

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

السَّلَامُ عَلَيْكُمْ وَرَحْمَةُ اللَّهِ وَبَرَكَاتُهُ

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Abdelhafid Boussouf University Center of Mila

1^{ère} année Master

Cours Intelligence Artificielle et ses Applications



Chapitre 01

Partie 2

Responsable module

Dr. MEGUEHOUT Hamza



Machine déterministe

Humain **résout un problème** → Machine **exécute la solution**

Machine/Système algorithmique (non intelligent)

- ✓ Système d'exécution de la solution ;
- ✓ Machine déterministe à états finis ;
- ✓ À tout moment durant l'exécution, la prochaine instruction à exécuter est bien déterminée.





Machine Non déterministe

Humain **définit le problème** → Machine **le résout**

Systeme de résolution (automatique) de Problèmes

- ✓ La Machine/Systeme est appelée à trouver → la solution à un problème donné (Systeme de résolution automatique de problèmes) ;
- ✓ Espace de recherche (espace d'états, de buts, ...)
- ✓ Stratégies de recherche ;
- ✓ Exploration de l'espace ;
- ✓ Prise de décision, devant différents choix.





Agents



Master I2A

Matière Intelligence artificielle : Principes et Applications



Agent de résolution

Nous pouvons dire qu'un **AGENT de résolution de problèmes**



EST un agent **dirigé** sur les résultats



se concentre toujours sur la **Satisfaction Des Objectifs**.





Qu'est-ce qu'un problème?



Master I2A

Matière Intelligence artificielle : Principes et Applications



Un problème

Collection d'**informations** que l'agent utilise pour décider quelles **actions** accomplir





Comment définir un problème ?



Master I2A

Matière Intelligence artificielle : Principes et Applications



Définir un problème (1)

1. Formulation **d'un but (état à atteindre)**
2. Formulation des **états et les actions** à considérer
3. Exploration des solutions → examiner les différentes séquences d'actions menant à un **état but** et **choisir la meilleure**





Définir un problème (2)

- ✓ Identifier un **état initial**
- ✓ Identifier les **actions possibles** par définition les opérateurs de changement d'état (*l'ensemble des états possibles du problème*)
- ✓ Une **fonction de successeur**, qui définit l'état résultant de l'exécution d'une **ACTION** dans un état
- ✓ Un ensemble d'**états buts**
- ✓ Une **fonction coût**, associant à chaque action un nombre non négatif (le coût de l'action)





Définir un problème (3)

Taquin

4	7	1
8		3
5	6	2



1	2	3
4	5	6
7	8	

- ✓ **État initial** → Les huit tuiles dans n'importe quelle case
- ✓ **Actions** → Déplacement gauche, haut, bas,...
- ✓ **Test de but** → Un état qui correspond à l'état final





Définir un problème (4)

Taquin

Les algorithmes d'aujourd'hui arrivent à résoudre les taquins 3×3 et 4×4 (espaces d'états de taille **181440** et d'environ **1,3 milliard** respectivement)

