

# Les systèmes experts

## les concepts de base

---

### Sommaire

- [Un peu d'histoire](#)
  - [Les différentes parties d'un système expert](#)
  - [Le mécanisme d'exploitation de la base de connaissances](#)
  - [La base de connaissances](#)
  - [La base des faits du problème à résoudre](#)
  - [Des règles, des faits, des raisonnements](#)
  - [Comment enrichir et maintenir la base de connaissances](#)
  - [exercices](#)
  - **tests**
- 

### Un peu d'histoire :

Le programme DENDRAL débute vers la fin des années 60 à l'université de Stanford sous la direction de B. Buchanan avec E. Feigenbaum et le prix Nobel de chimie J. Leberg. Ce programme détermine la structure d'une molécule à partir du spectre de masse établi par un spectrogramme de masse.

Les formes que peut prendre un composé moléculaire furent apportées par Leberg, elles sont extrêmement nombreuses et susceptibles d'un enrichissement permanent au fur et à mesure que de nouveaux composés apparaissent. C'est pourquoi partant des idées de Newell et Simon et de jeunes chercheurs de Stanford (D. Waterman entre autres), E. Feigenbaum eut l'idée de séparer les raisonnements et les connaissances de Leberg du reste de la machinerie informatique : le premier système expert était né. Nous y reconnaissons l'architecture particulière des systèmes basés sur la connaissance. C'est le premier programme auquel un non informaticien, le premier "expert", ait apporté sa connaissance identifiée et valorisée en tant que telle.

Le succès de DENDRAL devait conduire, dans les années 1975-80, aux systèmes experts. Le premier et le plus connu des systèmes de cette époque est MYCIN, conçu à l'université de Stanford par Shortliffe. C'est un système expert de diagnostic médical. Il est le premier à prendre en compte des connaissances incertaines. Il sera bientôt suivi sur ce point par PROSPECTOR (Duda et Hart, Nilsson et toute une équipe du SRI). Le succès de MYCIN qui travaille aussi bien que les experts, quant les experts sont d'accord, et le succès médiatique de PROSPECTOR, devaient conduire à un certain engouement.



### Les différentes parties d'un système expert:

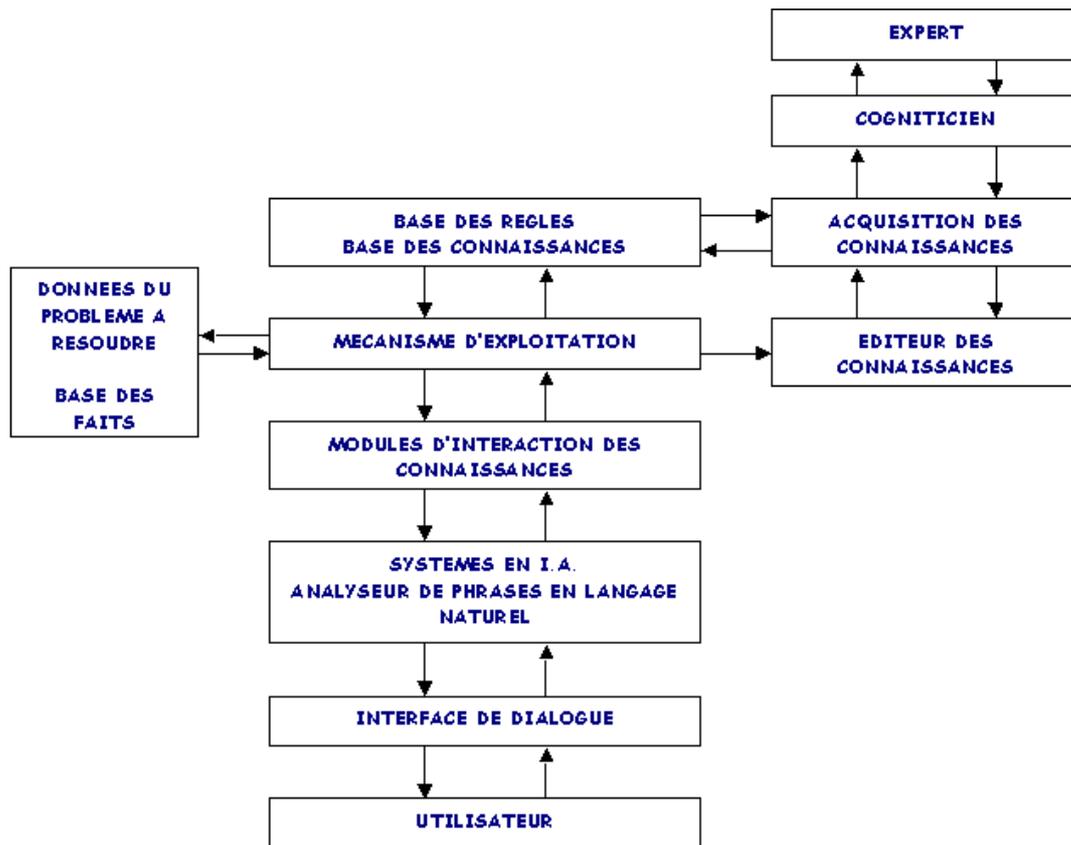
Le cœur d'un système en I.A. est composé de trois parties:

