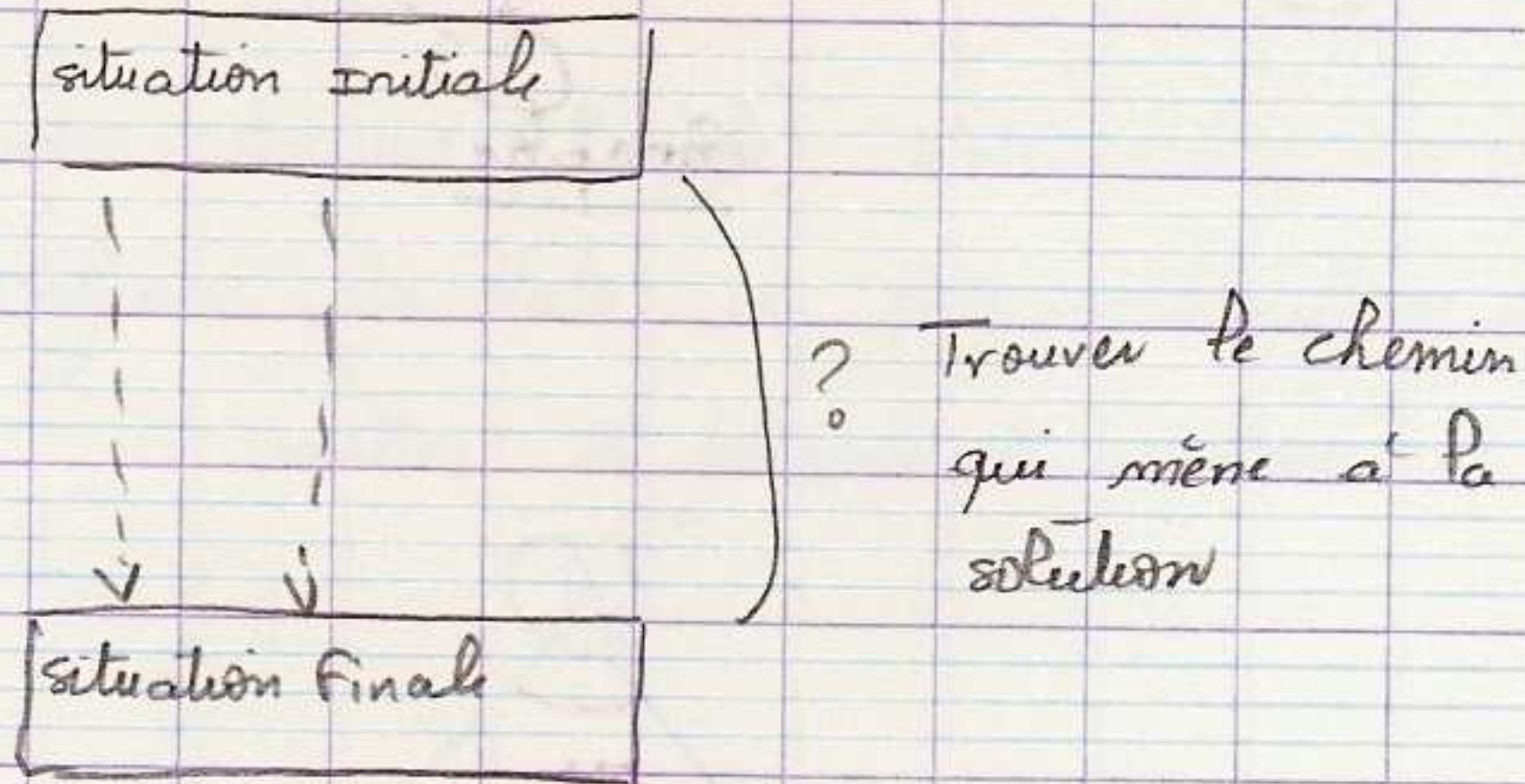


Chapitre 2:

