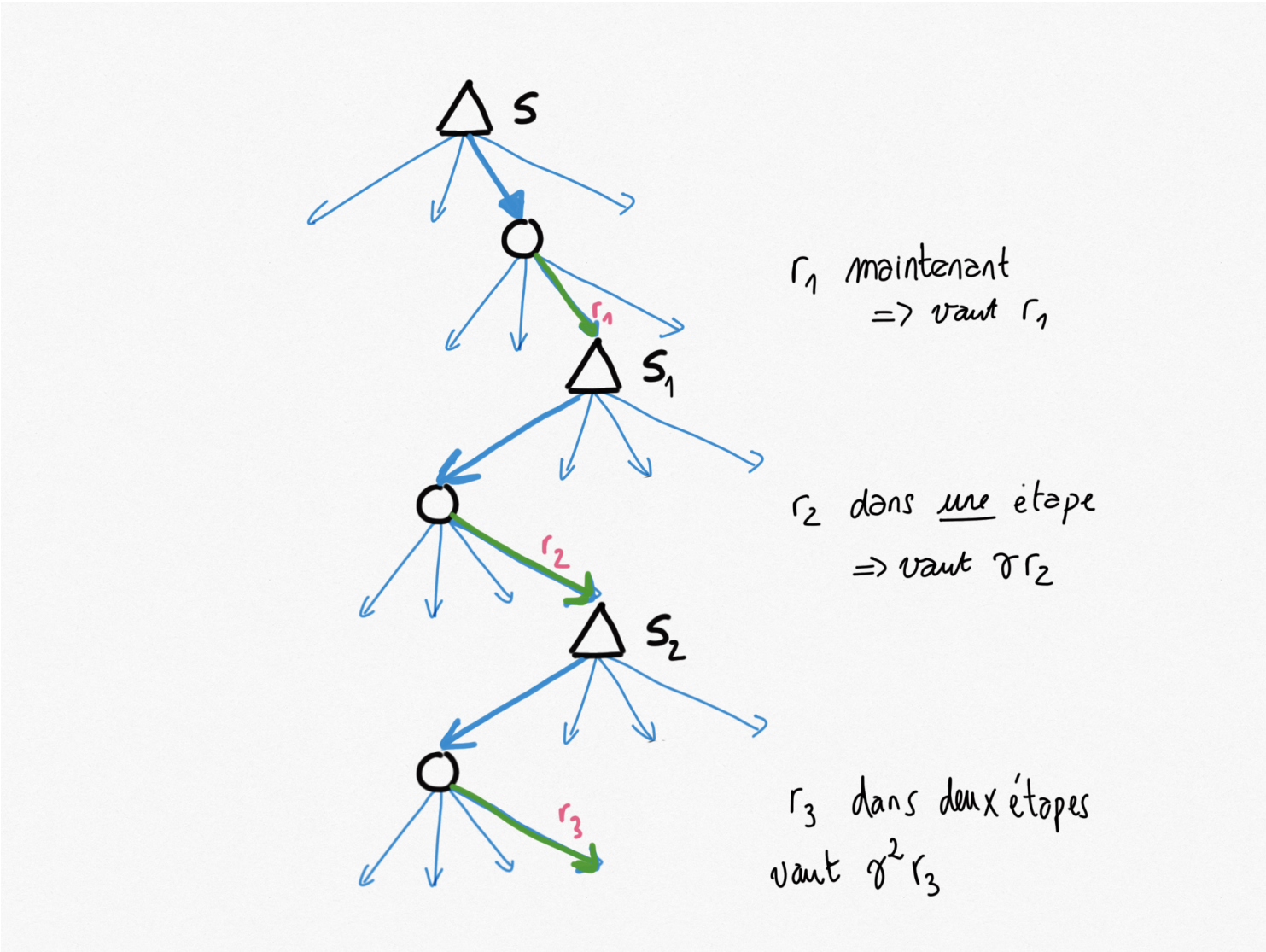
Quel est notre but

Dans tous les autres cas, on peut utiliser un taux d'escompte γ

- avoir 1 aujourd'hui vaut 1 aujourd'hui !
- avoir 0 aujourd'hui et 1 demain vaut aujourd'hui γ aujourd'hui
- avoir 0 aujourd'hui, 0 demain et 1 dans deux jours vaut γ^2 aujourd'hui

- $\gamma = 0$ l'agent est "myope" : il n'est intéressé que par la récompense immédiate
- $0 < \gamma < 1$ l'agent cherche un équilibre entre la récompense immédiate et celle qu'il obtiendra dans le futur

Récompense avec escompte



Si on raisonne sur ce chemin, la récompense sera $r_1 + \gamma r_2 + \gamma^2 r_3$.

