**Revision AI**

**TD1:**

Donnez une définition pour chacun des termes suivants :

1. Agent

2. Fonction d’un agent

3. Programme d’un agent

4. Rationalité

5. Agent réflexe simple

6. Agent réflexe avec état

7. Agent focalisé sur l’objectif

8. Agent focalisé sur l’utilité

**Solution**

**Agent :** Un agent est une entité autonome capable de percevoir son environnement, de traiter des informations, de prendre des décisions et d'agir sur cet environnement pour atteindre des objectifs.

**Fonction d’un agent :** La fonction d'un agent représente la relation entre les séquences de perceptions qu'il reçoit et les actions qu'il effectue en réponse. Elle définit le comportement de l'agent en fonction de son environnement.

**Programme d’un agent :** Le programme d'un agent est la mise en œuvre concrète de sa fonction. Il s'agit du code informatique, de l'algorithme ou de toute autre forme d'instructions qui permet à l'agent de traiter les informations, de prendre des décisions et d'effectuer des actions.

**Rationalité :** La rationalité d'un agent se réfère à sa capacité à prendre des décisions qui maximisent l'atteinte de ses objectifs, compte tenu de ses connaissances et de ses perceptions de l'environnement.

**Agent réflexe simple :** Un agent réflexe simple prend des décisions basées uniquement sur la perception immédiate de son environnement, sans tenir compte de l'historique des événements.

**Agent réflexe avec état :** Contrairement à l'agent réflexe simple, un agent réflexe avec état prend en compte l'historique des perceptions pour prendre des décisions. Il utilise l'information sur les états passés pour influencer ses actions présentes.

**Agent focalisé sur l’objectif** : Un agent focalisé sur l'objectif prend des décisions en fonction de la mesure dans laquelle chaque action le rapproche de ses objectifs spécifiques. Il est conscient de ses objectifs et cherche activement à les atteindre.

**Agent focalisé sur l’utilité :** Un agent focalisé sur l'utilité prend en compte non seulement la réalisation des objectifs, mais aussi l'utilité associée à chaque résultat possible. Il cherche à maximiser une mesure d'utilité spécifiée plutôt que simplement atteindre des objectifs.

**Exercice2**

Donnez une description du PEAS (Performance measure, Environment, Actuators, Sensors) pour chacun des agents intelligents suivant, ainsi qu’une description du type d’environnement :

1. Robot footballeur

2. Agent achetant des livres sur Internet

3. Véhicule autonome roulant sur Mars

4. Agent aidant un mathématicien à prouver un théorème

5. Agent jouant au sudoku

6. Agent jouant à la belote

7. Agent aidant au diagnostique médical

8. Agent analysant des images satellites

**Solution**

**Robot footballeur**

**Performance** :

score de l’équipe Environnement : ballon, joueurs, arbitre, terrain…

**Capteurs :** capteur auditif, capteur visuel, capteur sensitif…

Actionneurs : tête, les pieds, les mains….

**Les caractéristiques de l’environnement** :

• Partiellement observable (dépend des capteurs)

• Stochastique

• Séquentiel

• Dynamique

• Continu

• Multi-agents

**2. Agent achetant des livres sur Internet**

**Performance :** livres intéressants, livres pas chers, livraison à domicile

Environnement : serveur web, e-mail…

**Capteurs :** caméra, analyseur de pages web,

Actionneurs : clavier, souris …

**Les caractéristiques de l’environnement** :

• Complétement observable

• Déterministe

• Episodique (chaque achat est indépendant)

• Dynamique (site web dynamique)

• Discret

• Mono-agent

**3. Véhicule autonome roulant sur Mars**

Performance : nombres de photos effectuées, surface parcourue

Environnement : la planète Mars, station terrestre, l’atmosphère

**Capteurs :** capteurs visuels (caméra), capteurs sensitifs, thermomètre, boussole …

Actionneurs : moteur, roues, appareil photo, radio …

**Les caractéristiques de l’environnement :**

• Complétement observable (ou partiellement observable dépendant des capteurs)

• Stochastique

• Séquentiel

• Statique (si on néglige les changements climatiques)

• Continu

• Mono-agent

**4. Agent aidant un mathématicien à prouver un théorème**

**Performance :** temps de résolution, l’optimalité de la solution

Environnement : mathématicien, base de données des théorèmes existants

**Capteurs :** analyseur de fichiers, caméra ..

Actionneurs : clavier, souris, générateur de fichiers Les **caractéristiques de l’environnement :**