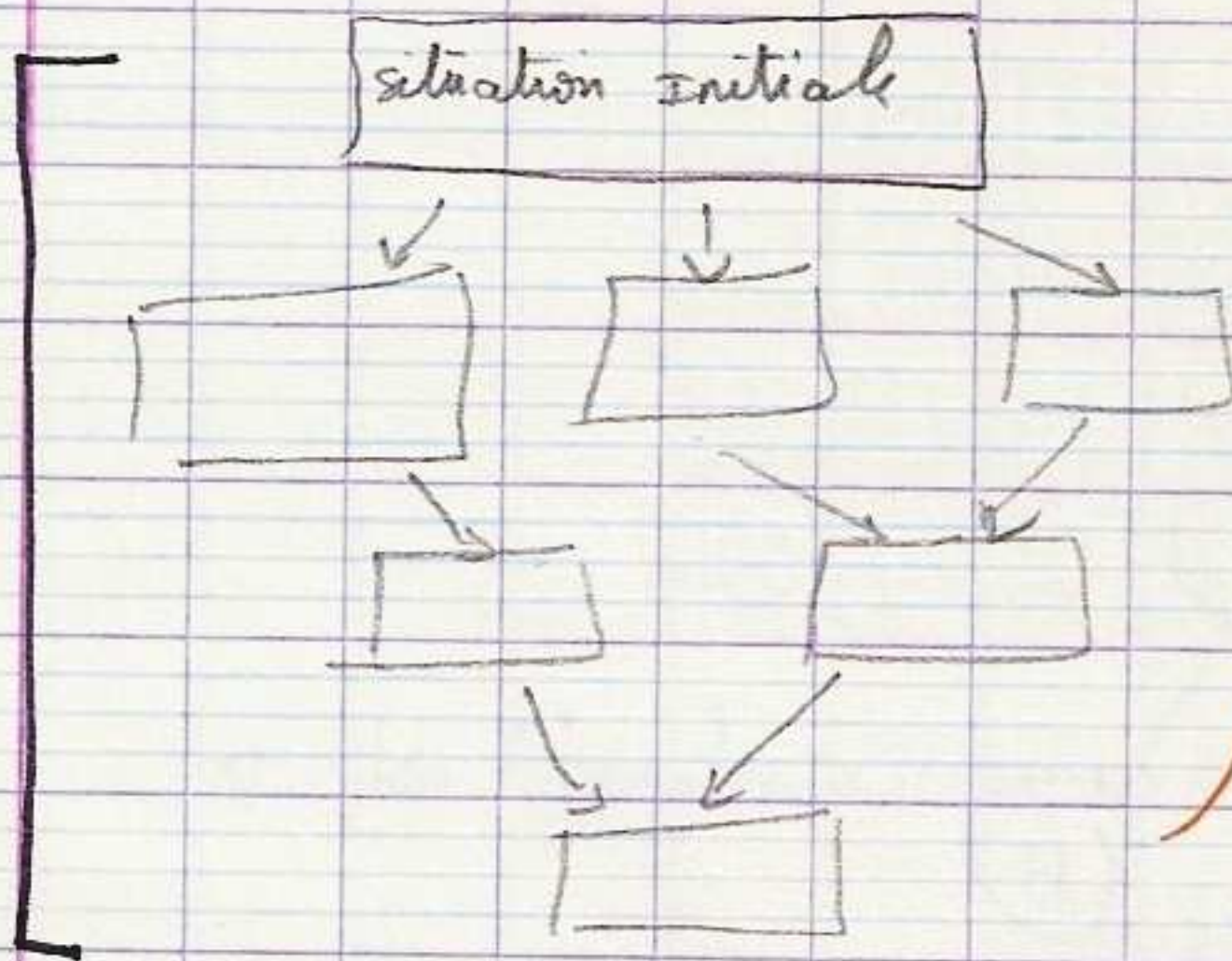
Resolution de Probleme en IA par la recherche

I] Introduction:



Résoudre un prob, c'est chercher un chemin qui permet d'aller d'une situation initiale vers une situation finale.

un Graphe
de
sous
probs



Decomposition de prob en sous
probs elementaires

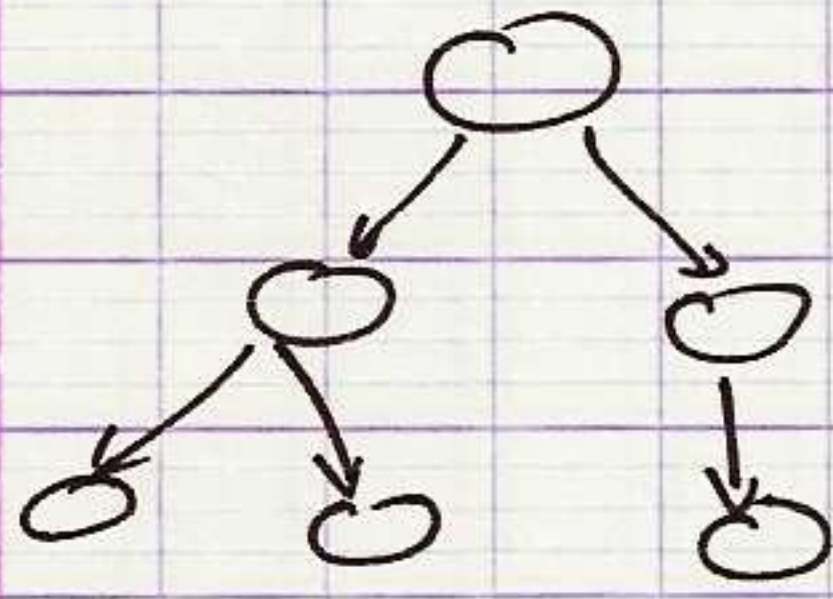
Pour résoudre le prob, il est intéressant de le décomposer en sous probs puis de les décomposer ceci en autres sous probs ... jusqu'à avoir que des probs dont les solutions est considéré comme immédiate (élémentaire)

L'ensemble des décompositions possibles peut être représenté par un graphe de sous probs.