Quel est notre but : version formelle

- pour des tâches épisodiques
  - il y a des états terminaux et initiaux
  - on repart dans un état initial une fois qu'on atteint un état terminal

  ⇨ maximise le cumul des récompenses sur *un épisode* (ici de longueur T)

$$G_T = r_1 + r_2 + \cdots + r_T$$

- pour des tâches en continue

  ⇨ maximise une récompense "escomptée" $G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$

  γ est le taux d'escompte
  - $\gamma = 0$ l'agent est "myope" : il n'est intéressé que par la récompense immédiate
  - $0 < \gamma < 1$ quand $\{r_t, t \in \mathbb{N}\}$ est bornée, alors R_T est bien définie.
    ⇨ l'agent cherche un équilibre entre la récompense immédiate et celle qu'il obtiendra dans le futur

- maximiser une récompense "en moyenne" $R_t = \frac{1}{t} \cdot \sum_{k=0}^{\infty} r_{t+k+1}$

Focus pour le cours

- problèmes itératifs en continue.
- objectif : maximiser la somme "avec dévaluation" $R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$
  - Pour éviter une récompense infinie si on tombe dans des cycles
  - Le futur reste uncertain ! Bon compromis entre court et long terme
  - Tendance naturelle vers le court terme
  - Mathématiquement, c'est quand même pratique !
- la fonction de transition est stochastique
- la fonction de récompense est connue

Comment trouver la meilleure politique ?

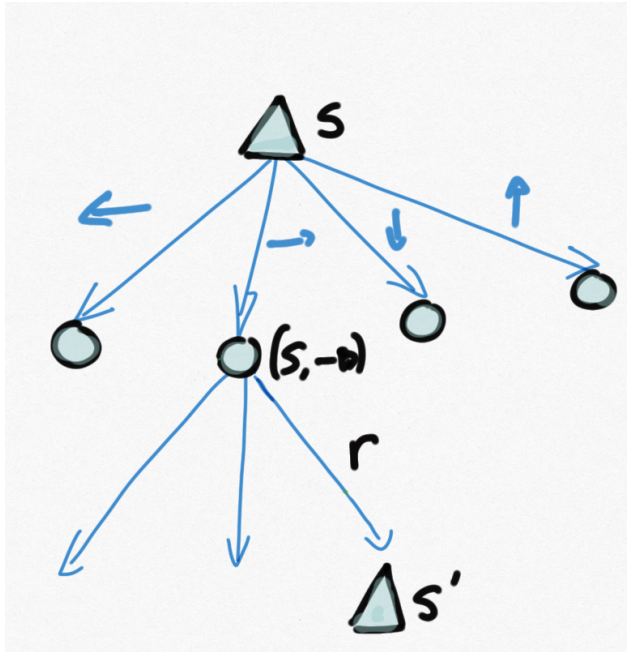
Evaluation

On va s'aider de deux quantités

$v^\star(s)$ quelle est la valeur de me trouver dans l'état s puis de continuer avec la politique optimale

$q^\star(s,a)$ quelle est la valeur de prendre l'action a dans l'état s puis de continuer avec la politique optimale

$\pi^\star(s)$ politique optimale pour l'état s (i.e. quelle est la meilleure action).



Valeur optimale d'un état $v^\star(s)$

- on calcule la valeur espérée en supposant qu'on suive la politique optimale
- on prend la moyenne pondérée des récompenses escomptée

⇨ comme dans expectimax !