1. **Une base de connaissances** où sont stockées, sous une forme appropriée, **toutes les connaissances** permettant de résoudre le problème que l'on veut traiter **dans un domaine déterminé**.
2. **Un mécanisme d'exploitation (moteur)** qui est un ensemble de programmes susceptibles de **traduire le raisonnement humain** (heuristique, progression par avancement et retour arrière, décomposition en sous ensemble...). Ces programmes utilisent les connaissances stockées dans la base de connaissance afin de résoudre le problème que l'utilisateur lui a posé en lui donnant des hypothèses de départ.
3. **Une base des faits** du problème à résoudre, qui est en fait une mémoire de travail où seront stockées les **hypothèses de départ** où viendront s'ajouter **des faits nouveaux** au fur et à mesure que progressera le mécanisme d'exploitation.

Un système, en I.A., doit posséder deux autres parties qui sont en fait des modules d'interface avec l'extérieur :

4. **Un module d'interaction avec les utilisateurs potentiels** : cette interface de dialogue doit être la plus conviviale possible. Ce module est lié au mécanisme d'exploitation.

5. Un module d'aide à l'acquisition des connaissances qui est surtout une interface de dialogue entre la base de connaissances et les experts du domaine qui doivent alimenter cette dernière.



### ARCHITECTURE GENERALE D'UN SYSTEME EXPERT

Une base de connaissance est complètement indépendante du mécanisme d'exploitation des connaissances. L'expert peut ajouter modifier ou supprimer des connaissances sans s'occuper du mécanisme d'exploitation.



## Le mécanisme d'exploitation de la base de connaissances :

La base de connaissances va exploiter la base des faits pour rechercher la ou les solutions (sinon échec). Le système essaie de **simuler le raisonnement humain** en utilisant certains concepts :

- Les heuristiques (par opposition aux algorithmes).
- La décomposition d'un problème en sous problème.
- La progression par avancement et retour arrière.
- L'évolution vers la solution par une progression d'étapes successives différentes.

**Le raisonnement peut être certain** : aucune connaissance n'est pondérée.

**Le raisonnement peut être incertain** : à chaque connaissance est attaché un coefficient de plausibilité (ou de vraisemblance) permettant d'utiliser une logique dite floue ou de l'incertain par laquelle il existe un continuum de possibilités allant du vrai au faux (ou de zéro à un).



## La base de connaissances:

### Les types des connaissances :

Une description des objets constituant le contexte du domaine du système en I.A., et ceci sous plusieurs formes :

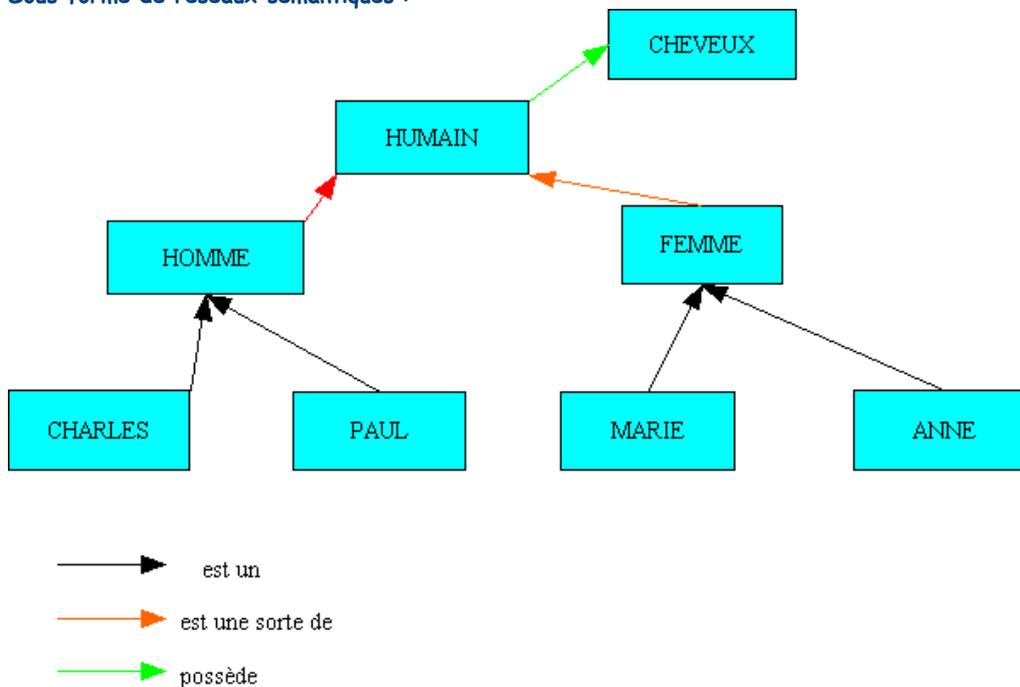
- Description-classification :  
"le canari est un oiseau".
- Description-structure :  
"le part-brise est un élément de la carrosserie d'une voiture, d'un camion, d'un car".
- Description-définition :  
"une chaise est un meuble qui a 4 pieds, 1 dossier et 1 siège".