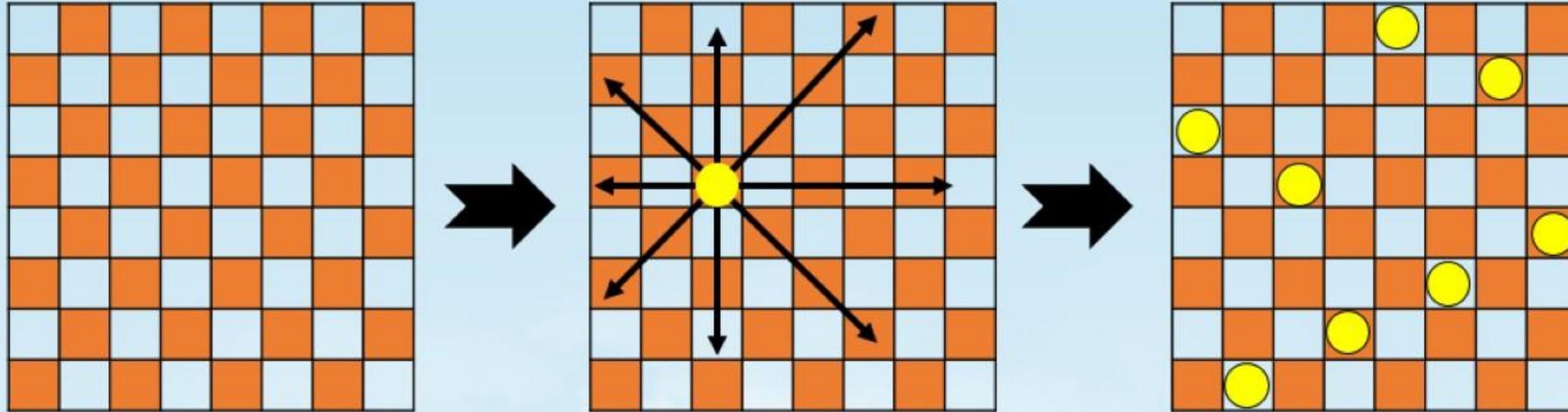
Mais les instances du taquin 5×5 (avec un espace d'états de taille **10^{25}**)

Restent Difficiles



Définir un problème (5)

8 reines



- ✓ **État initial** Aucune reine sur l'échiquier
- ✓ **Actions** Ajouter une reine sur une case vide
- ✓ **Test de but** Les 8 reines sont placées sur l'échiquier sans attaque



Définir un problème par un graphe

Problème comme un **graphe** orienté

- Les **nœuds** sont des **états** accessibles depuis l'**état initial**
- Les **arcs** sont des **actions**

Nous appellerons ce graphe l'**espace des états**





Définir un problème par un graphe

Solution sera un **chemin de l'état initial** à un **état but**

Une **solution est optimale**, si la **somme** des coûts des actions du chemin est **minimale** parmi toutes les solutions du problème





Définir un problème par un graphe

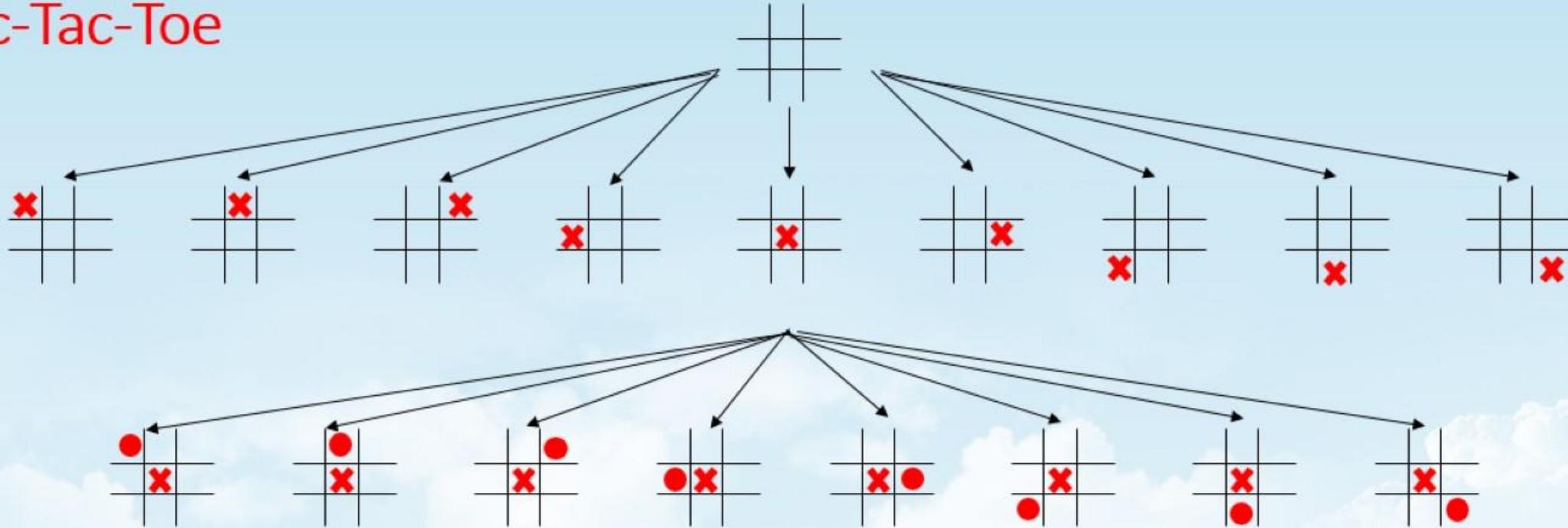
- ✓ Les **nœuds** représentent les **états**
- ✓ Un **arc (i,j)** représente **Opération/Action** permettant d'aller de l'état **i** à l'état **j**
- ✓ **Solution** → chemin entre l'**état initial** et l'**état final**
- ✓ **Recherche de Solution** → Recherche d'un chemin entre l'**état initial** et l'**état final**