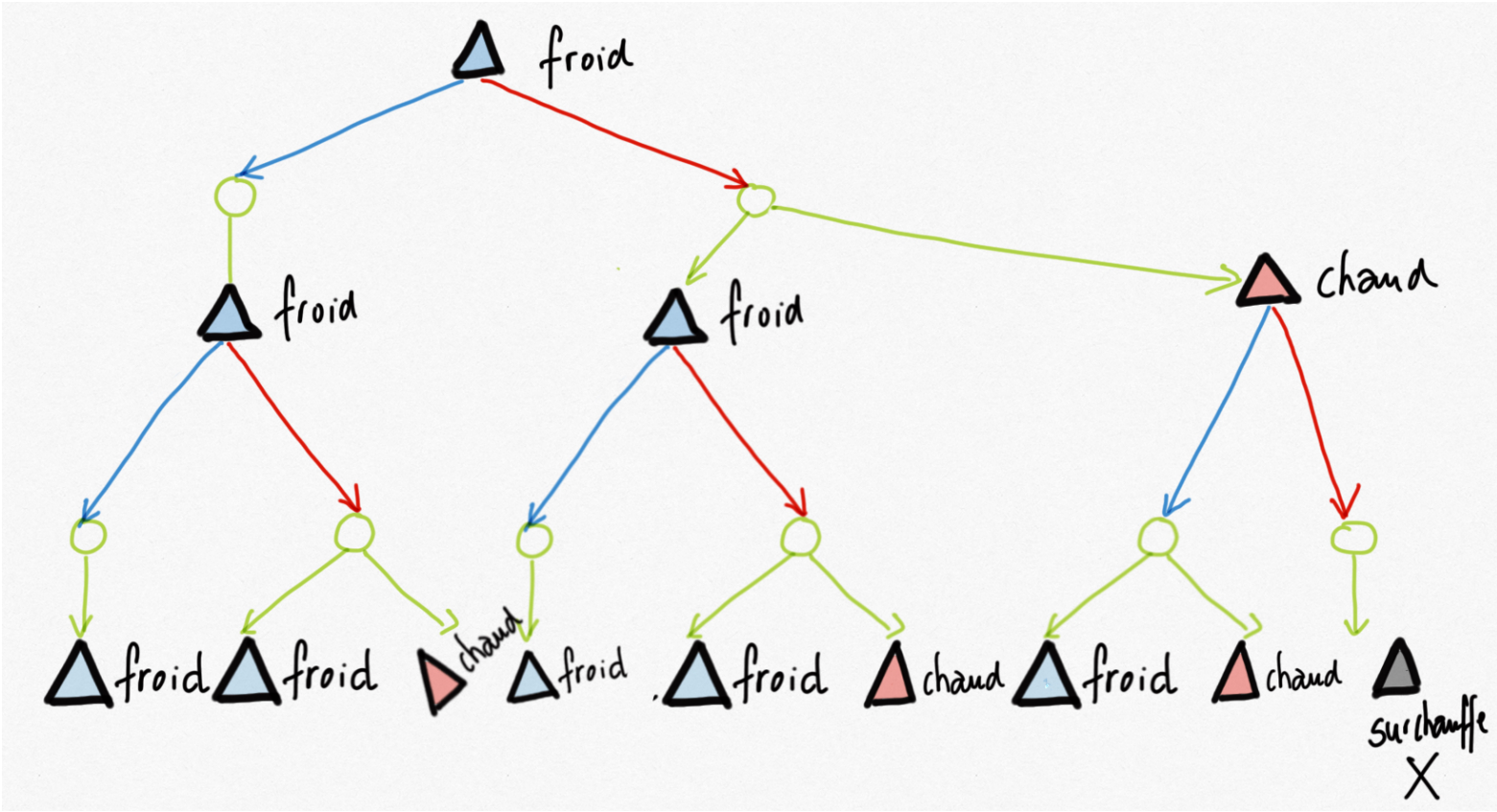
$$v^\star(s) = \max_{a \in A} q^\star(s,a)$$

$$q^\star(s,a) = R_s^a + \gamma \sum_{s' \in S} T_{ss'}^a v^\star(s')$$

donc

$$v^\star(s) = \max_{a \in A} \left[R_s^a + \gamma \sum_{s' \in S} T_{ss'}^a v^\star(s') \right]$$

