LABORATOIRE D'INFORMATIQUE INTELLIGENTE BISKRA

LABORATOIRE D'INFORMATIQUE INTELLIGENTE BISKRA

Cours IAA

chapitre 3: Notion de recherche dans un espace d'état

Professeur KAZAR Okba

Directeur du Laboratoire d'INFormatique Intelligente (LINFI)
Département d'informatique
Université de Biskra

Les jeux comme un problème de recherche

Qu'est-ce qu'on cherche?

solution, étapes pour arriver à la solution

Où on cherche?

dans une espace de recherche - ensemble d'objets (solution partiale) dans laquelle la recherche s'effectue (structure en arbre)

Comment on cherche?

dans un espace de recherche, les objets sont reliés les uns aux autres par des opérateurs qui transforment un objet en un autre

Représentation et résolution de problèmes

Pour résoudre un problème dans sa généralité, il est nécessaire de suivre une méthodologie assurant que la solution fonctionne dans tous les cas envisageables du problème. approche rigoureuse :

1. poser correctement le problème : définition du problème

2. analyser la structure du problème

3. définir une stratégie de résolution à partir de l'analyse

Arbre de recherche

- Un problème de jeu ou un problème d'intelligence artificielle peut être vu comme un problème de recherche dans un arbre :
  - Un état (configuration) initial
  - Un ensemble d'opérateurs (coups légaux ou actions légales)
    - Ces opérateurs génèrent des transitions dans l'arbre
  - Un test de terminaison
    - Indique si le jeu est terminé ou la solution est atteinte
  - Une fonction d'utilité pour les états finaux

Résolution sur des graphes d'états simples

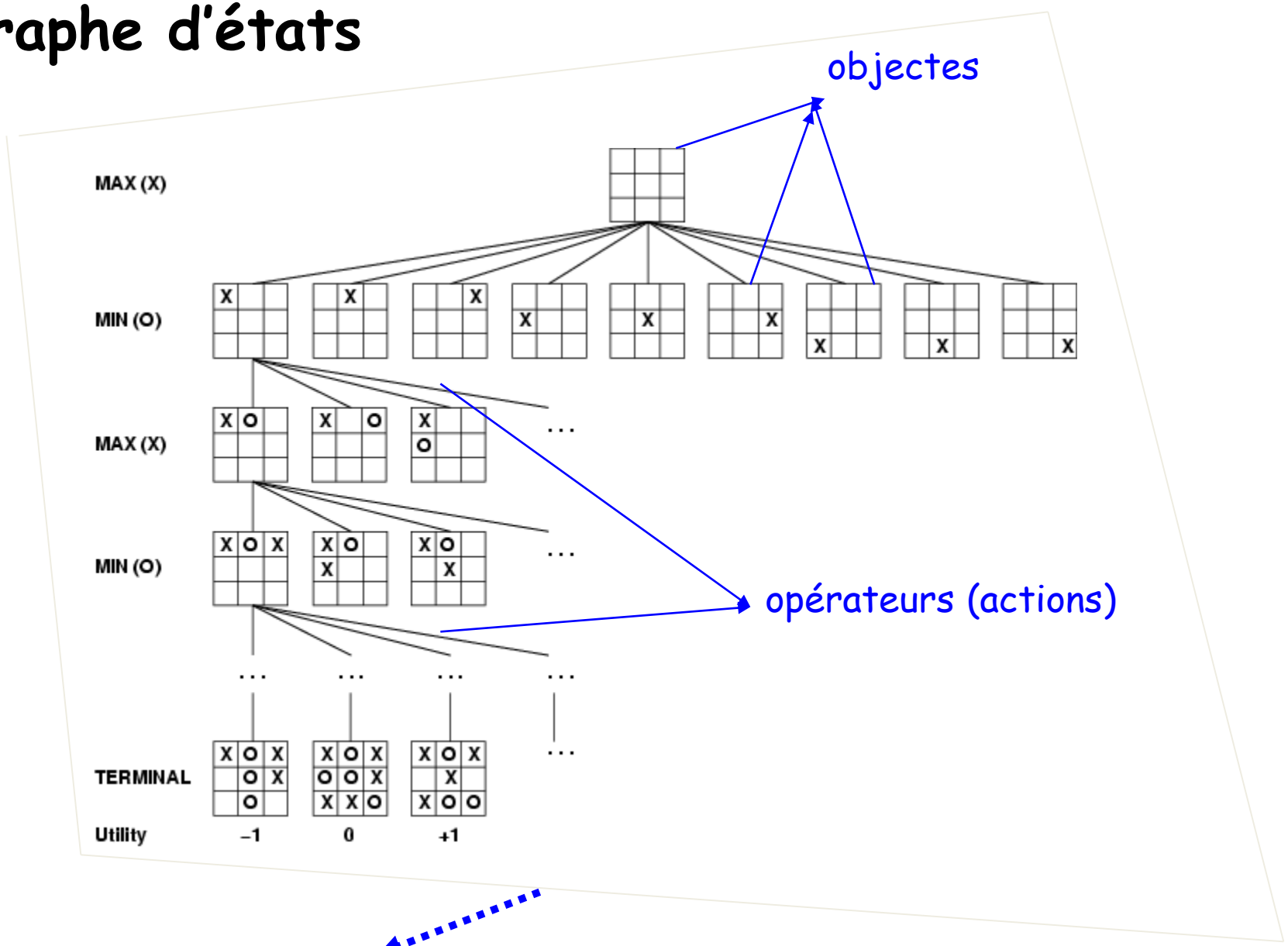
On part du principe que tous les problèmes étudiés sont composés d'un ensemble de situations ou objets que l'on peut décrire de façon univoque à l'aide d'un ensemble de variables appelés variables d'états du problèmes.

l'état d'un problème est alors l'ensemble des valeurs prises par les variables à un instant donné.

l'espace d'état d'un problème est l'ensemble de tous les états possibles de ce problème

On peut considérer que la résolution d'un problème est la découverte d'un état du problème ayant des caractéristiques données (=**la solution)**.