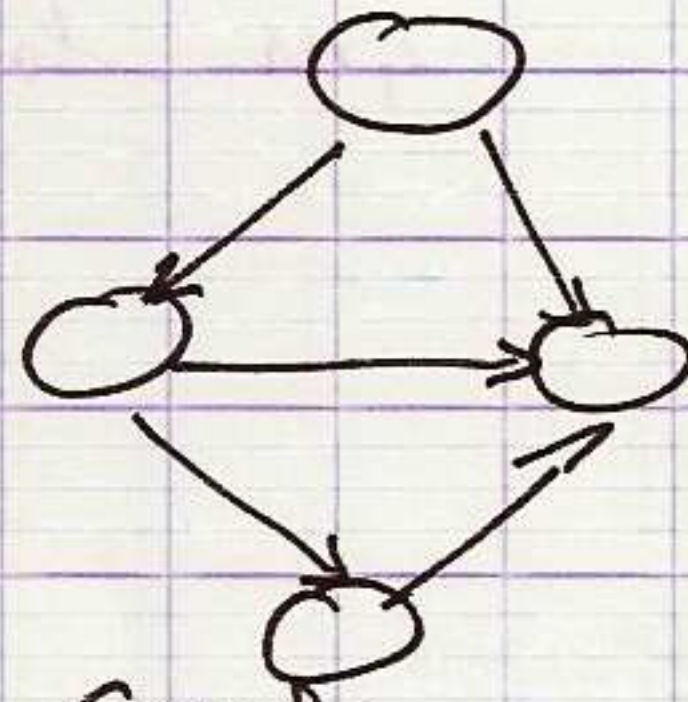
Résoudre un prob c'est de rechercher un certain sous graphe du graphe des sous probs.

→ Un graphe est représenté par un ensemble de nœuds.

→ graphe sans cycle est noté un arbre.

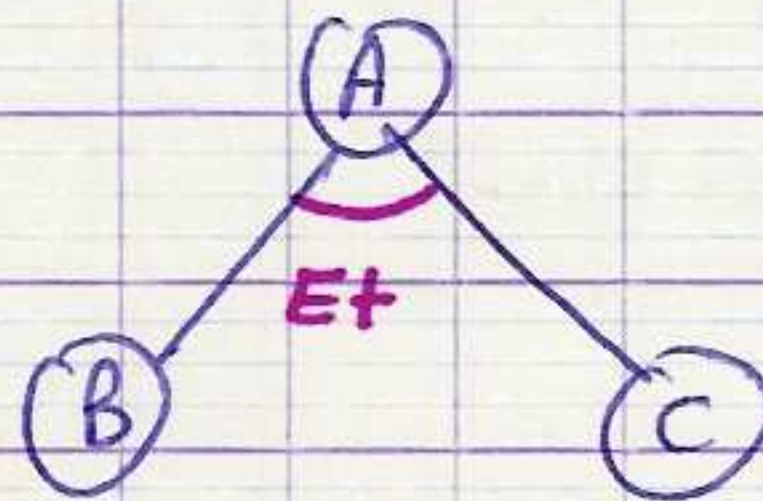
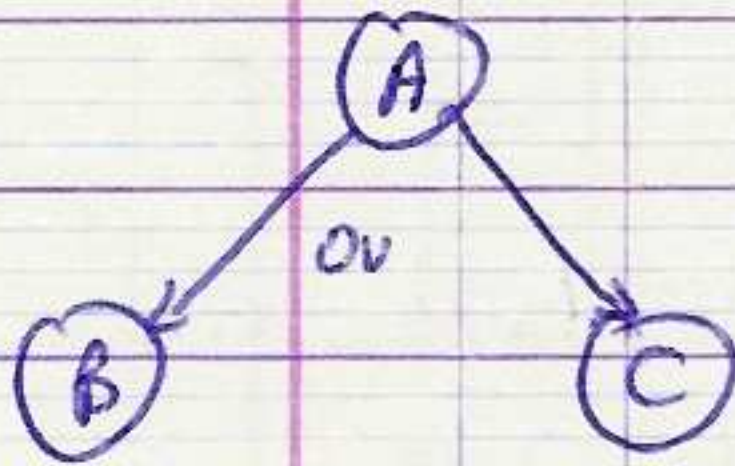