Définir un problème par un arbre

Jeu Tic-Tac-Toe



Nombre d'états possibles





Modélisation d'un problème

Modélisation par (E, e_0, Ef, T) :

- ✓ $E \rightarrow$ l'ensemble de tous les états
- ✓ $e_0 \rightarrow$ l'état initial, $e_0 \in E$
- ✓ $Ef \rightarrow$ l'ensemble des états finaux $Ef \subset E$
- ✓ $T \rightarrow$ la fonction de transition : associe à chaque état e_i un ensemble de couples (A_{ij}, e_{ij})
 A_j soit une action élémentaire permettant de passer de l'état e_i à l'état e_{ij} .





Un état

Un ensemble de **variables** d'états.

Les **valeurs attribuées** à ces **variables** permettront alors de définir un état donné

Ces variables décrivent généralement les attributs des différents objets (situations,...) de l'environnement du problème





C'est quoi la résolution des problèmes en intelligence artificielle ?



Master I2A

Matière Intelligence artificielle : Principes et Applications



Résolution des problèmes

Dans cette approche le principe général de l'IA est de considérer qu'il existe des méthodes générales permettant de résoudre n'importe quel type de problème. L'algorithme se doit donc d'être "**neutre**" sur le domaine concerné, et tout ce qui concerne les connaissances de description du problème et de sa résolution doit être clairement séparé de l'algorithme





Résolution des problèmes

Une tentative de programmation
des ordinateurs pour

Faire ce que l'humain fait mieux

Faire des choses dont on dit
qu'elle nécessite de
l'intelligence quand elles sont
faites par des humains

Étude des calculs qui rendent possible
la **perception**, le **raisonnement** et l'**action**





Résolution des problèmes

Une partie de l'intelligence artificielle



Englobe un certain nombre de techniques
un arbre, B-tree, des algorithmes heuristiques



Pour résoudre un problème.





Résolution des problèmes

Le problème de l'IA est directement lié à la nature des humains et à leurs activités