Graphe d'états



Graphes ET/OU

Certains problèmes peuvent être représentés par la technique de réduction de problème, c'est-à-dire que le problème peut être considéré comme **une conjonction de plusieurs sous-problèmes indépendants les uns des autres, que l'on peut résoudre séparément.**

D'autre part, il peut se présenter dans la résolution d'un problème un ensemble de **possibilités indépendantes les unes des autres (donc** exclusives) telles que la résolution d'une seule de ces possibilités suffit à résoudre le problème.

On peut représenter ces deux possibilités par des arcs différents dans le graphe représentant l'espace de recherche :

- Les arcs qui représentent différentes possibilités dans la résolution d'un problème associés au noeud dont ils découlent sont des arcs OU

- Les arcs qui représentent différents sous-problèmes composant la résolution d'un problème associés au noeud dont ils découlent sont des arcs ET.

Un tel graphe est solution d'un problème si :

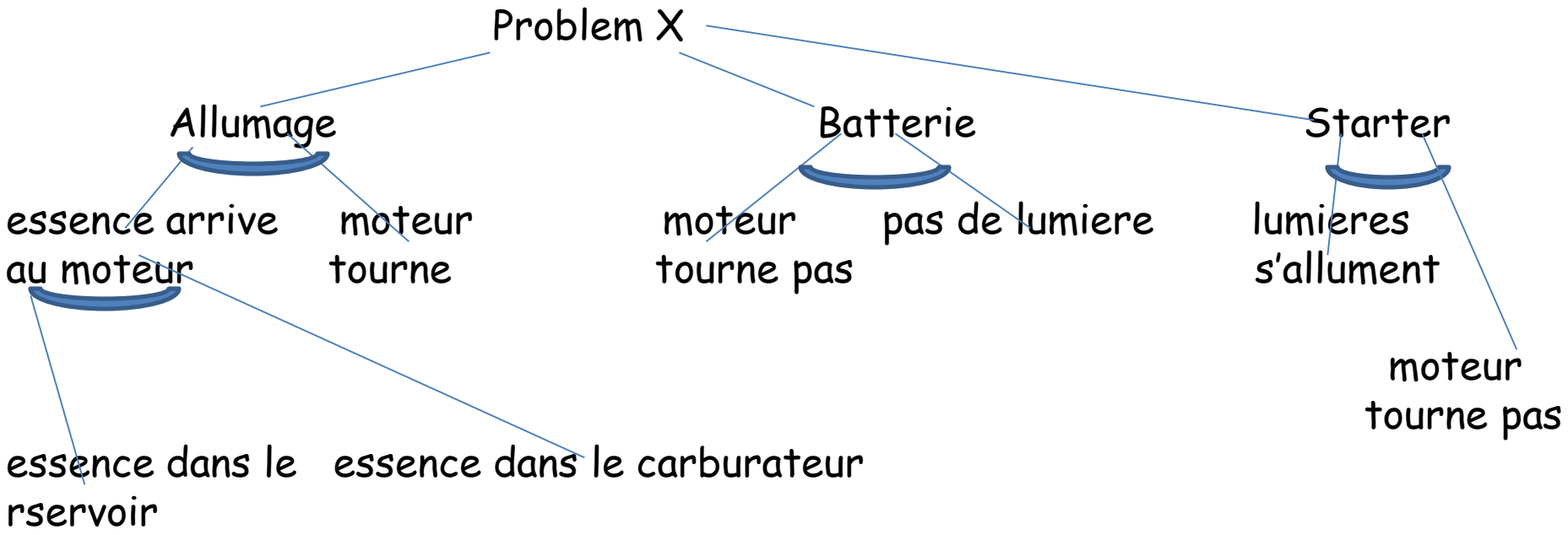
• il contient le sommet de départ (état initial du pb)

• tous les sommets terminaux (feuilles) sont des problèmes primitifs

(indécomposables) résolus.

s'il contient un arc ET il doit aussi contenir le groupe entier d'arcs ET qui sont frères de cet arc.

Exemple : diagnostic



Comment on cherche?

Application systématique des opérateurs

Vérification, après chaque transformation pour voir si l'objet qui résulte est un élément de l'ensemble des buts finaux.

Recherche Aveugle: Une méthode de recherche qui n'est pas guidée par des informations sur le domaine.

Mesure pour un espace: Un système de calcul de mesure de distance entre deux objets dans l'espace de recherche où la mesure de la valeur d'un objet donné dans cet espace.

Recherche Heuristique: Une méthode de recherche qui emploie une mesure pour guider la recherche.

La Recherche Heuristique

Heuristiques (Greek *heuriskein* = trouver, découvrir): « l'étude de méthodes et règles pour la découverte et l'invention".

- **Une heuristique est un critère ou une méthode permettant de** déterminer parmi plusieurs lignes de conduite celle qui promet d'être la plus efficace pour atteindre un but.

Ils sont des espaces de recherche trop grand pour une recherche aveugle : pour *chéquiers* il est 10^{40} chemins, échecs 10^{120}

En utilisant des heuristiques on diminue l'espace de recherche, on accélère la recherche – on doit utiliser une fonction pour garder les objects/les prochaines actions

Note

- Dans la vie réelle on utilise aussi l'heuristique:

- Exemple: Au supermarché, on choisit la queue la moins longue ou alors on choisit la queue dans laquelle les clients on le plus petit nombre d'objets dans leur panier.

- Avez-vous d'autres exemples?

Problème du voyageur de commerce

Etant donné un ensemble de villes séparées par des distances données, trouver le plus court chemin qui relie toutes les villes

Comment le faire?

1. Créer une représentation pour objets et opérateurs

2. Définir une mesure pour espace de recherche.

3. Créer une méthode efficace de comparaison ou d' évaluation d'objets en phase avec la mesure.

4. Créer une méthode efficace pour la sélection du nouvel objet à considérer dans l'espace

Les missionnaires et les cannibales: Le problème

Trois missionnaires et trois cannibales se trouvent sur la même rive d'une rivière. Ils voudraient tous se rendre sur l'autre rive.