• Totalement observable

• Déterministe

• Episodique

• Statique

• Discret

• Mono-agent

5. Agent jouant au sudoku

Performance : temps, grille complète, score Environnement : grille, ordinateur, smartphone …

Capteurs : caméra, capteurs visuels Actionneurs : les mains, clavier, souris

Les caractéristiques de l’environnement :

• Complétement observable

•Déterministe

•Episodique

• Statique

• Discret

• Mono-agent

**TD2 :**

**Exercice1 :**

Donnez une définition pour chacun des termes suivants :

1. Etat

2. Espace d’états

3. Arbre de recherche

4. Nœud

5. But

6. Action

7. Fonction successeurs

8. Coefficient de branchement

**Solution :**

**État :** Un état représente une configuration ou une situation spécifique dans laquelle un système ou un agent peut se trouver à un moment donné. Dans le contexte de la recherche de solutions, un état est une configuration particulière d'un problème.

**Espace d'états :** L'espace d'états est l'ensemble de tous les états possibles d'un système ou d'un problème. C'est l'ensemble complet des configurations que le système peut atteindre.

**Arbre de recherche :** Un arbre de recherche est une structure arborescente utilisée pour représenter toutes les séquences possibles d'états et d'actions dans un problème donné. Chaque nœud de l'arbre représente un état, et les branches représentent les actions possibles.

**Nœud :** Dans le contexte d'un arbre de recherche, un nœud représente un état particulier du système à un moment donné. Chaque nœud peut avoir des nœuds enfants représentant les états suivants résultant des actions possibles.

**But :** Le but est l'objectif ou la condition que l'agent ou le système cherche à atteindre dans le cadre d'un problème. La recherche consiste à trouver une séquence d'actions conduisant à un état qui satisfait les conditions du but.

**Action :** Une action est une opération ou une étape pouvant être effectuée par l'agent pour passer d'un état à un autre dans le processus de recherche ou de résolution de problèmes.

**Fonction successeurs :** La fonction successeurs, également appelée fonction de transition, définit l'ensemble des états atteignables à partir d'un état donné en appliquant toutes les actions possibles. Elle associe à chaque action les états résultants.

**Coefficient de branchement :** Le coefficient de branchement représente le nombre moyen d'enfants (nœuds successeurs) pour chaque nœud dans un arbre de recherche. Il est utilisé pour évaluer la complexité de l'espace d'états et peut avoir un impact sur l'efficacité des algorithmes de recherche. Un faible coefficient de branchement indique généralement une exploration plus simple de l'espace d'états.

**Exercice 2**

Donnez l’état initial, le but, la fonction successeur et la fonction de coût pour chacun des problèmes suivants :

1. Vous devez colorier une carte de façon à ce que les pays adjacents ne soient pas de la même couleur, et en sachant que vous avez à votre disposition 4 couleurs distinctes

2. Un singe mesurant 1 mètre se trouve dans une pièce de 3 mètres de hauteur. Une banane est suspendue au plafond de cette pièce, et le singe aimerait bien avoir cette banane. La pièce contient également 2 caisses qu’il peut déplacer et sur lesquelles il peut monter, chaque caisse mesurant 1 mètre

3. On a trois récipients pouvant contenir respectivement 3, 8 et 12 litres, et un robinet d’eau. On peut remplir les récipients, ou verser entièrement leur contenu dans un autre récipient ou sur le sol. Nous voulons obtenir exactement 1 litre d’eau

**Solution :**

**Problème 1:** Coloriage de carte

**État initial :** La carte non coloriée. But : Colorier la carte de manière à ce que les pays adjacents n'aient pas la même couleur, en utilisant 4 couleurs distinctes.

**Fonction successeur :** Pour chaque région non encore coloriée, choisir une couleur qui n'est pas utilisée par ses voisins.

**Fonction de coût :** Peut être basée sur la minimisation du nombre total de couleurs utilisées.

**Problème 2:** Singe et banane

**État initial :** Le singe est au sol, les caisses sont également au sol, la banane est suspendue au plafond.

**But :** Le singe atteint la banane.

**Fonction successeur :** Le singe peut se déplacer, monter sur une caisse, déplacer une caisse, sauter.

**Fonction de coût :** Peut être basée sur le nombre d'actions nécessaires ou la distance parcourue.

**Problème 3:**

Mesure d'eau

**État initial :** Les trois récipients sont vides. But : Avoir exactement 1 litre d'eau dans l'un des récipients.

**Fonction successeur :** Remplir un récipient, vider un récipient, verser l'eau d'un récipient à un autre.

**Fonction de coût :** Peut être basée sur le nombre d'actions nécessaires ou la quantité d'eau déplacée.

**Exercice 3 :**

Considérez un espace de recherche dans lequel l’état initial est 1 et la fonction successeur pour un nœud n retourne deux états contenant les entiers 2n et 2n+1.