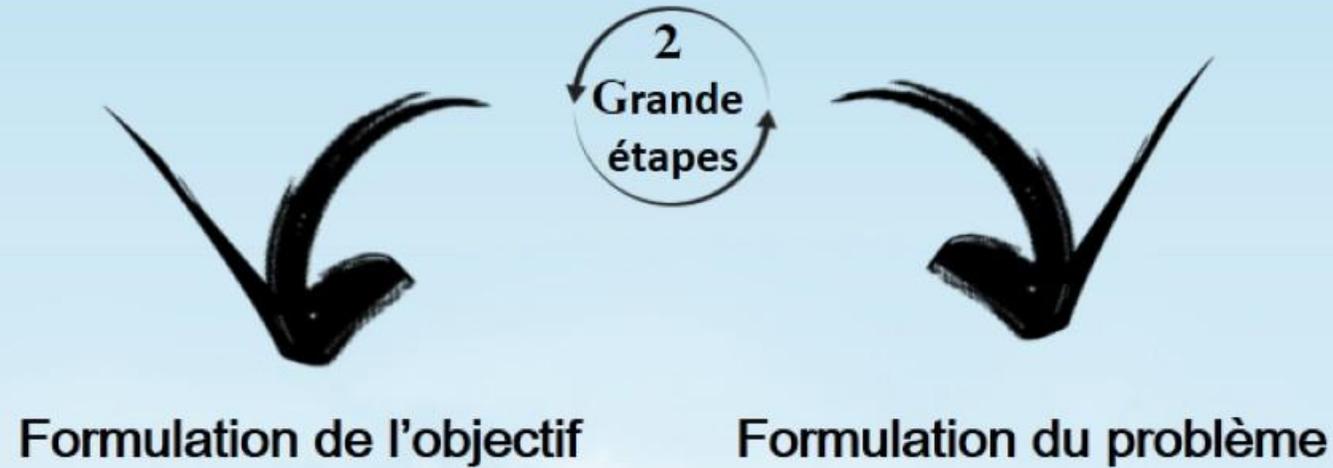
↓DONC↓

Nous avons besoin d'un certain nombre d'étapes finies pour résoudre un problème qui rend les travaux humains faciles





Résolution des problèmes



Simple étape de la résolution de problèmes. Il organise des étapes finies pour formuler une cible / des objectifs qui nécessitent une action pour atteindre l'objectif

Étapes fondamentales de la résolution de problème qui décide de l'action à entreprendre pour atteindre l'objectif formulé





Exemples de problèmes



Master I2A

Matière Intelligence artificielle : Principes et Applications



Toys Problèmes

Des problèmes types utilisés pour tester des algorithmes

le taquin, le chien, la chèvre et le chou, le Wumpus, l'aspirateur, le voyageur de commerce, le labyrinthe, les mots croisés, l'arithmétique cryptée, les jeux d'échecs, de dames, etc.





Toys Problèmes

- Problèmes de jeux bien définis ;
- La modélisation est facile ;
- Intéressant pour comparer les différentes stratégies de résolution.





Problèmes du monde réel

- Calcul de routes, voyageur de commerce, navigation de robots ;
- Difficiles à résoudre dans le cas général (*trop de paramètres*) d'où l'importance de la modélisation.
- Résolution de problèmes complexes, comme les problèmes d'allocation de ressources : *Préparation automatisée des plannings des enseignants/salles/classes en collège et lycée, optimisation des services d'urgences.*





Exemple d'un problème (1)

Le problème du **Fermier**, du **Loup**, de la **Chèvre**, du **Chou** et la traversée de la rivière

Un fermier avec son loup, sa chèvre et son chou veulent traverser la rivière. Il y a un canot à bord, mais seul le fermier peut ramer, et il ne peut contenir que deux entités à la fois (y compris le rameur). Si le loup reste seul avec la chèvre, il va la manger, si la chèvre reste seule avec le chou, elle va le manger. Que devrait être la séquence des traversées pour que les quatre soient sur l'autre bord





