

## Corrigé de l'examen de mi-session

### Intelligence artificielle II (IFT-17587)

1. a. Simple-réflexe : l'agent ne fait qu'obéir à des règles simples. Par exemple, si la pression dépasse une certaine valeur, ouvrir la valve.  
b. But : L'agent a le but de sortir du labyrinthe.  
c. Utilité : Il y a plusieurs alternatives et de l'incertain.  
d. Utilité : Il y a des décisions à prendre entre la sécurité et le rendement et il y de l'incertain.  
e. Réflexe : Un agent réflexe est suffisant pour cette tâche, mais j'accepte aussi réflexe avec état interne si on tient compte que l'agent doit se souvenir si les armoires sont pleines.

2. a. Il y a plusieurs manières possibles de formuler un problème de recherche. Mais voici une des réponses possibles :

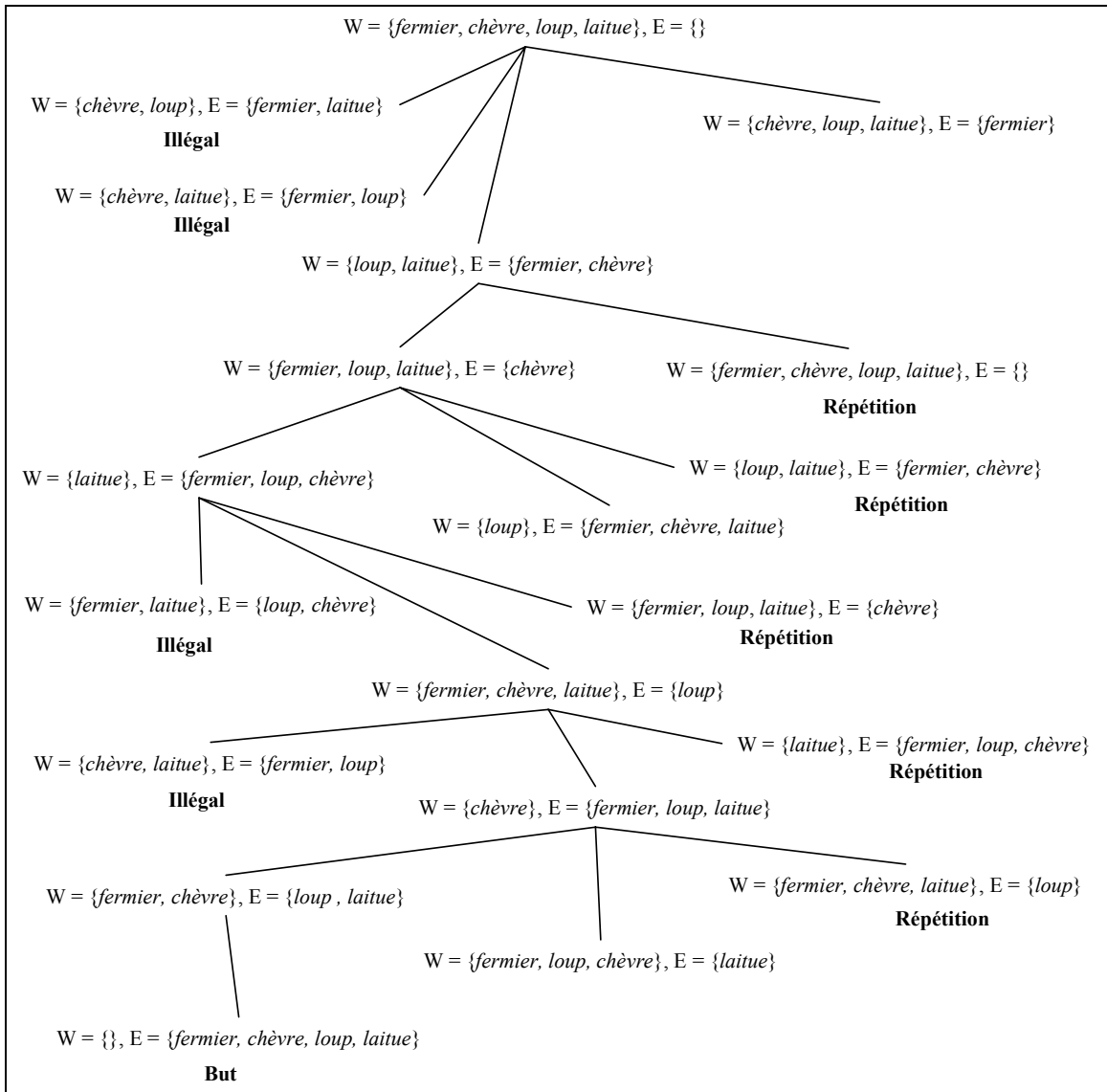
Représentation des états : Les états peuvent être représentés par deux ensembles O et E contenant les éléments sur la rive ouest et est respectivement. Pour les éléments, on peut utiliser les symboles *fermier*, *chèvre*, *loup* et *laitue*.