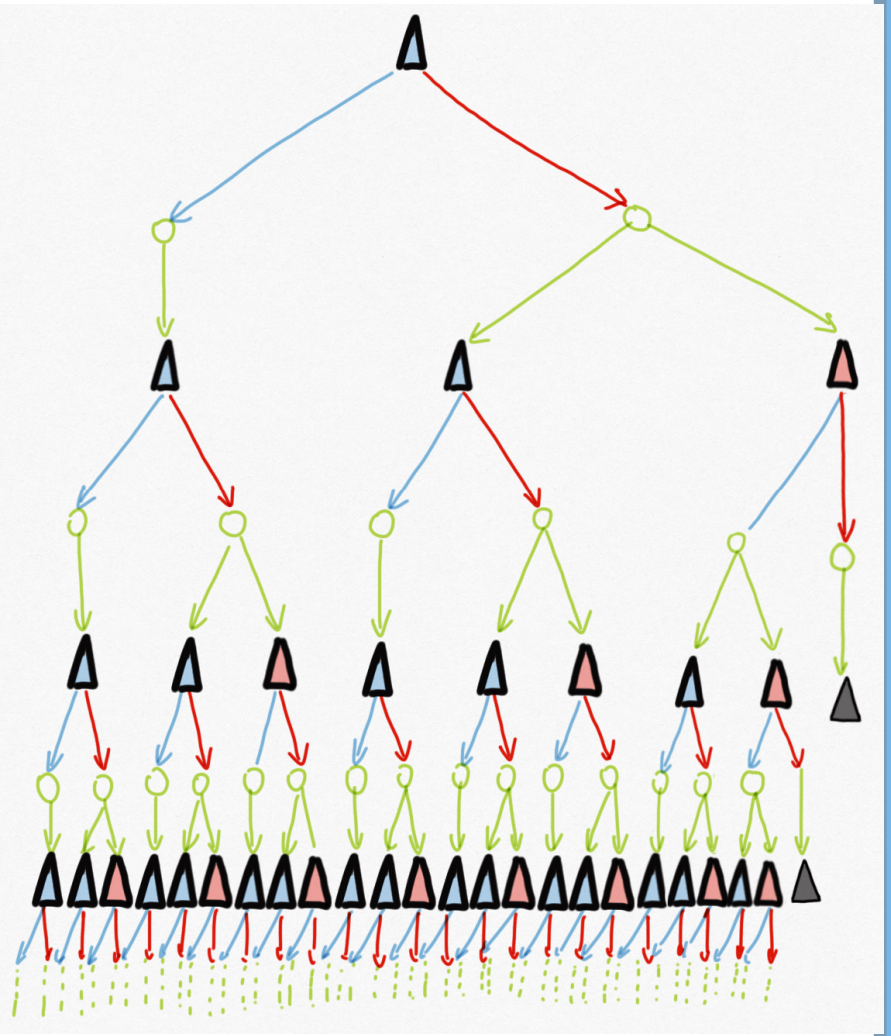
Arbre de recherche



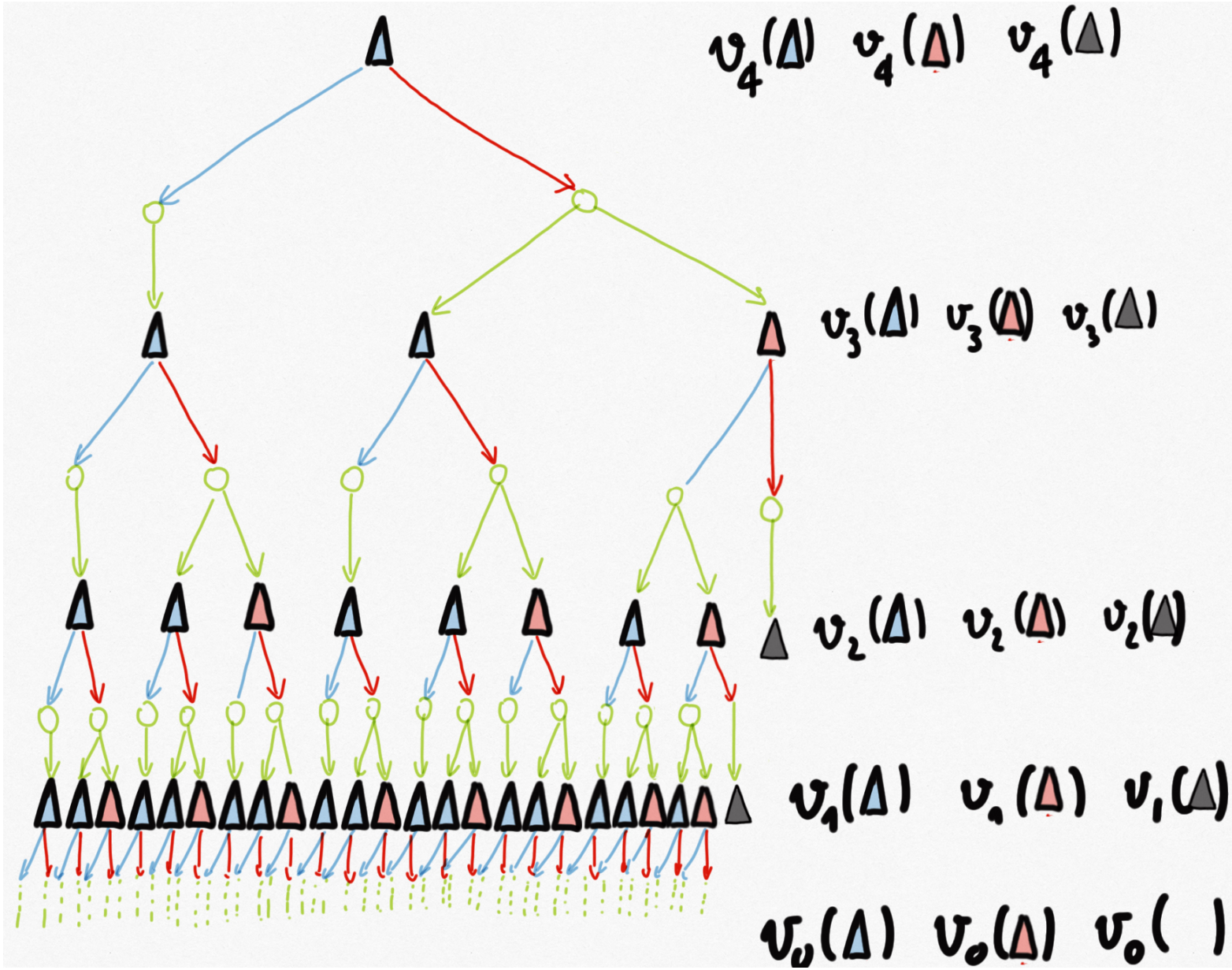
Arbre de recherche

Problèmes

- les noeuds se répètent dans l'arbre
- ⇨ on ne va pas répéter le calcul
- la profondeur de l'arbre est infinie
- ⇨ à cause de γ, plus on va profondemment, plus les quantités deviennent négligeable !
- ⇨ depth-limited



Idée d'itération sur les valeurs



Value Iteration

```
1  for each s ∈ S
2      V(s) ← 0
3
4  repeat
5
6      for each s ∈ S
7
8          V(s) ← max_{a∈A} [ R^a_s + γ Σ_{s'∈S} T^a_{ss'} V(s') ]      /* mise à jour */
9
10 until convergence
```

Value Iteration

```
1  for each s ∈ S
2      V(s) ← 0
3
4  repeat
5      Δ ← 0                                   /* mesure le plus grand changement */
6      for each s ∈ S
7          v ← V(s)          /* sauvegarde l'ancienne valeur pour mesurer le changement */
8          V(s) ← max_{a∈A} [ R_s^a + γ Σ_{s'∈S} T_ss'^a V(s') ]          /* mise à jour */
9          Δ ← max(Δ, |v − V(s)|)          /* mise à jour du plus grand changement*/
10 until Δ < ε                                          /* test convergence */
```

- On n'a pas de politique explicite
- On a un théorème de convergence

Fonctions de valeurs

On peut définir deux fonctions de valeurs pour une politique fixée π.

Définition (fonction de valeurs pour les états)