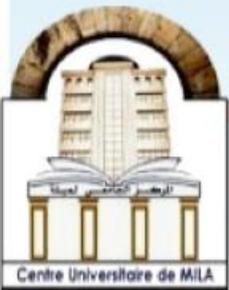
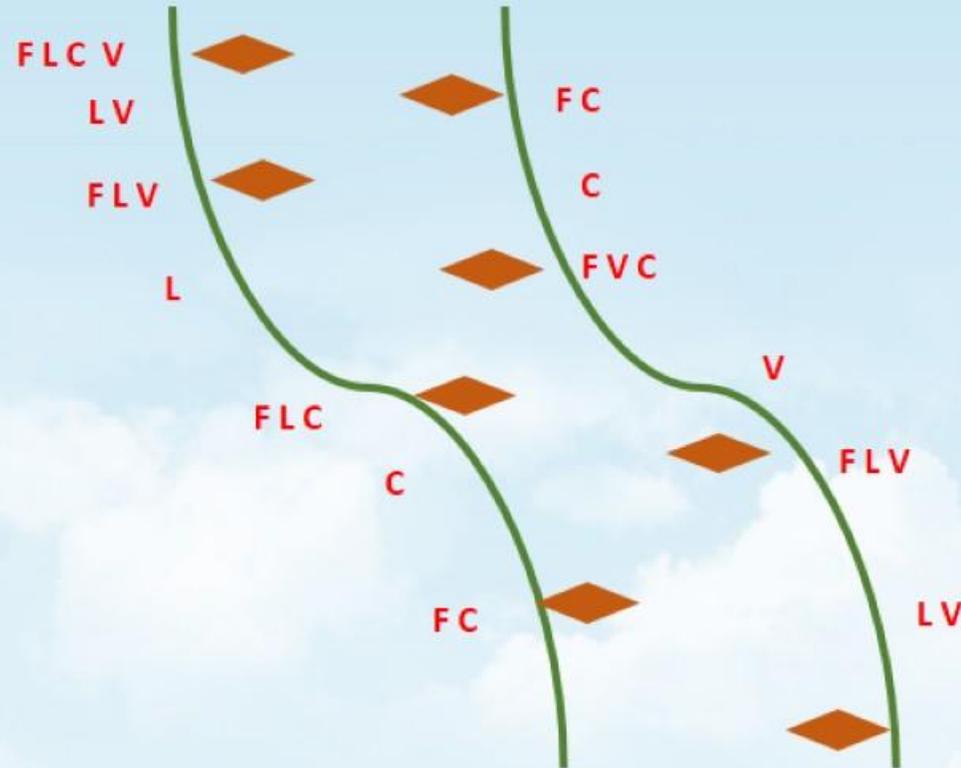
Exemple d'un problème (1)

Fermier → F

Loup → L

Chèvre → C

Chou → V





Exemple d'un problème (2)

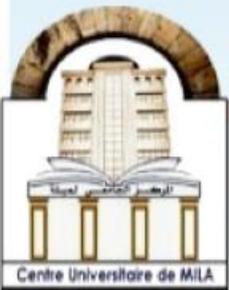
Se rendre de son domicile (Mila) à un lieu précis (Arrivée) d'une autre ville (Alger « Musée الشهيد »), le plus rapidement possible, avant une date donnée (obligation de passer par une ville étape B).
(Exemple : de Mila au Musée d'Alger avant 10h)



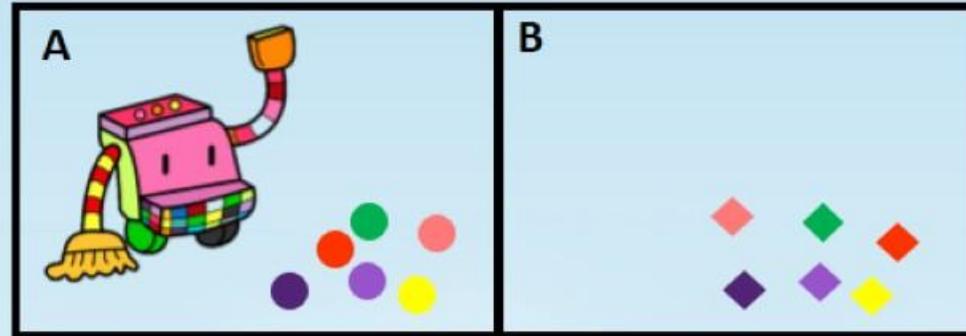


Exemple d'un problème (2)

- ✓ **État initial** : La station (...) d'où part le client, la date et l'heure.
- ✓ **Actions** : Le trajet d'une willaya ou station à l'autre.
- ✓ **Test de but** : Les états où le client est au lieu de sa destination.
- ✓ **Etas** : Chaque état est composé d'une station (...), la date et l'heure actuelle.
- ✓ **Coût des actions** : Va dépendre des préférences du client. Ce pourrait être 1 pour chaque action (pour minimiser le nombre d'action), ou la durée des voyages (pour minimiser le temps), ou le prix des trajets (pour trouver le voyage le moins cher).