Arbre

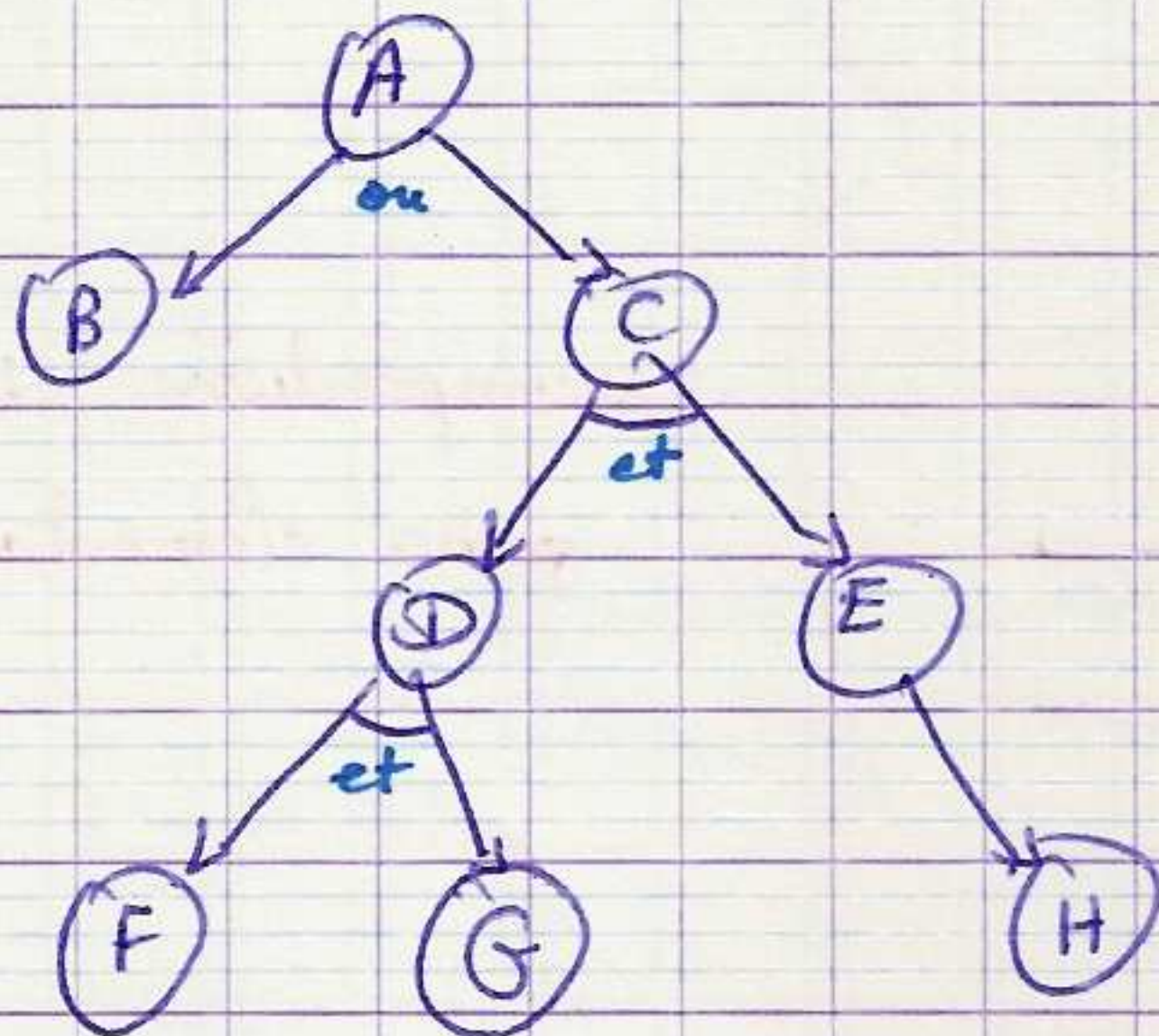


Graphe

• Arbre ET / OU



exemple:



II] Représentation d'un prob:

Le monde réel excessivement complexe



L'espace d'états doit être une abstraction de réalité

(modèle)

Pour se faire :

- Définir les états du prob.
- Définir l'objectif à atteindre
- Définir les operateurs (les outils) qui vont permettre de passer d'un état à un autre.

1. Exemple :

jeu de taquin (Puzzle) 4x4.

1) Définir les états du prob ⇒ situation initiale (état initial).

- mettre 15 jetons de nombres et laisser le case vide.

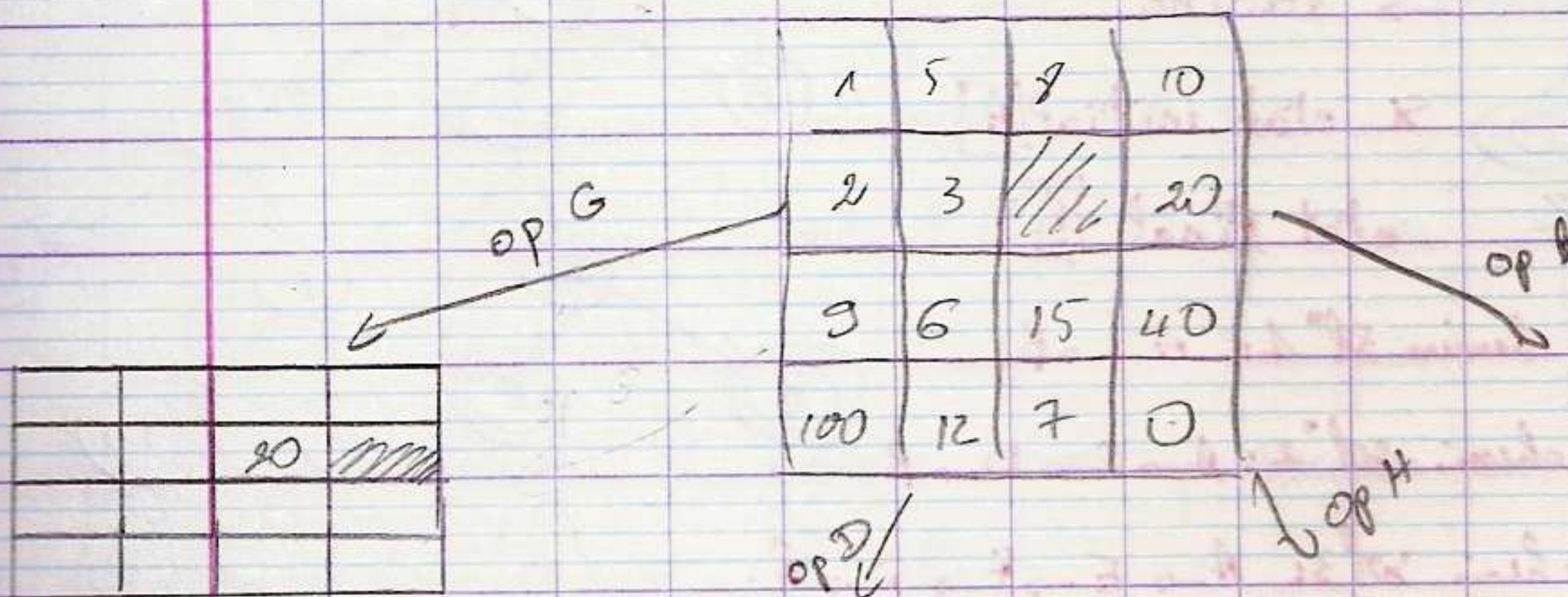
1	5	8	10
2	///	3	20
9	6	15	40
100	12	7	0