Opérateurs : Déplacer *fermier* de l'ensemble E à l'ensemble O et vice versa. Déplacer *fermier* et un des éléments *chèvre*, *loup* ou *laitue* de l'ensemble E à l'ensemble O et vice versa.

État de départ :  $O = \{ \textit{fermier}, \textit{chèvre}, \textit{loup}, \textit{laitue} \}$ ,  $E = \{ \}$

État but :  $O = \{ \}$ ,  $E = \{ \textit{fermier}, \textit{chèvre}, \textit{loup}, \textit{laitue} \}$

b. Voici un arbre de recherche possible.



3. a. Le code devrait vérifier si  $Fl$  est un but dès qu'il est sortie de la liste. Le code ne devrait pas vérifier si les enfants sont des buts. Ceci fait en sorte de contourner la liste qui est triée en fonction de la fonction de coût. De cette manière, un fils qui est une solution coûteuse peut être choisie avant une meilleure solution qu'on aurait trouver si on avait utilisé la liste.

b. Le bon pseudo-code est le suivant :

$F = \{D\}$

**Tant que**  $F$  n'est pas vide

Prendre  $FI$ , le premier élément dans  $F$

**Si**  $FI$  est un but

Retourner Succès et Sortir

$Nœuds\_enfants = développer(FI)$

Éliminer les nœuds dans  $Nœuds\_enfants$  qui ont déjà été visités

Ajouter les enfants à  $F$

Trier  $F$  selon la fonction suivante :  $f = coûtChemin(D \text{ à } nœud) + h(nœud)$

**Fin tant que**

4. Gêne : Points à visiter

Individu : Liste de points (chaque individu est un parcours possible)

Population : Un ensemble de parcours

Opérateur de reproduction :

- On choisit aléatoirement deux points de coupe.
- On intervertit, entre les deux individus, les parties qui se trouvent entre ces deux points.
- On supprime, à l'extérieur des points de coupes, les gènes qui sont déjà placés entre les points de coupe.
- On recense les gènes qui n'apparaissent pas dans chacun des deux parcours.
- On remplit aléatoirement les trous dans chaque parcours.

Opérateur de mutation : On choisit aléatoirement deux gènes dans l'individus et on les intervertit.

Évaluation des individus : Plus le chemin est court, plus l'individu est fort. Donc, on peut prendre l'inverse de la distance comme fonction d'utilité.

Pour le fonctionnement, il suffit d'expliquer le fonctionnement des algorithmes génétiques.

5. Voici les règles à ajouter. Pour qu'il n'y ait pas de conflit entre les règles, il fallait ajouter un nouveau prédicat RetournerDebut qui indique si l'agent a atteint la case (2,2) ou non. Lorsque l'agent atteint la case (2,2), RetournerDebut est mis à vrai.

$\text{Pos}(x,y) \wedge \text{Saleté}(x,y) \rightarrow \text{Faire}(\text{aspirer})$

$\text{Pos}(0,0) \wedge \text{Orientation}(\text{Nord}) \wedge \neg \text{Saleté}(0,0) \rightarrow \text{Faire}(\text{avancer})$

$\text{Pos}(0,1) \wedge \text{Orientation}(\text{Nord}) \wedge \neg \text{Saleté}(0,1) \rightarrow \text{Faire}(\text{avancer})$

$\text{Pos}(0,2) \wedge \text{Orientation}(\text{Nord}) \wedge \neg \text{Saleté}(0,2) \rightarrow \text{Faire}(\text{tourner})$

$\text{Pos}(0,2) \wedge \text{Orientation}(\text{Est}) \rightarrow \text{Faire}(\text{avancer})$

$\text{Pos}(1,2) \wedge \text{Orientation}(\text{Est}) \wedge \neg \text{Saleté}(1,2) \rightarrow \text{Faire}(\text{tourner})$

$\text{Pos}(1,2) \wedge \text{Orientation}(\text{Sud}) \rightarrow \text{Faire}(\text{avancer})$