Des connaissances opérationnelles sur les objets décrits précédemment ; elles indiquent :

- Des lois, des théorèmes :  
"la vitesse est donnée par la distance parcourue sur le temps mis pour la parcourir".
- Des règles d'expertise :  
"si un oiseau est jeune, alors c'est un canari".
- Des stratégies :  
"si golf alors consulter les sports individuels".

## La représentation des connaissances en I.A. :

- Sous forme de règles de production (ou logique des propositions) :  
SI...ALORS...  
si Jean est un homme alors Jean est mortel.
- En logique du premier ordre ou le calcul des prédicats :  
Homme(x) alors mortel(x).
- Sous forme de réseaux sémantiques :



- Sous forme d'objet, frame, schémas :  
La plupart des systèmes experts utilisent une représentation des connaissances à règles de production  
N.B. : Certains S.E. combinent plusieurs formalismes.

La représentation des connaissances est liée au sujet traité.

- La logique propositionnelle ou du premier ordre pour la démonstration automatique de théorèmes mathématiques.
- Les réseaux sémantiques pour l'analyse du langage naturel, pour la traduction automatique.
- Les règles de production pour la description d'un domaine circonscrit parfaitement connu par un expert.
- etc...

C'est dans la base de connaissance que seront stockées les connaissances ou le savoir qui nous permettront par exemple :

- de jouer aux échecs,
- de traduire du français en anglais et vice-versa,
- de reconnaître des profils d'avions, de bateaux...,
- de résoudre des problèmes mathématiques.



## La base des faits du problème à résoudre :

La base des faits, appelée aussi mémoire de travail, contient les hypothèses de départ à savoir les données propres au problème à résoudre.

- Les hypothèses mathématiques d'un problème à résoudre.
- Les éléments permettant de découvrir une maladie.
- etc...

Ces hypothèses peuvent être formalisées différemment suivant les systèmes.

### Exemple 1.1 :

système expert SPHINX.

FAITS : (fièvre 40°, boutons, amaigrissement ).

### Exemple 1.2 :

système expert EXPERT.

FAITS : (ANIMAL a des plumes),  
(ANIMAL vole)

La base des faits s'enrichira progressivement à la demande du système expert pour qu'il puisse progresser vers la solution. Par exemple, avec les faits initiaux de l'exemple 2, le système expert trouve le fait que ANIMAL est oiseau. La base des faits contiendra :

(ANIMAL a des plumes), (ANIMAL vole), (ANIMAL est oiseau).

Cette base des faits peut jouer le rôle de mémoire auxiliaire, en mémorisant tous les résultats intermédiaires. Ils permettent de conserver une trace des raisonnements effectués par le système. Ils peuvent expliquer l'origine des connaissances déduites par le système expert.



## Des faits, des règles, des raisonnements :

Nous faisons l'hypothèse selon laquelle les raisonnements sont formalisés sous forme de règles de production. Une règle se met sous la forme :

**SI** Condition\_1 et Condition\_2 et Condition\_3 ... et Condition\_n  
**ALORS** Action\_1 et Action\_2 et Action\_3 ... et Action\_p.

### Exemple 1.3:

**SI** l'animal pond des oeufs et l'animal vole **ALORS** l'animal est un oiseau.

La partie gauche est une prémisses. La partie droite est une conclusion. Une telle règle est aussi appelée inférence. La partie gauche exprime des conditions. Si toutes les conditions sont satisfaites l'autre partie devient vraie.

Ensemble des règles = la base des règles = la base des connaissances,  
Ensemble des faits = la base des faits.

Comment fonctionne en pratique un système expert ?