2) Définir l'objectif final ⇒ état final

9	100	8	10
2	40	20	///
1	7	5	3
12	10	6	15

- 3) les Operateurs
- 1) op G = déplacement à gauche de la case vide
 - 2) op D = " " droite " "
 - 3) op H = " " en haut " "
 - 4) op B = " " en bas " "

4) → Test But (foncⁿ qui permet de vérifier si l'état est un état final)



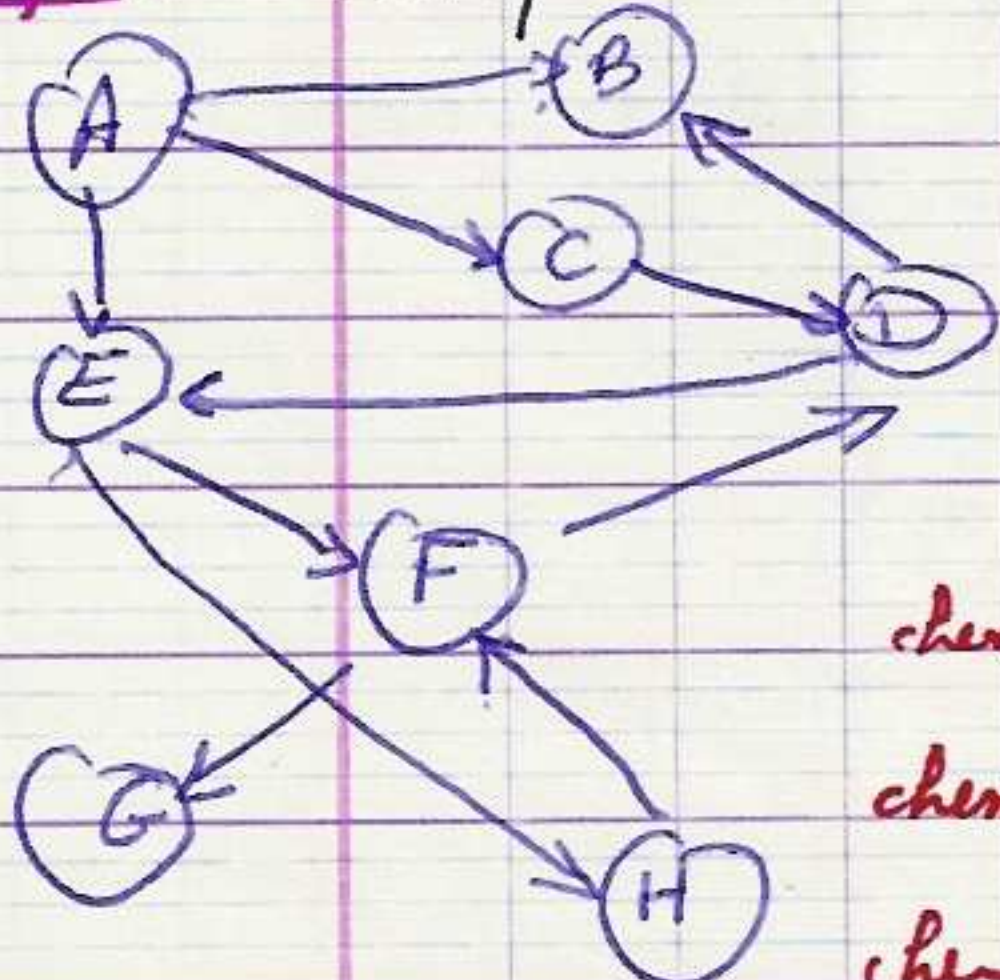
III] Types de prob:

1. Deterministe accessible: prob a etat unique.
 - etat exacte connue.
 - effet des actions connu.
2. Deterministe inaccessible: prob a etat multiple.
 - * un ensemble parmi plusieurs ensembles d'etats
 - * effet des actions connu.
3. Non Deterministe inaccessible: perception limitée.
 - Besoin de percevoir durant l'execution.
 - mélange entre recherche et execution.
 - effet conditionnel des actions.
 - la sol. a une structure d'arbre.
4. Espace d'etats inconnus: prob d'exploration en ligne (rechercher)
 - △ Besoin d'expérimenter pour trouver la sol.
 - △ l'execution révéle les états.
 - △ Besoin d'algorithmes complexes.