$\text{Pos}(1,1) \wedge \text{Orientation}(\text{Sud}) \wedge \neg \text{Saleté}(1,1) \rightarrow \text{Faire}(\text{avancer})$

$\text{Pos}(1,0) \wedge \text{Orientation}(\text{Sud}) \wedge \neg \text{Saleté}(1,0) \rightarrow \text{Faire}(\text{tourner})$

$\text{Pos}(1,0) \wedge \text{Orientation}(\text{Ouest}) \wedge \neg \text{RetournerDebut} \rightarrow \text{Faire}(\text{tourner})$

$\text{Pos}(1,0) \wedge \text{Orientation}(\text{Nord}) \rightarrow \text{Faire}(\text{tourner})$

$\text{Pos}(1,0) \wedge \text{Orientation}(\text{Est}) \rightarrow \text{Faire}(\text{avancer})$

$\text{Pos}(2,0) \wedge \text{Orientation}(\text{Est}) \wedge \neg \text{Saleté}(2,0) \rightarrow \text{Faire}(\text{tourner})$

$\text{Pos}(2,0) \wedge \text{Orientation}(\text{Sud}) \wedge \neg \text{RetournerDebut} \rightarrow \text{Faire}(\text{tourner})$

$\text{Pos}(2,0) \wedge \text{Orientation}(\text{Ouest}) \wedge \neg \text{RetournerDebut} \rightarrow \text{Faire}(\text{tourner})$

$\text{Pos}(2,0) \wedge \text{Orientation}(\text{Nord}) \rightarrow \text{Faire}(\text{avancer})$

$\text{Pos}(2,1) \wedge \text{Orientation}(\text{Nord}) \wedge \neg \text{Saleté}(2,1) \rightarrow \text{Faire}(\text{avancer})$

$\text{Pos}(2,2) \wedge \text{Orientation}(\text{Nord}) \wedge \neg \text{Saleté}(2,2) \rightarrow \text{Faire}(\text{tourner})$

$\text{Pos}(2,2) \wedge \text{Orientation}(\text{Est}) \rightarrow \text{Faire}(\text{tourner})$

$\text{Pos}(2,2) \wedge \text{Orientation}(\text{Sud}) \rightarrow \text{Faire}(\text{avancer})$

$\text{Pos}(2,1) \wedge \text{Orientation}(\text{Sud}) \rightarrow \text{Faire}(\text{avancer})$

$\text{Pos}(2,0) \wedge \text{Orientation}(\text{Sud}) \wedge \text{RetournerDebut} \rightarrow \text{Faire}(\text{tourner})$

$\text{Pos}(2,0) \wedge \text{Orientation}(\text{Ouest}) \wedge \text{RetournerDebut} \rightarrow \text{Faire}(\text{avancer})$

$\text{Pos}(1,0) \wedge \text{Orientation}(\text{Ouest}) \wedge \text{RetournerDebut} \rightarrow \text{Faire}(\text{avancer})$

6. Voici les calculs

$$y_2 = (3 * 0,3) + (1 * -1,1) = -0,2 \quad o_2 = -1$$

$$y_3 = (0,1 * -1) + (1 * -0,5) = -0,6 \quad o_3 = -1$$

$$y_4 = (0,6 * -1) + (1 * -0,9) = -1,5 \quad o_4 = -1$$

$$\delta_3 = -1(1 + 1)(1 + 1) = -4$$

$$\delta_4 = -1(1 + 1)(1 + 1) = -4$$

$$\delta_2 = -1(1 + 1)((0,1 * -4) + (0,6 * -4)) = 5,6$$

$$\Delta w_{21} = 0,05 * 5,6 * 3 = 0,84$$

$$\Delta w_{25} = 0,05 * 5,6 * 1 = 0,28$$

$$\Delta w_{32} = 0,05 * -4 * -1 = 0,2$$

$$\Delta w_{36} = 0,05 * -4 * 1 = -0,2$$

$$\Delta w_{42} = 0,05 * -4 * -1 = 0,2$$

$$\Delta w_{47} = 0,05 * -4 * 1 = -0,2$$

$$w_{21} = 0,3 + 0,84 = 1,14$$

$$w_{25} = -1,1 + 0,28 = -0,82$$

$$w_{32} = 0,1 + 0,2 = 0,3$$

$$w_{36} = -0,5 + -0,2 = -0,7$$

$$w_{42} = 0,6 + 0,2 = 0,8$$

$$w_{47} = -0,9 + -0,2 = -1,1$$