1. Dessiner la partie de l’espace de recherche contenant les nœuds de 1 à 15

2. Supposer que le but soit 11. Donner l’ordre de parcours des nœuds pour les algorithmes :

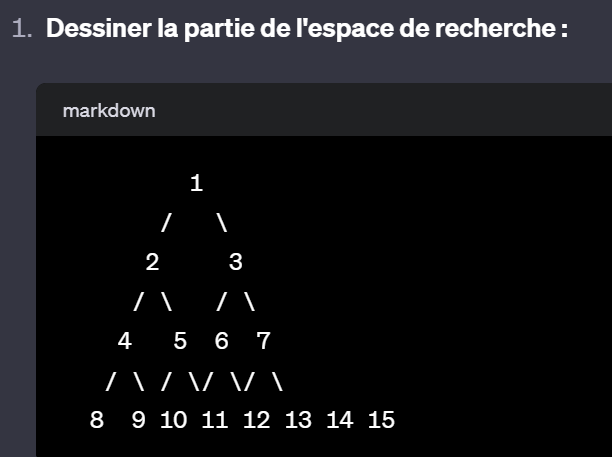
(a) largeur d’abord

(b) profondeur d’abord

(c) profondeur d’abord limitée à 2

(d) profondeur itérative

**Solution :**



Ordre de parcours des nœuds :