IV] Formulation d'un prob a etat unique:

- Etat Initial.
 - Etat final: sol.
 - les Opérateurs
 - Test au but.
 - Cost chemin: permet de déterminer quel est le meilleur chemin qui mène à la sol. (si plusieurs \exists).
- L'app des opérateurs sur les états en commençant par l'état initial conduit à la construction d'un arbre arborescence

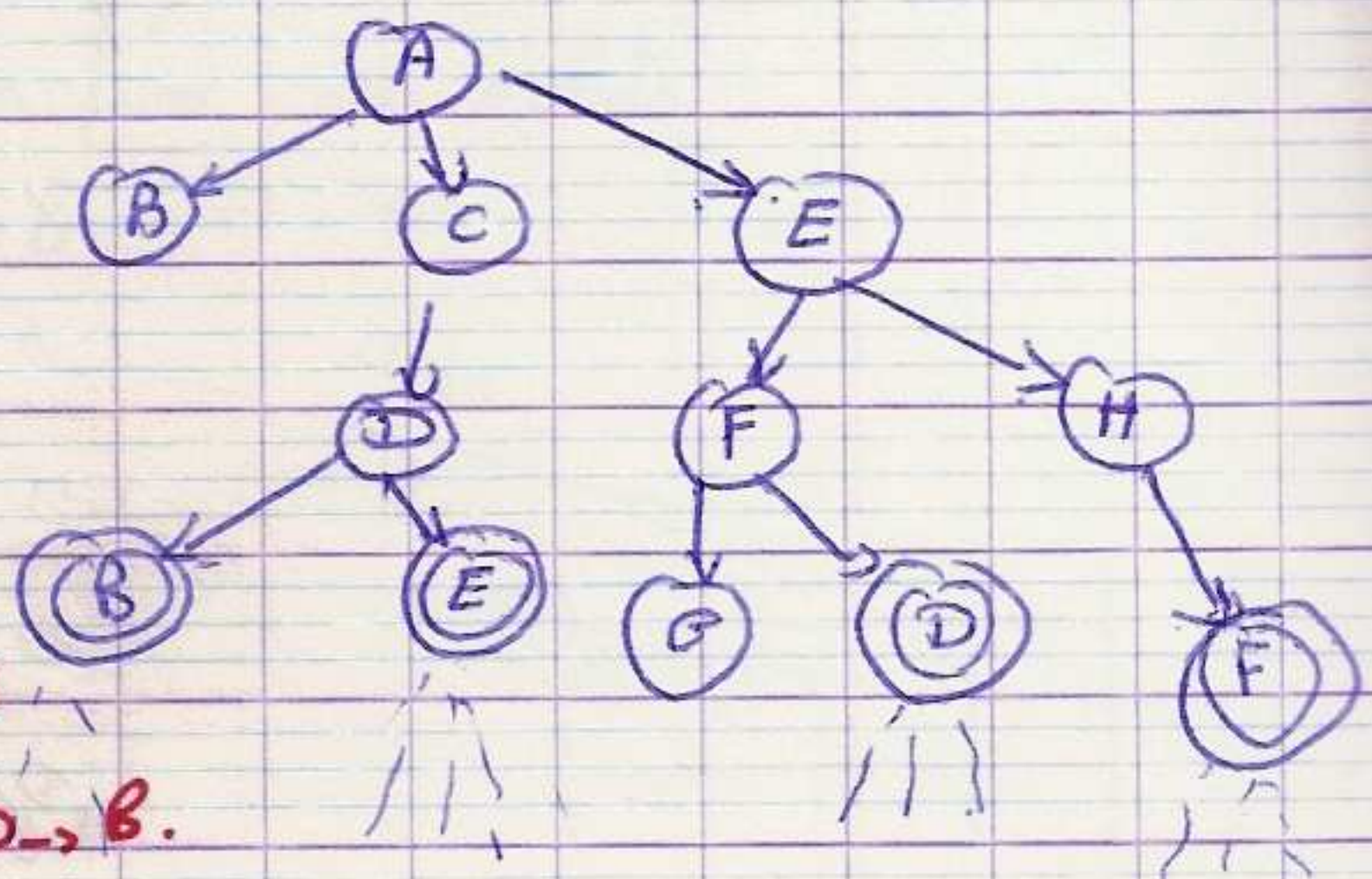
exple: Graphique



→ racine
 si etat initial [A]
 etat final [B]

- chemin sol 1: A → B
- chemin sol 2: A → C → D → B
- chemin sol 3: A → E → F → D → B.