Cependant, si le nombre de cannibales est supérieur à celui des missionnaires, alors les cannibales mangeront les missionnaires.

Il faut donc que le nombre de missionnaires présents sur l'une ou l'autre des rives soit toujours supérieur à celui des cannibales. Le seul bateau disponible ne peut pas supporter le poids de plus de deux personnes. Comment est-ce que tout le monde peuvent traverser la rivière sans que les missionnairesrisquent être mangé?

Première étape de résolution : le codage du problème. Plusieurs solutions s'avèrent possibles :

• Un triplet (X,Y,P) où X est le nombre de missionnaires sur la rive gauche, Y celui de cannibales, et P la position du bateau (G ou D).

• une liste des présents sur chaque rive (M, C ou B).

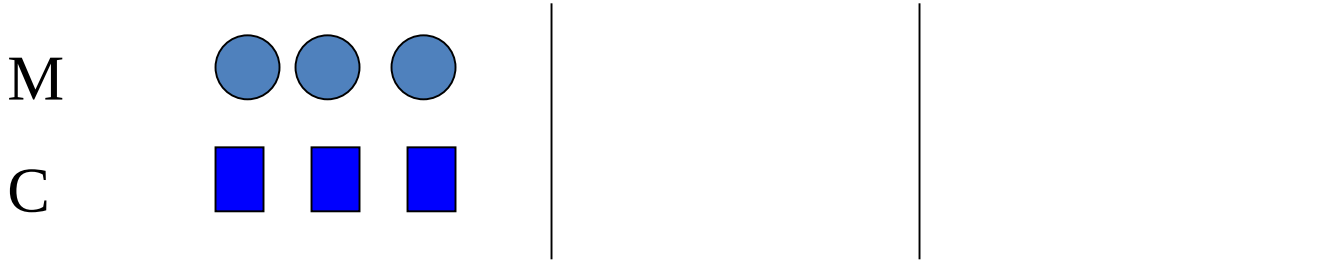
Dans le premier cas quels sont les codages des états initiaux et finaux et les contraintes portant sur les données ?

1. $0 \leq X \leq 3$ et $0 \leq Y \leq 3$, et P=D ou G
   état initial = (3,3,G)
   état final = (0,0,D)

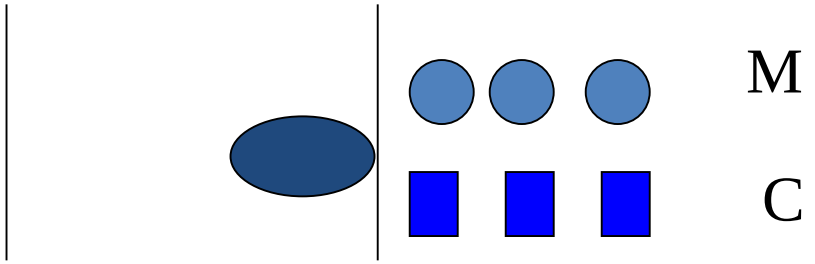
2. état initial rg=(M,M,M,C,C,C,B) rd=()
   état final rg=() rd=(M,M,M,C,C,C,B)

Représentation du problème

Configuration initiale:



- ## Configuration finale



Cette représentation n'est pas appropriée pour un ordinateur:
les règles et les contraintes ne sont pas formulées.

