



UNIVERSITÉ LIBRE DE BRUXELLES

INFO-F311
INTELLIGENCE ARTIFICIELLE

Synthèse IA

Étudiants :

Rayan CONTULIANO BRAVO

Enseignants :

T. LENAERTS

2 octobre 2023

A large, faint, light blue watermark of the ULB seal is visible in the background. It features a circular border with the text 'SCIENTIA VINCERE' and 'UNIVERSITAS BRUXELLENSIS'. In the center is a sunburst emblem, and below it are two crossed torches.

Contents

1	Introduction	2
2	Recherche	5
2.1	Problème de recherches	5

1 Introduction

Définition 1.1. Qu'est-ce que l'ia

L'intelligence artificielle est une branche de l'informatique qui crée des systèmes qui pensent de manière **rationnelle**

Définition 1.2. Décisions rationnelles

Penser de manière rationnelle signifie qu'on va se concentrer sur le **choix de décisions** qui *maximisent la probabilité* d'atteindre un objectif donné. On va faire agir les systèmes de manière **optimale**

Remarques:

1. Être rationnel signifie donc **maximiser** l'utilité attendue.
2. On définit un objectif par son **utilité**.

Définition 1.3. Agent

Un agent est un système qui perçoit son environnement par des **capteurs** et agit sur celui-ci par des **effecteurs**.

Définition 1.4. Agent rationnel

Un agent rationnel est un agent qui agit de manière à maximiser son utilité attendue.

Remarque: Les **capteurs**, **effecteurs** et l'**environnement** permettent à l'agent de percevoir et d'agir sur le monde de manière **rationnelle**. L'**agent** est le système qui prend les décisions.

Définition 1.5. Fonction agent

La fonction agent est une fonction qui prend en entrée une séquence de perceptions et retourne une action. $f : \mathcal{P}^* \rightarrow \mathcal{A}$

Exemple:

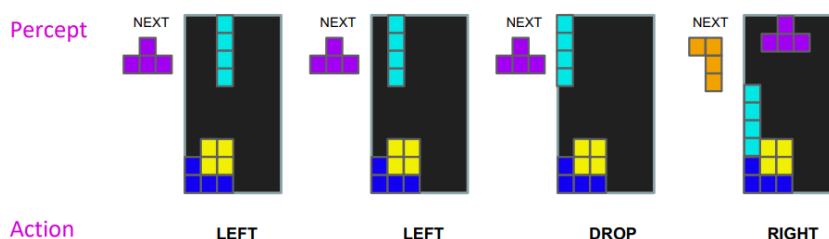


FIGURE 1.1 – Représentation fonction agent dans le jeu Tetris

Définition 1.6. Programme Agent

Un **programme agent** l est exécuté sur une **machine** M afin d'implémenter la fonction agent f .

Remarque: Les machines dans le monde réel sont **imparfaites** et **limitées** en temps et en mémoire.

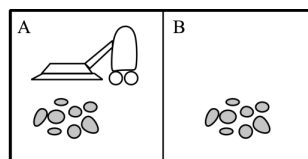


FIGURE 1.2 – Etat de l’environnement de l’aspirateur

Exemple: Nous pouvons représenter un aspirateur comme un agent qui perçoit son environnement par des capteurs et agit sur celui-ci par des effecteurs.

- **Perception** : capteurs qui détectent la saleté et sa localisation dans l’espace
- **Action** : effecteurs qui déplacent l’aspirateur dans l’espace et aspire ou non

En imaginant la situation en figure 1.2, nous pouvons définir la fonction agent de l’aspirateur comme suit :

TABLE 1 – Fonction agent de l’aspirateur

Sequence de perception	Action
[A, Clean]	Right
[A, Dirty]	Suck
[B, Clean]	Left
[B, Dirty]	Suck
[A, Clean], [B, Clean]	Left
[A, Clean], [B, Dirty]	Suck
etc...	etc...

Pour que notre agent soit bien rationnel, il nous faut une manière de **mesurer** la **performance** de celui-ci. Pour cela, nous allons définir une **fonction de performance** qui va mesurer la qualité des actions de l’agent.

Exemple: On peut lui faire gagner des points ou bien lui en retirer en fonction d’une action

De cette manière, l’agent va savoir quelles actions lui permettent de **maximiser** son utilité attendue.