Arbre



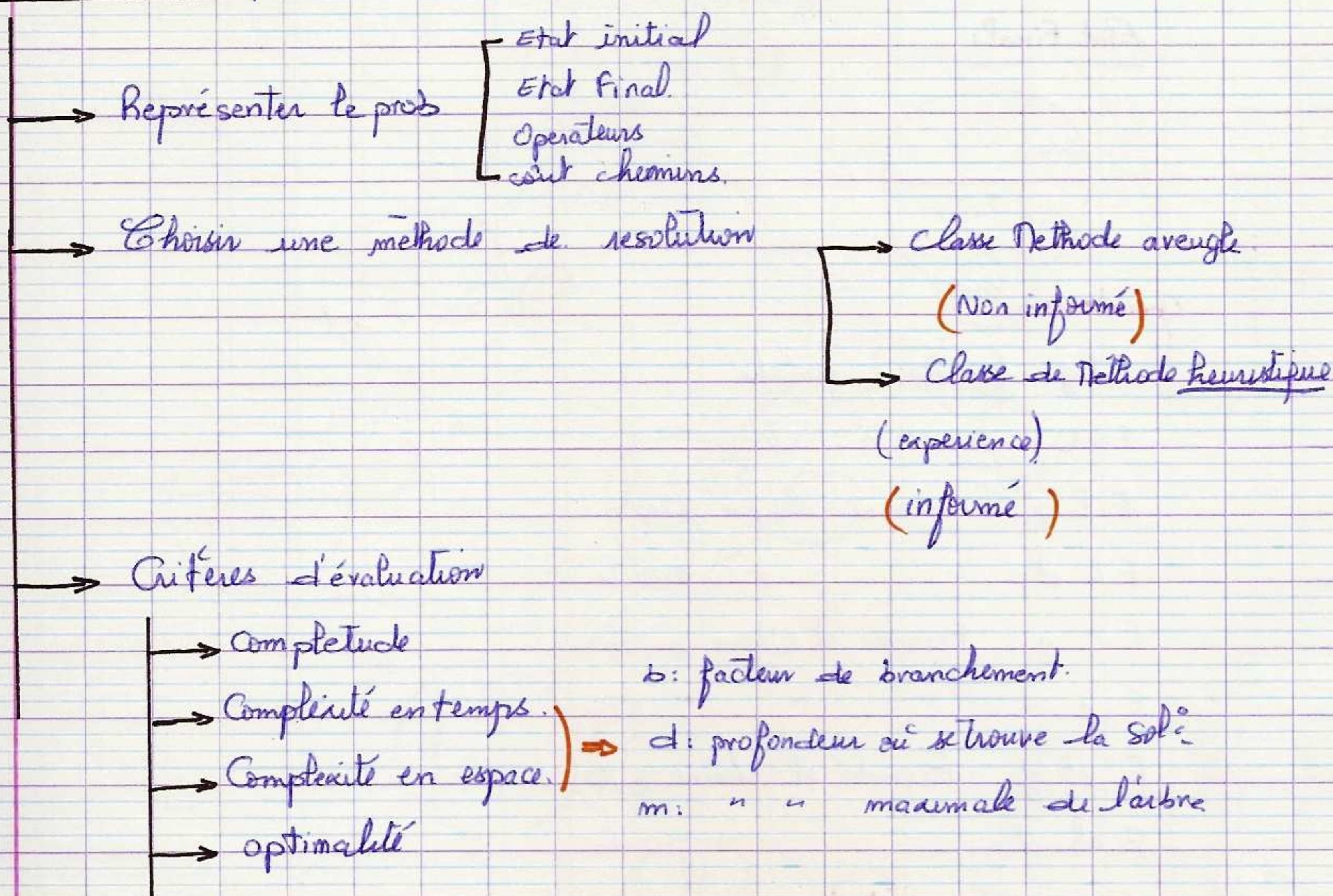
* si état initial [E]

état final [B]

sol: $E \rightarrow F \rightarrow D \rightarrow B$

3] Critères d'évaluation:

Pour résoudre un prob:



prob du fermier:

4 Acteurs (le fermier (F), le loup (L), la chèvre (CH), le chou (C)) se trouvent sur l'arrivé gauche d'une rivière.

prob: Comment faire pour passer les 4 acteurs à l'arrivé droite.

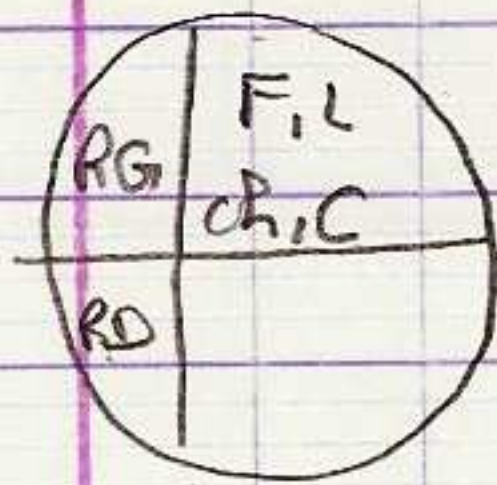
On considère le fermier dispose d'un bateau, qui peut transporter le fermier seul ou avec 1 des 3 acteurs restant de G à D ou de D à G.

- le loup peut manger la chèvre sans présence du fermier.