**(a) Largeur d'abord (BFS) :** 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11

**(b) Profondeur d'abord (DFS) :** 1, 2, 4, 8, 9, 5, 10, 11

**(c) Profondeur d'abord limitée à 2 :** 1, 2, 4, 8, 9, 5, 10, 11

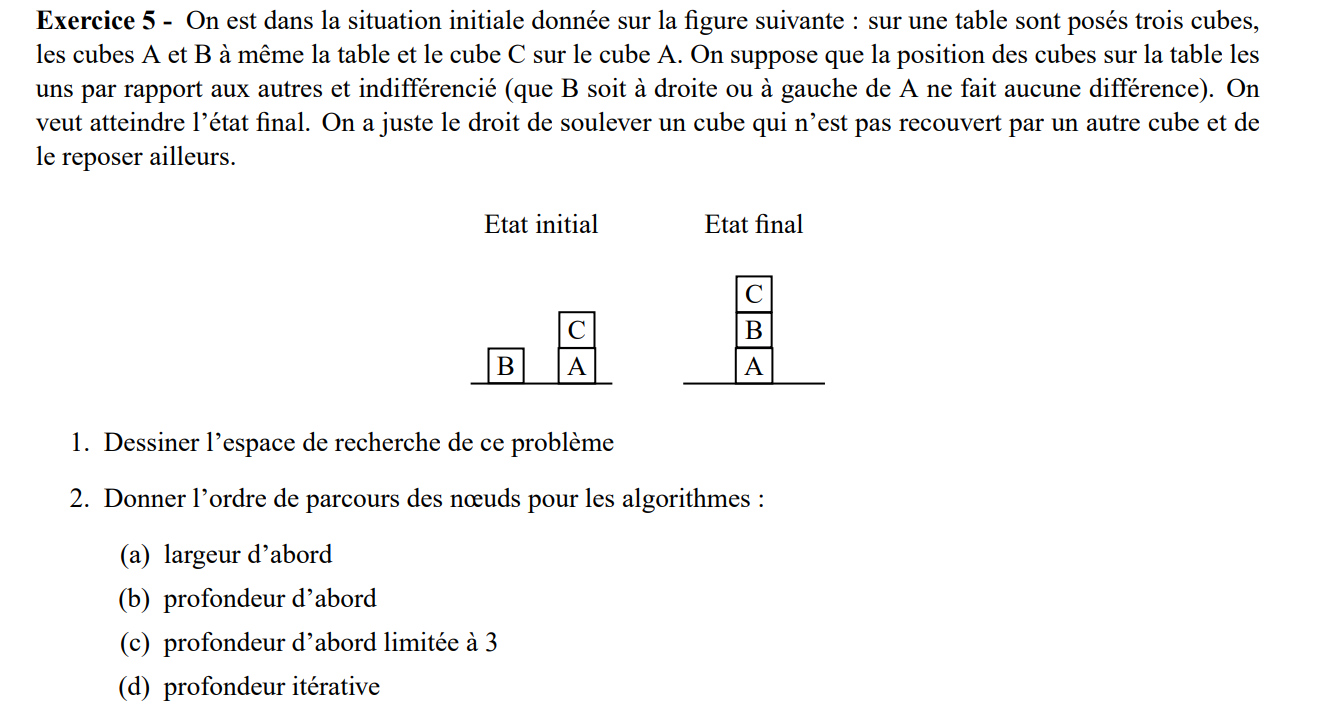
**(d) Profondeur itérative :**

Profondeur 1: 1, 2, 3

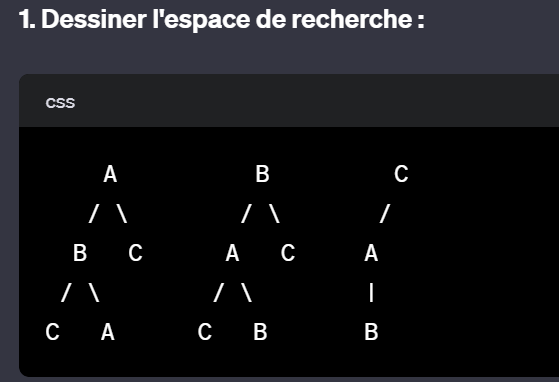
Profondeur 2: 1, 2, 4, 5, 3, 6, 7

Profondeur 3: 1, 2, 4, 8, 9, 5, 10, 11

**Exercice 4**



**Solution :**



**2. Ordre de parcours des nœuds :**

(a) Largeur d'abord (BFS) :

État initial: ABC

Profondeur 1: BCA, CAB

Profondeur 2: BAC, CBA, ACB

Profondeur 3: Solution (ABC)

(b) Profondeur d'abord (DFS) :

État initial: ABC

Profondeur 1: BCA

Profondeur 2: BAC

Profondeur 3: Solution (ABC)

**(c) Profondeur d'abord limitée à 3 :**

État initial: ABC

Profondeur 1: BCA

Profondeur 2: BAC, CBA

Profondeur 3: Solution (ABC)

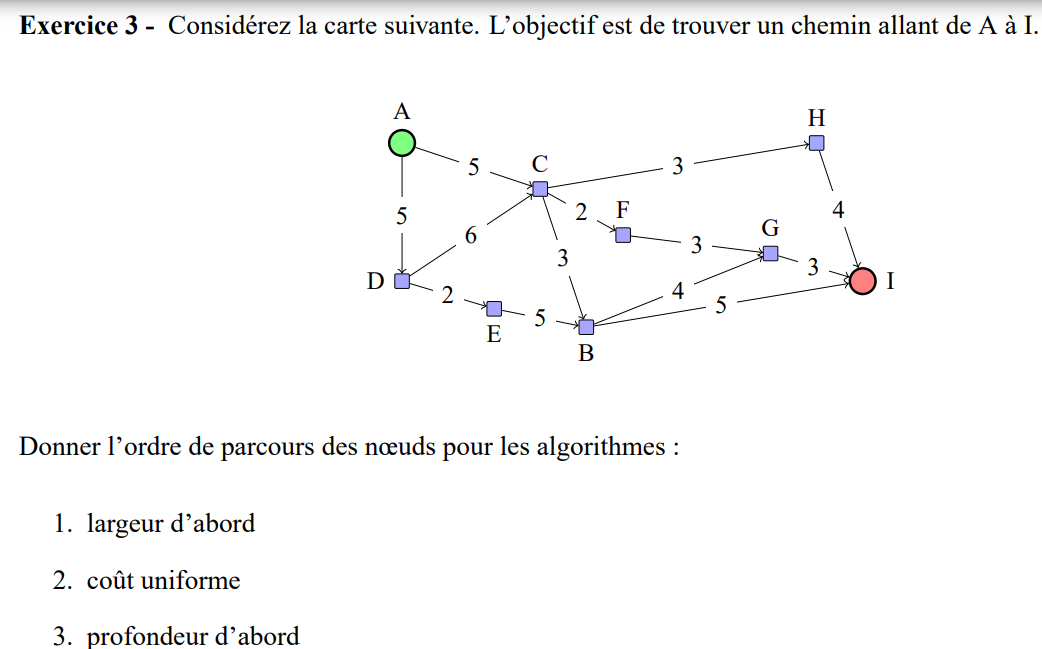
**(d) Profondeur itérative :**

Profondeur 1: BCA, CAB

Profondeur 2: BAC, CBA, ACB

Profondeur 3: Solution (ABC)