Afin de bien déterminer un environnement, les particularité de notre agents, il nous faut **avant toute chose** définir **PEAS**

Définition 1.7. PEAS

- **Performance** : mesure de la qualité des actions de l’agent
- **Environnement** : type d’environnement dans lequel l’agent va évoluer
- **Actuateurs** : les effecteurs de l’agent
- **Sensors** : les capteurs de l’agent

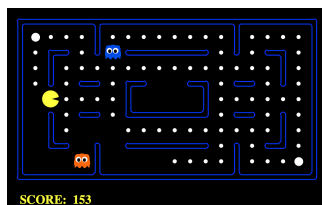


FIGURE 1.3 – Environnement Pacman

Exemple: Pour l'environnement Pacman de la figure 1.3, nous pouvons définir PEAS comme suit :

- **Performance** : -1/pas, +10/nourriture, +500/partie gagnées, -500/mort, +200/tuer un fantôme effrayé
- **Environnement** : labyrinthe **dynamique** de pacman
- **Actuateurs** : Haut, Bas, Gauche, Droite
- **Capteurs** : L'état entier visible

Définition 1.8. Types d'environnement

Il y a plusieurs type d'environnement :

- **Mono-agent** : un seul agent
- **Multi-agent** : plusieurs agents qui maximisent leur **propre** tâche (coop ou concurrentiel)
- **Déterministe** : l'état de l'env est déterminé **seulement** par les actions de l'agent
- **Stochastique** : l'environnement est non déterministe
- **Épisodique** : les actions de l'agent n'affectent pas les actions futures
- **Séquentiel** : les actions de l'agent affectent les actions futures
- **Dynamique** : l'environnement peut changer pendant que l'agent réfléchit
- **Statique** : l'environnement ne change pas pendant que l'agent réfléchit
- **Complètement observable** : les capteurs de l'agent perçoivent l'état complet de l'environnement
- **Partiellement observable** : les capteurs de l'agent perçoivent une partie de l'état de l'environnement
- **Discret** : un nombre fini d'états
- **Continu** : un nombre infini d'états
- **Connu** : l'agent connaît les lois de l'environnement

Il existe plusieurs types d'agents qui répondent à des environnements plus complexes :

- **Agent réflexe simple** : l'agent choisit son action en fonction de la **dernière** perception
- **Agent réflexe basé sur un modèle** : l'agent choisit son action en fonction de la **dernière** perception et d'un **état interne**(dépend de l'**historique** des perceptions)
- **Agent fondés sur des buts** : l'agent choisit son action en fonction de la dernière perception ainsi que des infos relatives à l'objectif
- **Agent fondés sur l'utilité** : l'agent choisit son action en fonction de sa satisfaction par rapport à l'état résultant

2 Recherche

Définition 2.1. Agent de Plannification

Un agent de planification font des **hypothèses** sur les conséquences des actions entreprises et utilise un **modèle** de l'environnement pour trouver un plan qui atteint son objectif.

Note:-

Un plan est une séquence d'actions qui mène à l'objectif.

2.1 Problème de recherches

Définition 2.2. Problème de Recherche

Un problème de recherche est défini par :

- **Ensemble d'État** S : Une situation dans lequel l'environnement peut être agencé
- **État initial** s_o : l'état dans lequel le problème commence
- **Actions** $A(s)$: les actions possibles dans l'état s
- **Modèle de Transition** $Result(s, a)$: la fonction qui définit les conséquences des actions
- **Solution** : Une séquence d'actions qui mène de l'état initial à l'état final
- **État final** : l'état que l'on veut atteindre

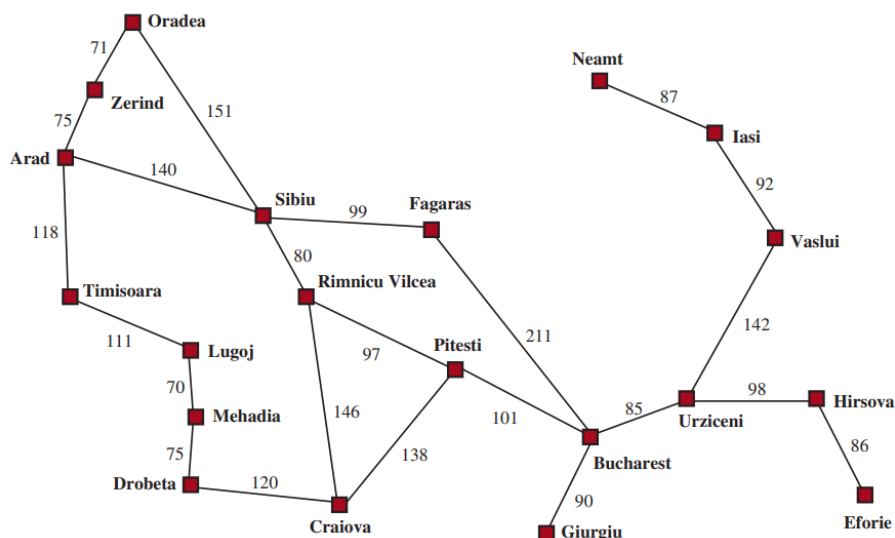


FIGURE 2.1 – Représentation simple de la Roumanie en graphe

Exemple: Voyage en Roumanie :

- **États** : les villes de Roumanie
- **État initial** : Arad
- **Actions** : les routes entre les villes adjacentes
- **Modèle de transition** : Atteindre une ville adjacente
- **Cout de l'action** : distance entre les villes
- **État final** : Bucharest