Etape 1 Représentation du problème

Configuration initiale

MMMCCCB|

Configuration finale

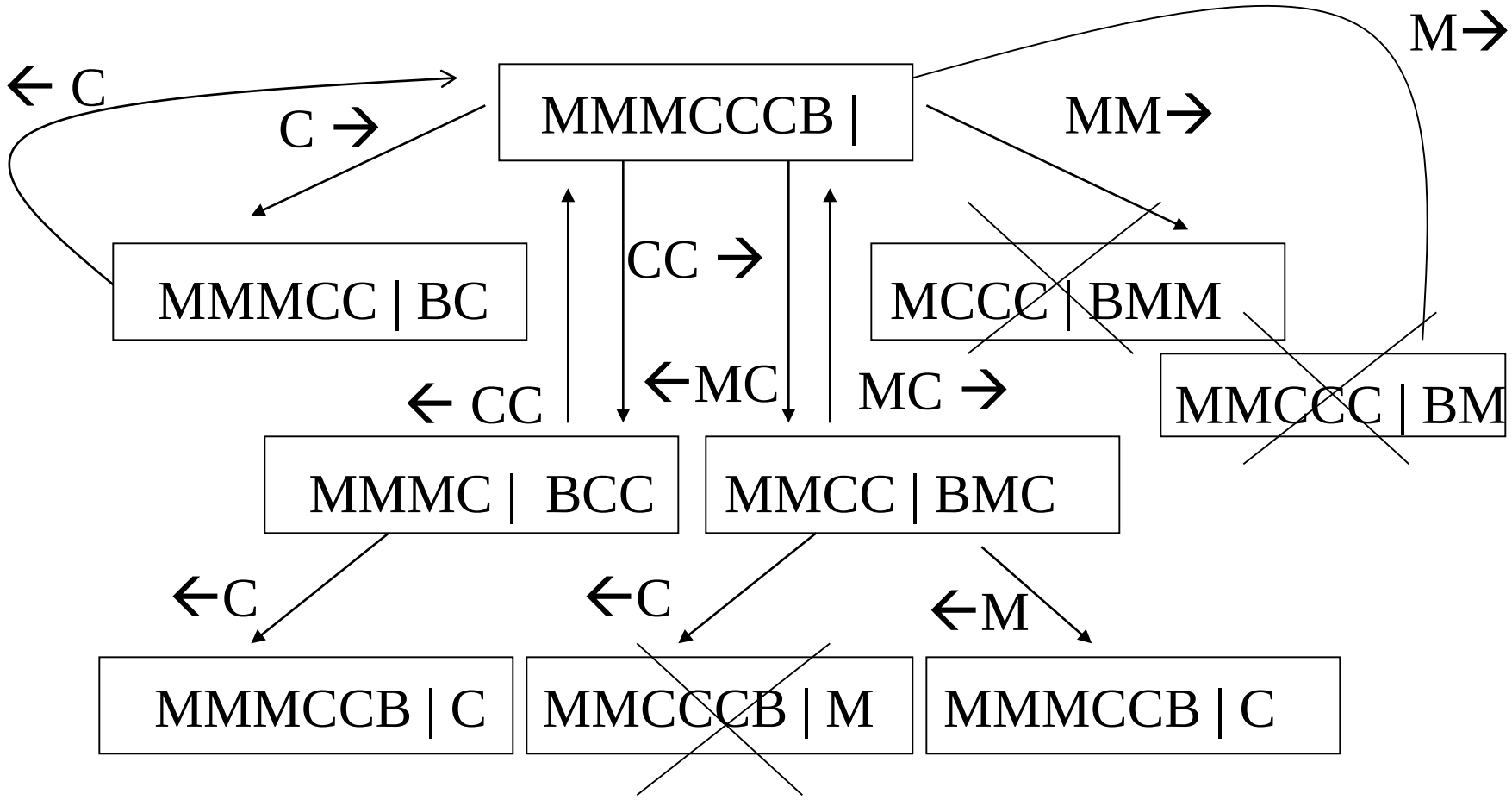
|MMMCCCB

Déplacement légaux

Contraintes

- Les cannibales ne doivent pas être plus nombreux que les missionnaires sur les deux rives
- Le bateau ne peut pas supporter plus de deux personnes.

On continue à étendre l’ espace de recherche jusqu’ à l’arrivée d’une Configuration finale



Le jeu

M&C

Une solution pour le problème

- MMMCCCB |
- MMMC | BCC
- MMMCCB | C
- MMM | BCC
- MC | BMMCC
- MMCCB | MC
- CC | BMMMC
- CCCB | MMM

- C | BCCMMM
- CCB | CMMM
- | BCCCMMM
- Le développement explicite de l' espace de recherche en entier n'est pas une solution pratique! L' espace de recherche doit être contenu à ses parties significatives

Problème 2: Le Loup, le mouton et le chou

C'est Karim, accompagné d'un loup, d'un mouton et d'un chou qui doit traverser une rivière pour rentrer chez lui.

Malheureusement, l'en a qu'une petite barque qui ne lui permet que de transporter un seul objet ou animal à la fois.

Ainsi, à chaque fois, y doit en laisser deux sur la rive sans faire gaffe à eux, le temps de traverser. Bien sûr, le loup mange le mouton et le mouton mange le chou.

Étape 1 : Représentation du problème

Configuration initiale ....|LMCB

Configuration finale LMCB|....

Opérateurs

une petite barque qui ne lui permet que de transporter un seul objet ou animal à la fois

bien sûr, le loup mange le mouton et le mouton mange le chou.

Étapes 2 et 3

Vérifier si le conditions sont satisfaites

Étapes 4

Faire juste des transportation légales

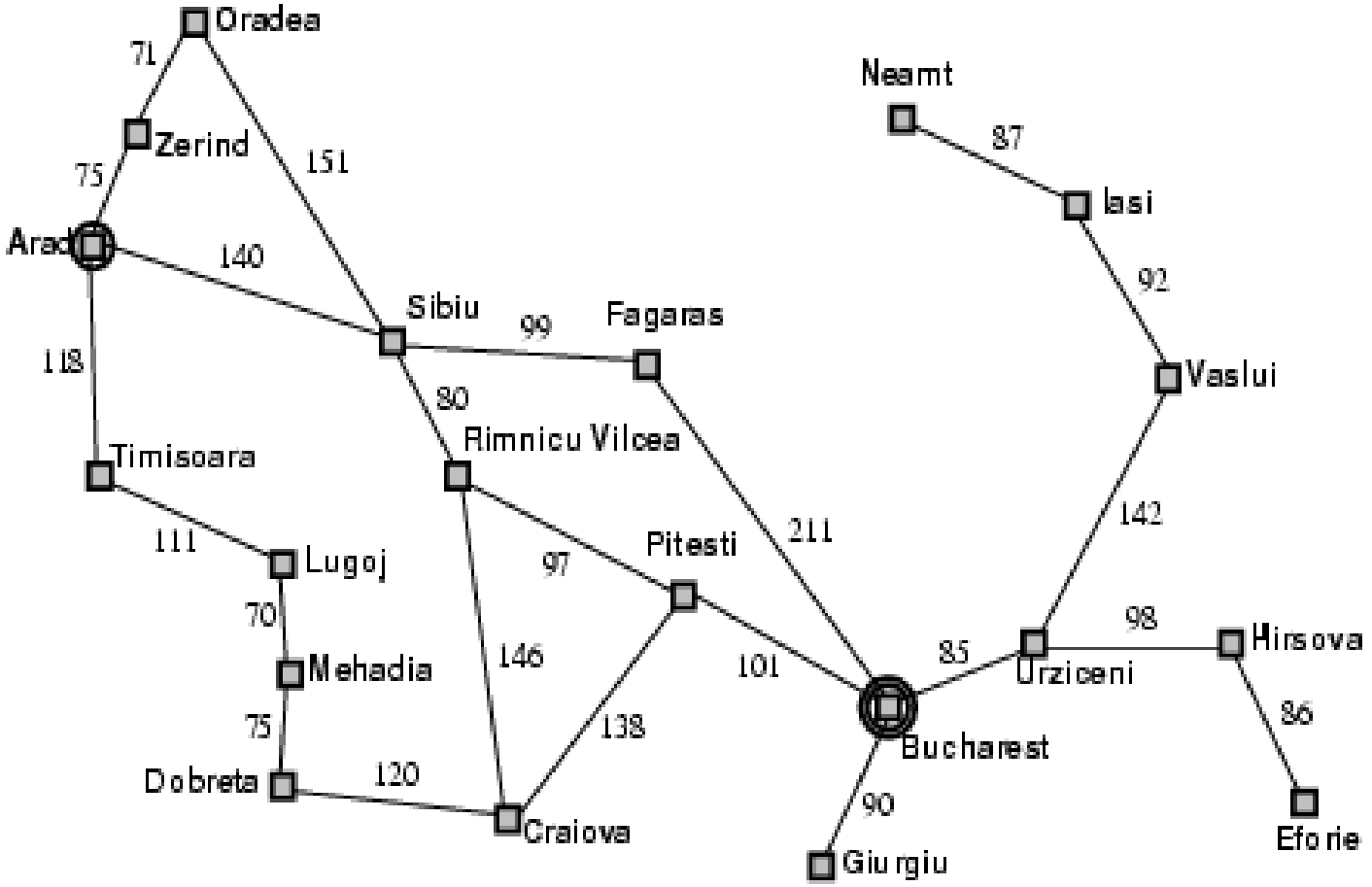
Trouver la solution!!!