La *fonction de valeurs pour les états* $v_\pi(s)$ d'un PDM est la valeur espérée de gains en partant dans l'état s et en poursuivant la politique π.

Définition (fonction de valeurs pour les paires (état, action))

La *fonction de valeurs pour les paires état-actions* $q_\pi(s,a)$ d'un PDM est la valeur espérée de gains en partant dans l'état s, en effectuant l'action a *puis* et en poursuivant la politique π.

Equation de Bellman : pour les états

Dans l'état s

on tire notre action avec la politique $\pi(s)$

pour chaque action, on choisit l'action a avec la probabilité $\pi(a|s)$,

on va effectuer l'action a puis continuer avec π dans l'état suivant

➪ on peut utiliser q_π !

$$\begin{aligned} v_\pi(s) &= \mathbb{E}_\pi\left[r_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s\right] \\ &= \sum_{a \in A} \pi(a|s) q_\pi(s,a) \end{aligned}$$

Equation de Bellman : pour les actions

Similairement pour la fonction de valeurs pour les actions

- le modèle de récompense nous donne la récompense pour avoir effectué l'action a dans l'état s.
- le modèle de transition nous donne l'état suivant s'

⇨ dans ce nouvel état s', on peut utiliser v_π !

$$\begin{aligned} q_\pi(s,a) &= \mathbb{E}_\pi\left[r_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) \mid S_t = s, A_t = a\right] \\ &= R_s^a + \gamma \sum_{s' \in S} T_{ss'}^a v_\pi(s') \end{aligned}$$