Exemple d'un problème (3)



- ✓ **État initial** : Un état aléatoire.
- ✓ **Actions** : Gauche, droite et aspirer.
- ✓ **Test de but** : Vérifie que le sol est propre.
- ✓ **Etas** : Les emplacements du robot et de la poussière.
- ✓ **Coût des actions** : 1 par déplacement.

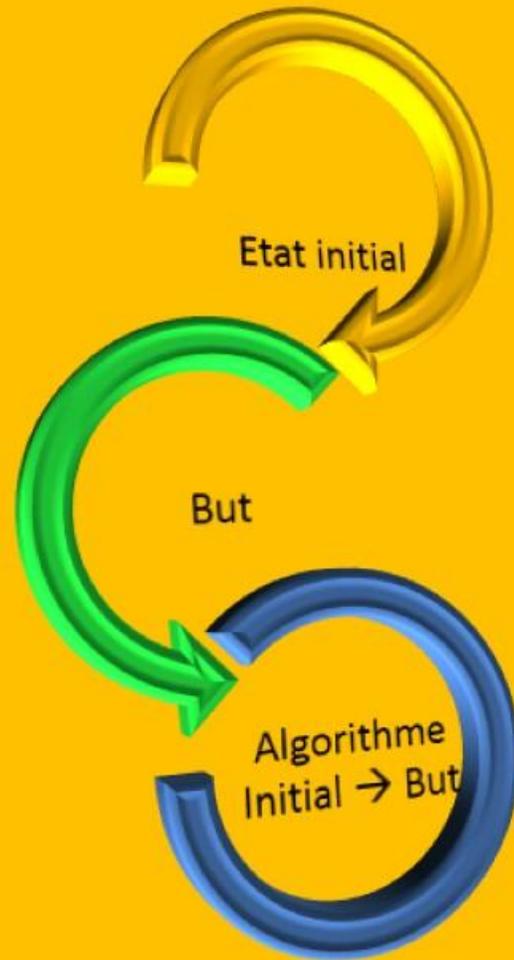
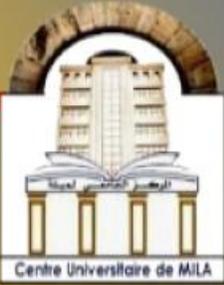


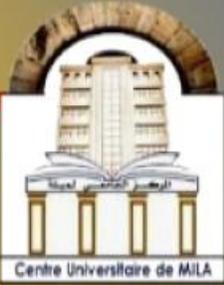
Quelle est la différence en Taquin et 8 reines

- Avec le taquin, nous savons depuis le début quel état nous voulons, et la difficulté est de trouver une séquence d'actions pour l'atteindre.
- Le problème des huit reines, nous ne sommes pas intéressés par le chemin, mais seulement par l'état but obtenu.

Ces deux jeux sont des exemples de deux grandes classes de problèmes étudiés en IA : des **problèmes de planification** et des problèmes de **satisfaction de contraintes**.







Représentation formelle des états du problème (*l'ensemble des variables d'états*): cette étape peut faire appel à l'un des formalismes de représentation de connaissances ou directement à des structures de données comme les listes et les tableaux.

Le choix de l'algorithme de recherche qui va choisir quels opérateurs va-t-il appliquer et sur quels états du problème en vue de trouver **l'état but** le plus vite possible. Cet algorithme doit donc disposer de procédures de test des états pour déterminer s'il a **oui** ou **non** atteint son but.



*Tout est possible à qui rêve, ose, travaille et
n'abandonne jamais*