LMC

Problème de voyage

En vacance en Roumanie

Ville de départ: Arad Ville d' arrive: Bucarest



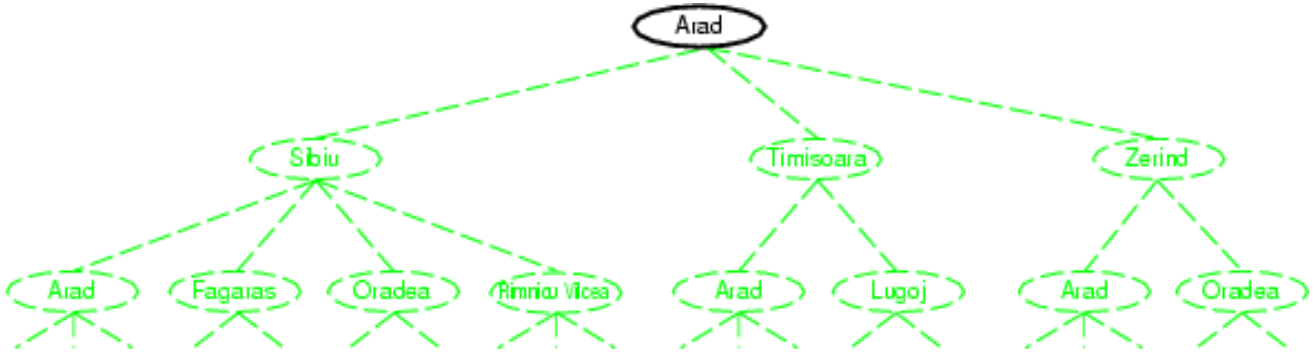
Étape 1

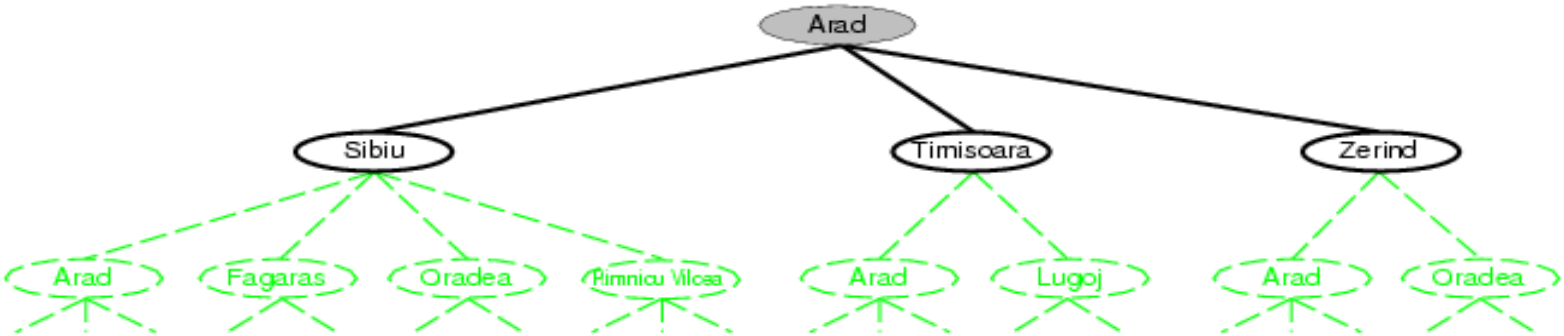
Objects: les villes

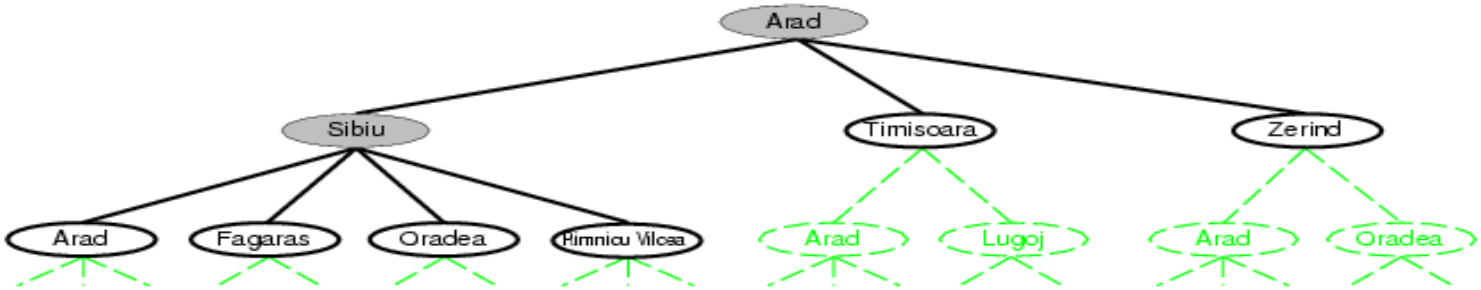
Actions: conduire entre les villes

La solution: une suite des villes - e.g., Arad, Sibiu, Fagaras, Bucharest

Représentation utilisant des arbres




Arad
Sibiu
Timisoara
Zerind
Arad
Fagaras
Oradea
Pimnicu Vilcea
Arad
Lugoj
Arad
Oradea

