

Agents Intelligents  
Licence 3 Intelligence Artificielle — Portail Sciences et  
Technologies  
Travaux pratiques N° 2 : Agents Rationnels

Andrea G. B. Tettamanzi  
Université Côte d’Azur  
andrea.tettamanzi@univ-cotedazur.fr

Année universitaire 2021/2022

**Résumé**

Dans cette séance, qui est une continuation de la précédente, nous allons faire d’autres expériences avec différents types d’agents, toujours dans le “monde de l’aspirateur” vu en cours.

## 1 Introduction

Nous allons considérer le “monde de l’aspirateur” vu en cours. Dans chaque unité de temps, l’agent aspirateur peut soit bouger d’une case, soit nettoyer la case dans laquelle il se trouve (si elle est sale). L’agent est équipé avec deux capteurs : un capteur de position, qui lui renvoie ses coordonnées, et un capteur de salissure.

La note de performance de l’agent est calculée en comptant un point pour chaque case nettoyée à chaque unité de temps. On suppose une “durée de vie” de l’agent de 1000 unités de temps.

La topographie de l’environnement est connue *a priori* par l’agent, mais la distribution de la salissure et la position initiale de l’agent ne le sont pas.

Nous allons réutiliser le simulateur de l’environnement réalisé dans la séance précédente.

## 2 Consignes

1. On va maintenant créer un nouvel agent doté d’un modèle, c’est-à-dire d’une représentation interne de ce que l’agent sait (ou croit savoir) de l’état de l’environnement et d’une description de la façon dont l’état suivant dépend de l’état courant et de l’action effectuée par l’agent :
  - (a) Décidez comment représenter les croyances de l’agent sur l’état de l’environnement. Ce qu’il faut représenter est la position à laquelle l’agent croit être et si chaque case est sale ou propre. Supposons pour l’instant que l’agent a une connaissance innée de la topographie de l’environnement.
  - (b) Codez la fonction (ou méthode) ACTUALISER-ÉTAT, qui, étant données la dernière action exécutée et la perception courante, actualise la représentation interne de l’état de l’environnement ; le code de cette fonction pourra être considérée le “modèle” qui décrit la façon dont l’état suivant dépend de l’état courant et de l’action effectuée.

- (c) Modifiez votre agent pour que son comportement soit décrit par des règles condition-action : comment peut-on représenter ces règles dans notre cas ?
  - (d) Codez la fonction (ou méthode) TROUVER-RÈGLE, qui, étant donné la représentation interne de l'état, renvoie la règle à utiliser.
  - (e) Écrivez les règles qui déterminent le comportement de l'agent.
2. Modifiez maintenant le simulateur de l'environnement pour le rendre non-déterministe :
    - (a) à chaque fois que l'action *aspirer* est effectuée, il y a une probabilité de 20% que cette action échoue (la propreté de la case ne change pas) ;
    - (b) à chaque fois que l'action de bouger est effectuée, il y a une probabilité de 10% que les roues glissent et que la position de l'aspirateur reste inchangée ;
    - (c) à chaque instant, chaque case propre a une probabilité de 5% de devenir sale.
  3. Modifiez les règles qui déterminent le comportement de l'agent pour gérer de manière rationnelle ce nouvelle version de l'environnement.
  4. Faisons tomber l'hypothèse que l'agent a une connaissance innée de la topographie de l'environnement, de sorte à ce qu'il doive la découvrir. Modifiez la représentation interne de l'agent e sa fonction (ou méthode) ACTUALISER-ÉTAT pour gérer ce cas.

### 3 Rendu

Rédigez un document avec vos réponses aux questions et vos éventuelles remarques et envoyez par courriel à l'enseignant votre code Python et vos observations dans un archive zippé.