Equation de Bellman

On a établi :

$$v_\pi(s) = \sum_{a \in A} \pi(a|s) q_\pi(s,a)$$

$$q_\pi(s,a) = R_s^a + \gamma \sum_{s' \in S} T_{ss'}^a v_\pi(s')$$

Ensemble on obtient :

$$v_\pi(s) = \sum_{a \in A} \pi(a|s) \left(R_s^a + \gamma \sum_{s' \in S} T_{ss'}^a v_\pi(s') \right)$$

$$q_\pi(s,a) = R_s^a + \gamma \sum_{s' \in S} T_{ss'}^a \sum_{a' \in A} \pi(a'|s') q_\pi(s',a')$$

Equation de Bellman

$$
\begin{aligned}
v_\pi(s) &= \sum_{a \in A} \pi(a|s) \left(R_s^a + \gamma \sum_{s' \in S} T_{ss'}^a v_\pi(s') \right) \\
&= \sum_{a \in A} \pi(a|s) R_s^a + \gamma \sum_{a \in A} \sum_{s' \in S} \pi(a|s) T_{ss'}^a v_\pi(s') \\
&= \sum_{a \in A} \pi(a|s) R_s^a + \gamma \sum_{s' \in S} \sum_{a \in A} \pi(a|s) T_{ss'}^a v_\pi(s') \\
&= R_s^\pi + \gamma \sum_{s' \in S} T_{ss'}^\pi v_\pi(s')
\end{aligned}
$$

On peut donc écrire l'expression vectorielle

$$v_\pi = R^\pi + \gamma T^\pi v_\pi$$

Evaluation d'une politique fixée π

On peut utiliser l'équation de Bellman comme une règle de mise à jour :

$$v_{k+1}(s) \leftarrow \sum_{a \in A} \pi(a \mid s) \left(R_s^a + \gamma \sum_{s' \in S} T_{ss'}^a v_k(s') \right)$$

Theorème

La séquence $\{v_k\}_{k \in \mathbb{N}}$ converge vers v^π.

Convergence plus rapide

Pour réaliser l'algorithme :

- avoir deux vecteur v_{old} et v_{new}
- calculer completement v_{new} à partir de v_{old}

⇨ "full back up"

On peut aussi n'utiliser qu'*un seul* vecteur

- on remplace directement l'ancienne entrée par la nouvelle
- le vecteur v contient à la fois des nouvelles et des anciennes valeurs

⇨ on utilise les nouvelles valeurs au plus vite
convergence toujours garantie et plus rapide
l'ordre de mise à jour joue un rôle sur la vitesse de convergence.

Critère d'arrêt de l'algorithme

- garantie de convergence à la limite
- en pratique, on peut arrêter avant
par exemple : $\max_{s \in S} |v_{k+1}(s) - v_k(s)| < \epsilon$ pour une valeur de ϵ donnée.

Améliorer une politique

On peut essayer d'améliorer la politique en se comportant de manière "gloutonne"

Une fois v_π évaluée : on peut calculer $q^\pi(s,a) = R_s^a + \gamma \sum_{s' \in S} T_{ss'}^a v_\pi(s')$

- si $q^\pi(s,a) > v_\pi(s)$: on a trouvé une amélioration ![a]
- ⇨ on peut regarder tous les états $s \in S$ et mettre à jour la politique $\pi'(s) = arg\max_{a \in A} q_\pi(s,a)$

Si aucune amélioration n'est trouvée, on a donc $v_\pi = v_{\pi'}$
⇨ $v_{\pi'} = \max_{a \in A} R_s^a + \gamma \sum_{s' \in S} T_{ss'}^a v_{\pi'}(s')$
On reconnait là l'équation de Bellman pour la fonction de valeurs *optimale*
On a donc trouvé $v^\star = v_{\pi'}$!

a. il faut une petite démonstration sur ce point

Itération de politique

L'idée est donc d'alterner

1- l'évaluation d'une politique

2- l'amélioration de la politique

*jusqu'à ce qu'*on converge vers une politique qui sera la politique optimale.

Pour les politiques deterministes, il y a un nombre fini de politiques, on va converger en un nombre fini d'itérations.

Variantes : quand arrêter l'évaluation ?

- convergence à un ϵ près
- après k itérations (k a une petite valeur)
- pourquoi pas après chaque itération ?