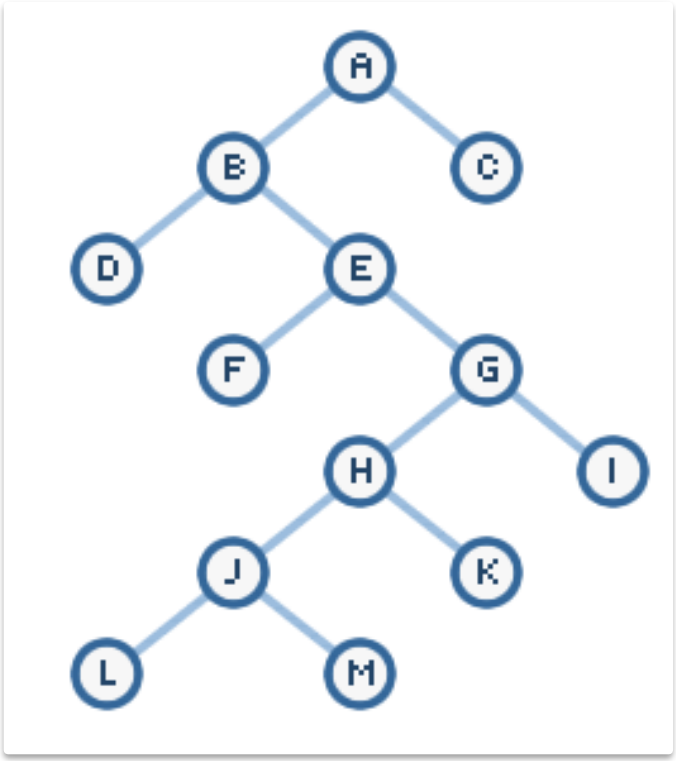

Arad
Sibiu
Timisoara
Zerind
Arad
Fagaras
Oradea
Rimnicu Vilcea
Arad
Lugoj
Arad
Oradea

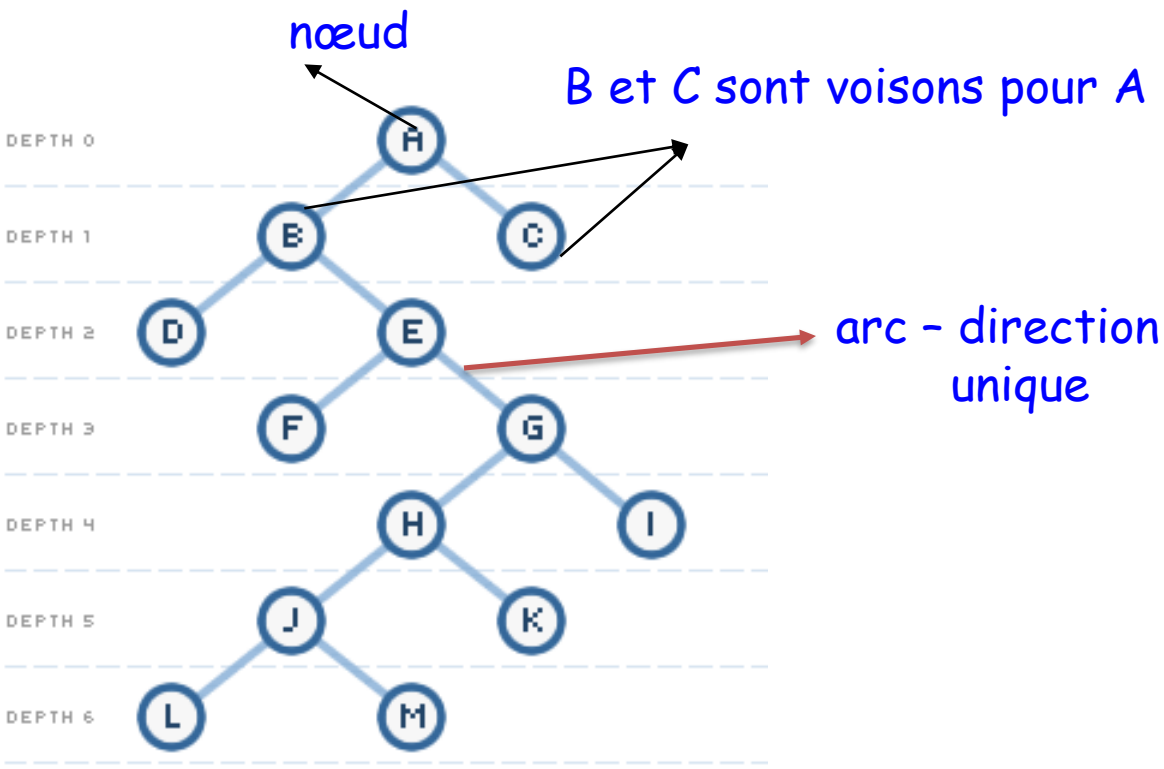
Avec les techniques de recherche

On donne aux ordinateurs une intelligence, prendre une décision -> la recherche est une grande partie de l'intelligence artificielle