**Exercice 4**



**1. Largeur d'abord (BFS) : A, C, D, F, E, B, G, H, I**

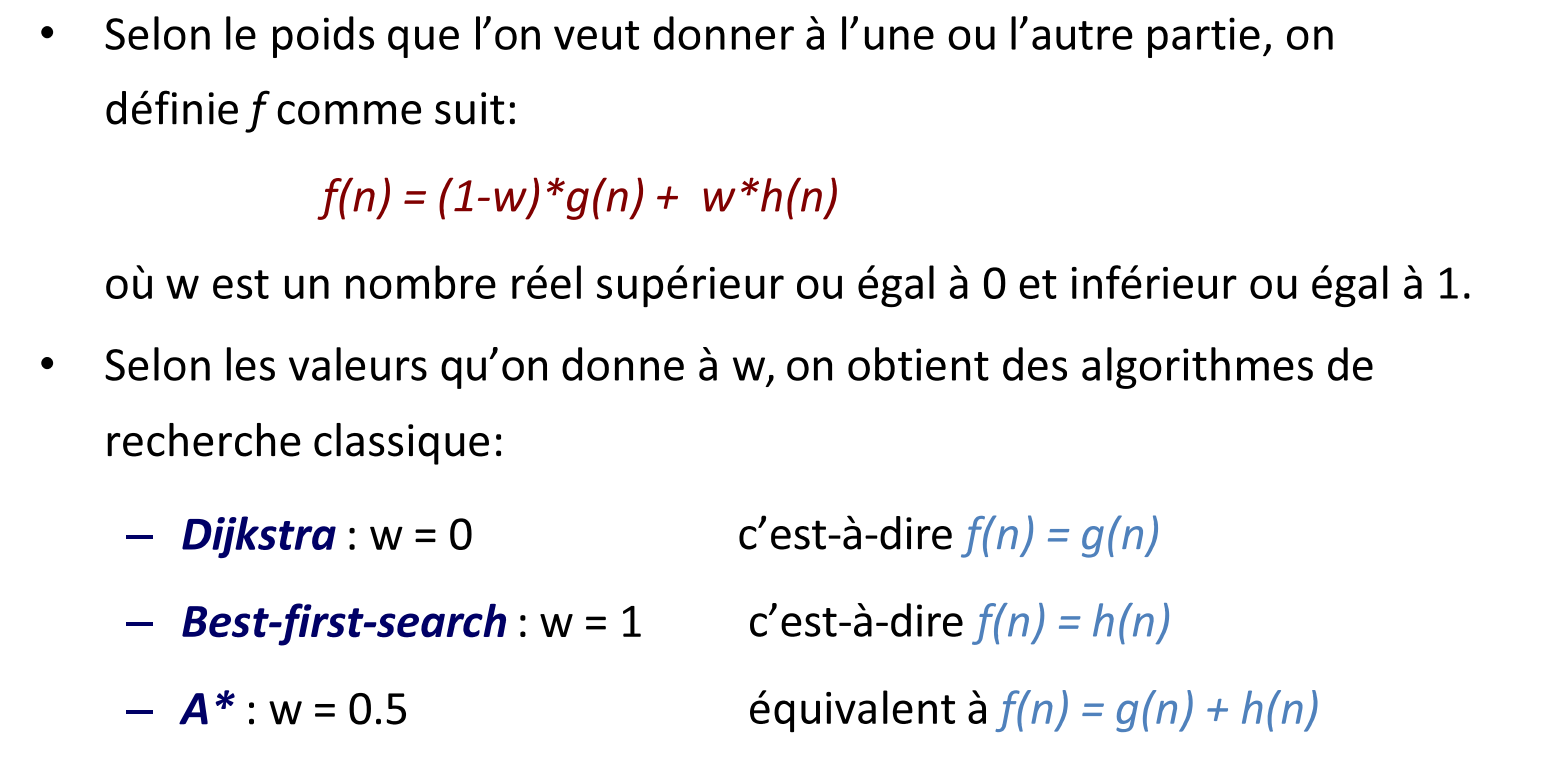
**2. Coût uniforme : A, D, E, F, C, G, H, I, B 3.**