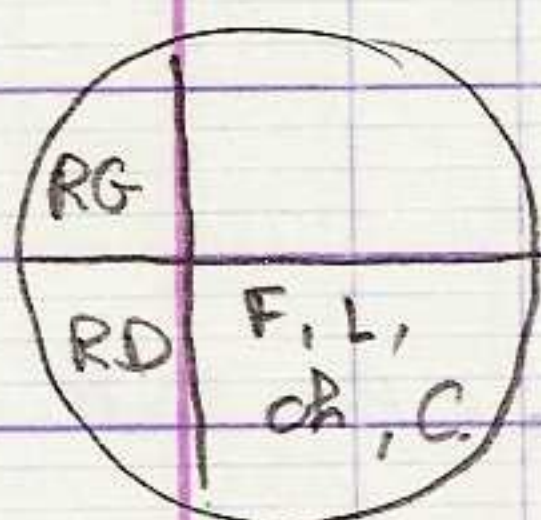
- la chèvre peut " " le chou " " " "

Correction:

Etat Initial:



Etat Final:



Operateurs:

F a g. Fermiers à gauche.

F a d. Fermiers à droite.

F L g. Fermiers et loup à gauche.

F L d. " " " droite.

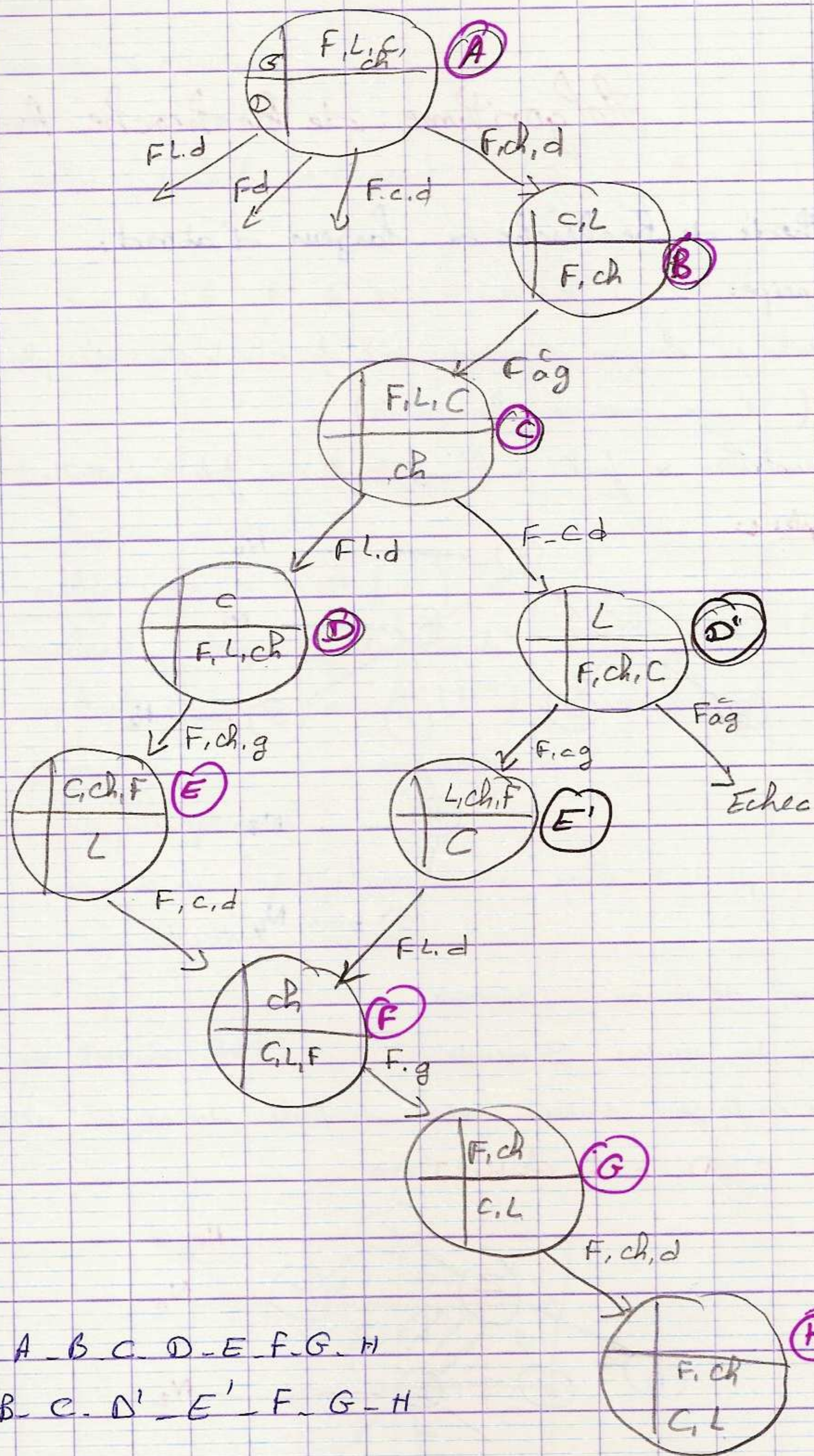
F c g. Fermiers et Chou à G.

F c d. " " " D.

F ch g. Fermiers et chevre à G.

F ch d. " " " D.

diagramme d'état:



le chemin Sol: A - B - C - D - E - F - G - H

autres: A - B - C - D' - E' - F - G - H