Représentation pour la recherche

Arbres de recherche



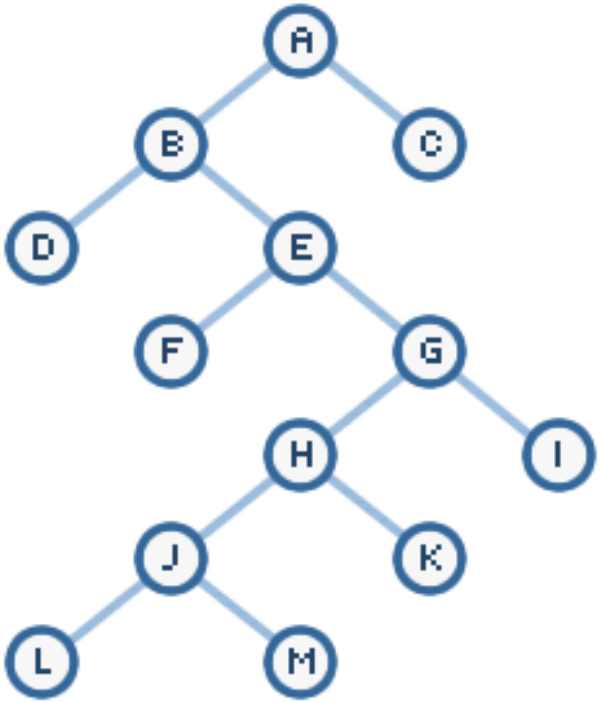

nœud
B et C sont voisons pour A
arc – direction unique
DEPTH 0
DEPTH 1
DEPTH 2
DEPTH 3
DEPTH 4
DEPTH 5
DEPTH 6
A
B
C
D
E
F
G
H
I
J
K
L
M

Recherche Depth-First (RDF)

1. On prendre un nœud on vérifie les voisins prenant le premier

2. On vérifie si le nœud actuel est la solution
   a. si oui on s'arrête
   b. si non on fait le voisin du nœud actuel le nœud actuel et on continue avec 1.

Trouver la route A->F



Étape 0

On commence avec le nœud source A

On utilise 2 listes

Liste Ouverte: les actions (nœud) courantes

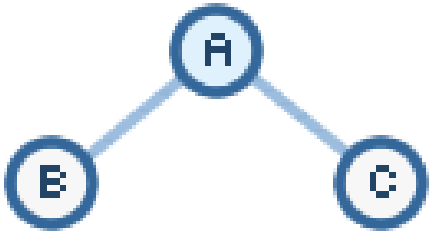
Liste Ferme: les actions (nœud) passe

Liste Ouverte: A

Liste Ferme: <>

Étape 1

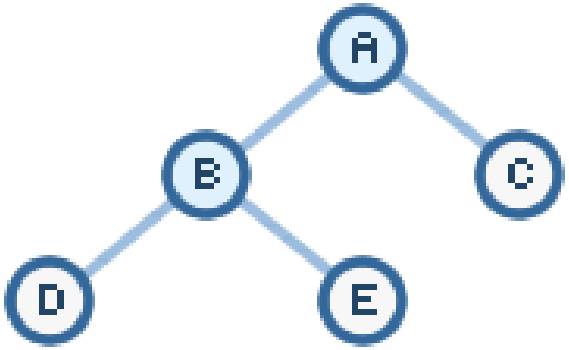
1. Liste Ouverte: *A*
   Liste Ferme: <>
2. Est-ce *A* la solution?
3. Étendre *A*



Liste Ouverte: B, C
Liste Ferme: A

Étape 2

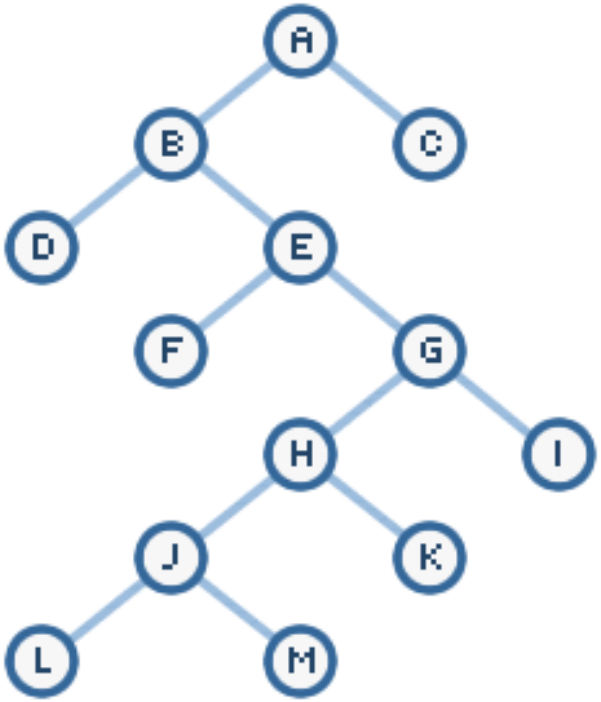
1. Liste Ouverte: B, C
   Liste Ferme: A
2. Est-ce B la solution?
3. Étendre B



Liste Ouverte: D, E, C
Liste Ferme: A,B

Étape 3

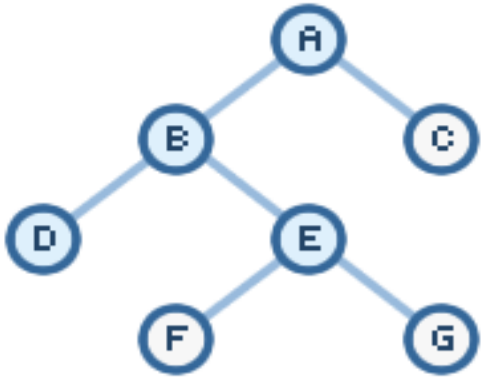
1. Liste Ouverte: D, E, C
   Liste Ferme: A,B
2. Est-ce D la solution?
3. Étendre D



Liste Ouverte: E, C
Liste Ferme: A,B,D

Étape 4

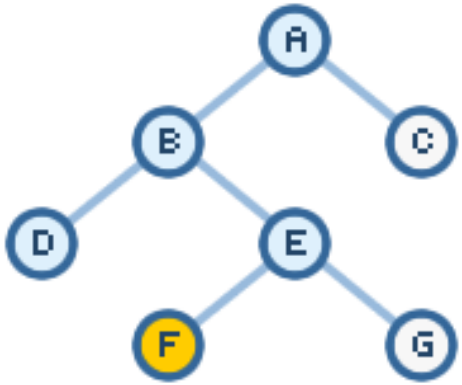
1. Liste Ouverte: E, C
   Liste Ferme: A,B,D
2. Est-ce E la solution?
3. Étendre E