**Profondeur d'abord (DFS) : A, C, F, G, I, H, B, D, E**

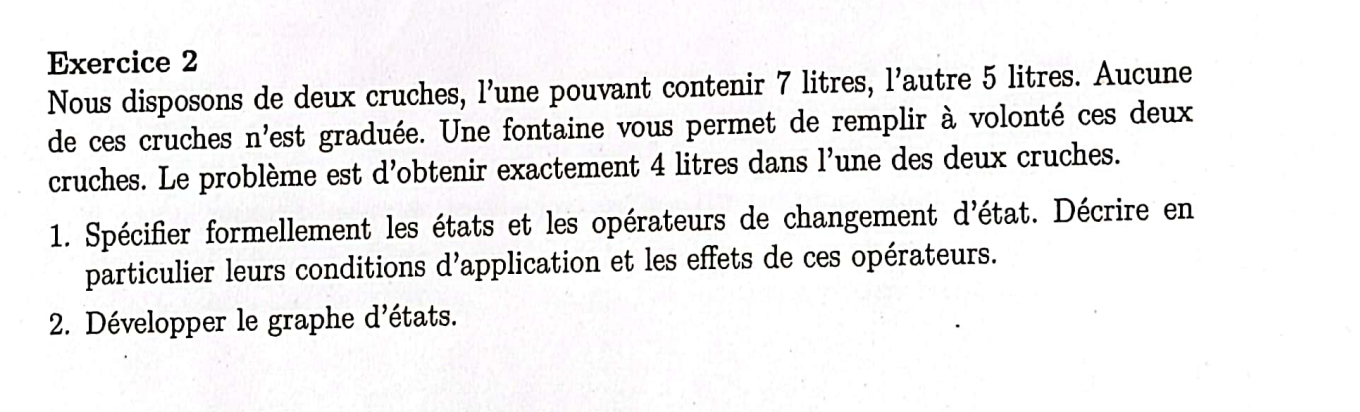
**Recherche non informée :**

Recherche en largeur (Breadth-First-Search) Recherche en profondeur (Depth-First-Search) Sans limite de profondeur (version classique) Avec limite de profondeur Incrémentale (Iterative Depth-First-Search) Algorithme de Dijkstra

**Recherche informée :** A\*



**Examen :**



1. **Spécification formelle des états et des opérateurs :**
2. **États :** Chaque état peut être représenté par un couple (x, y), où x est la quantité d'eau dans la première cruche (cruche de 7 litres) et y est la quantité d'eau dans la deuxième cruche (cruche de 5 litres). Les états valides sont ceux où x et y sont des nombres entiers compris entre 0 et la capacité respective des cruches (7 pour la première, 5 pour la deuxième). L'ensemble des états possibles est donc défini comme suit : E ˊ tats = { ( � , � ) ∣ 0 ≤ � ≤ 7 , 0 ≤ � ≤ 5 } E ˊ tats={(x,y)∣0≤x≤7,0≤y≤5} Opérateurs : Remplissage de la première cruche (opérateur � 1 F 1 ​ ): Condition d'application : � < 7 x<7 Effet : � x devient 7, et � y reste inchangé. Remplissage de la deuxième cruche (opérateur � 2 F 2 ​ ): Condition d'application : � < 5 y<5 Effet : � y devient 5, et � x reste inchangé. Vider la première cruche (opérateur � 1 E 1 ​ ): Condition d'application : � > 0 x>0 Effet : � x devient 0, et � y reste inchangé. Vider la deuxième cruche (opérateur � 2 E 2 ​ ): Condition d'application : � > 0 y>0 Effet : � y devient 0, et � x reste inchangé. Transvaser de la première cruche à la deuxième (opérateur � 12 T 12 ​ ): Condition d'application : � > 0 x>0 et � < 5 y<5 Effet : On verse l'eau de � x dans la deuxième cruche jusqu'à ce qu'elle soit pleine ou que la première cruche soit vide. Transvaser de la deuxième cruche à la première (opérateur � 21 T 21 ​ ): Condition d'application : � > 0 y>0 et � < 7 x<7 Effet : On verse l'eau de � y dans la première cruche jusqu'à ce qu'elle soit pleine ou que la deuxième cruche soit vide. 2) Graphe d'états : Le graphe d'états peut être construit en représentant chaque état comme un nœud et en reliant les états atteignables par les opérateurs avec des arêtes dirigées. La première cruche est représentée par l'axe horizontal, et la deuxième cruche par l'axe vertical. Chaque arête est annotée avec l'opérateur correspondant. Voici une représentation simplifiée du graphe d'états :