Liste Ouverte: F,G, C
Liste Ferme: A,B,D,E

Étape 5

1. Liste Ouverte: F,G, C
   Liste Ferme: A,B,D,E
2. Est-ce F la solution?
   Oui F



Liste Ouverte: G, C
Liste Ferme: A,B,D,E,F

Recherche breadth-first (RBF)

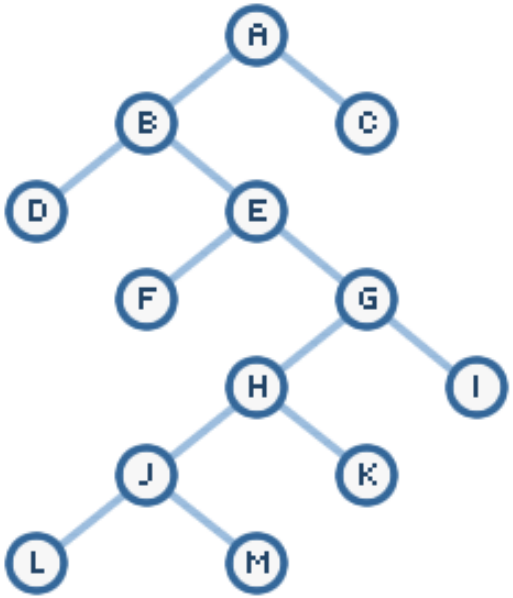
La seule différence est le placement des nœuds qui sont étendus

RDF – on met les nœuds au début de la Liste ouverte (une représentation pile)

BDF – on met les nœuds à la fin de la Liste ouverte (une représentation file)

Recherche breadth-first (RBF)

- Trouver la route A->E



Étape 0

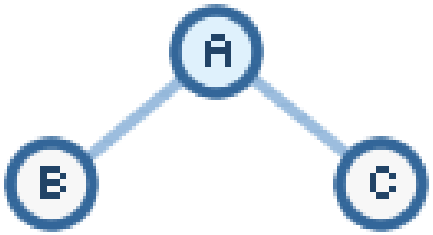
On commence avec le nœud source A

Liste Ouverte: A

Liste Ferme: <>

Étape 1

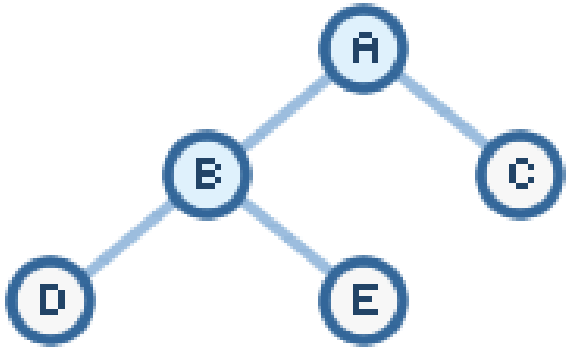
1. Liste Ouverte: *A*
   Liste Ferme: <>
2. Est-ce *A* la solution?
3. Étendre *A*



Liste Ouverte: B, C
Liste Ferme: A

Étape 2

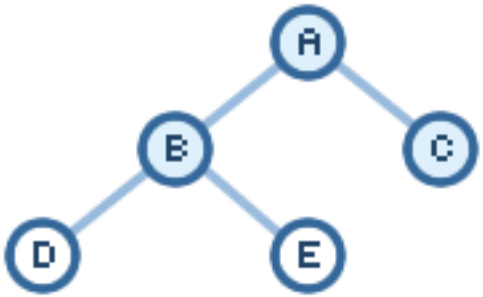
1. Liste Ouverte: B, C
   Liste Ferme: A
2. Est-ce B la solution?
3. Étendre B



Liste Ouverte: C, D, E
Liste Ferme: A,B

Étape 3

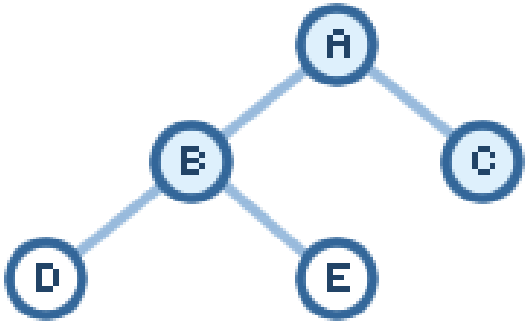
1. Liste Ouverte: C, D, E
   Liste Ferme: A,B
2. Est-ce C la solution?
3. Étendre C



Liste Ouverte: D, E
Liste Ferme: A,B, C

Étape 4

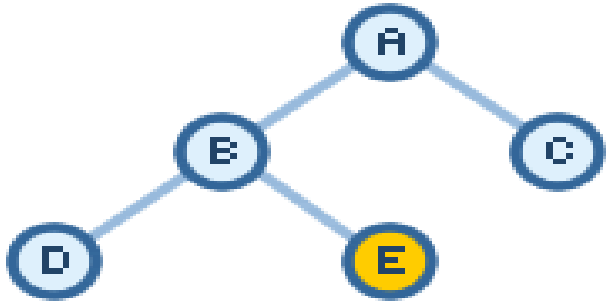
1. Liste Ouverte: D, E
   Liste Ferme: A,B,C
2. Est-ce D la solution?
3. Étendre D



Liste Ouverte: E
Liste Ferme: A,B,C,D

Étape 5

1. Liste Ouverte: E
   Liste Ferme: A,B,C,D
2. Est-ce E la solution?
   Oui E



Liste Ouverte: G, C
Liste Ferme: A,B,D,E

Quelle recherche? quand?

- **RDF:** Préférée pour chercher dans un espace de recherche structuré en un arbre fini avec des nœuds finaux dans les feuilles de l'arbre.

- **RBF:** Préférée lorsque le "branching factor" est petit, les opérateurs ont une application coûteuse et les nœuds finaux sont attendus a une profondeur raisonnable.

- Place aux questions