### Exemple 1.4 : Système expert ANIMAUX

| Base des règles "ANIMAUX" :   | Base des faits "ANIMAUX" :   |
|---|--|
| <ul style="list-style-type: none"> <li>• R1 : <b>SI</b> l'animal est oiseau et l'animal pèse moins de 10 gr<br/><b>ALORS</b> l'animal est colibri.</li> <li>• R2 : <b>SI</b> l'animal vole et l'animal pond des oeufs<br/><b>ALORS</b> l'animal est oiseau</li> </ul> | <ul style="list-style-type: none"> <li>• Animal vole</li> <li>• Poids animal est inférieur à 10 gr</li> <li>• Animal pond des oeufs</li> </ul> |

Le mécanisme d'exploitation ou moteur d'inférence prend une règle et regarde dans la base de faits si les faits stockés vont rendre la prémisses de la règle vraie.

Tirons la première règle **R1** : l'une des conditions n'est pas remplie. En effet, on ne sait pas si l'animal est oiseau. Ce fait n'est pas dans la base des faits.

Tirons la règle suivante **R2** : les conditions sont vraies. La base des faits contient les deux conditions animal vole et animal pond des oeufs. La règle est exécutée.

La base des faits s'enrichit d'un fait supplémentaire : "**l'animal est oiseau**".

| Nouvelle base des faits<br>"ANIMAUX" :  |
|---|
| <ul style="list-style-type: none"> <li>• Animal vole</li> <li>• Poids animal est inférieur à 10 gr</li> <li>• Animal pond des oeufs</li> <li>• Animal est oiseau</li> </ul> |

Tirons maintenant de nouveau la règle **R1** : les conditions sont vraies, la règle est exécutée. Un fait nouveau est ajouté à la base des faits : "**l'animal est colibri**".

### Exemple 1.5 : Système expert VILLE

| la base de règles "ville" :  | la base des faits "ville":  |
|--|---|
| <ul style="list-style-type: none"> <li>• <b>R1</b> : Si belle ville et très bon restaurants alors ville méritant le voyage.</li> <li>• <b>R2</b> : Si ville historique alors ville méritant le voyage.</li> <li>• <b>R3</b> : Si autochtones accueillants et traditions folkloriques alors ville méritant le voyage.</li> <li>• <b>R4</b> : Si monuments et végétation abondante alors belle ville.</li> <li>• <b>R5</b> : Si tradition culinaire alors bons restaurants.</li> <li>• <b>R6</b> : Si restaurants 3 étoiles alors très bons restaurants.</li> <li>• <b>R7</b> : Si restaurants 3 toques alors très bons restaurants.</li> <li>• <b>R8</b> : Si musées et ville ancienne alors ville historique.</li> <li>• <b>R9</b> : Si Provence et bord de mer alors autochtones accueillants.</li> <li>• <b>R10</b> : Si parcs verdoyants et avenues larges alors végétation abondante.</li> </ul> | <ul style="list-style-type: none"> <li>• Parcs verdoyants</li> <li>• Avenues larges</li> <li>• monuments</li> <li>• restaurants 3 toques</li> <li>• ville ancienne</li> </ul> |

### Exemple 1.6 : Système expert ABC

Soit la base de connaissances suivante :

Base de faits initiale :

{H,K}

Base des règles :

- **R1** : A --> E
- **R2** : B --> D
- **R3** : H --> A
- **R4** : E et G --> C
- **R5** : E et K --> B
- **R6** : D et E et K --> C
- **R7** : G et K et F --> A

La règle **R1** signifie SI A ALORS E  
("De A on peut déduire E")

Le démonstrateur de théorèmes se propose d'induire des faits nouveaux à partir de la base de connaissances et de la règle de **modus ponens**, c'est-à-dire de la règle : "**Si p est vrai et si p --> q alors q est vrai**".

Deux stratégies générales de démonstration sont alors possibles :

**Le chaînage avant** : on part de la base de faits initiale ; on déclenche toutes les règles dont les prémisses sont satisfaites ; on ajoute les faits ainsi obtenus ; on poursuit jusqu'à "saturation".

Dans l'exemple précédent, on obtient la chaîne de dérivation suivante :

- $H \rightarrow A$  (R3) et la base de faits  $B = \{A, H, K\}$
- $A \rightarrow E$  (R1) et  $B = \{A, E, H, K\}$
- $E$  et  $K \rightarrow B$  (R5) et  $B = \{A, B, E, H, K\}$
- $B \rightarrow D$  (R2) et  $B = \{A, B, D, E, H, K\}$
- $D$  et  $E$  et  $K \rightarrow C$  (R6) et  $B = \{A, B, C, D, E, H, K\}$

**Le chaînage arrière** : supposons que l'on cherche à démontrer si le fait D est vérifié ; c'est le but recherché. On regarde toutes les règles qui ont le but dans leurs conséquences. Chacune de ces règles est considérée : si toutes ses prémisses sont satisfaites dans la base de faits initiale, le but est atteint, sinon on enregistre les prémisses inconnues comme autant de nouveaux buts et on recommence le cycle sur chacun d'eux.

Dans l'exemple précédent :

- On considère la règle (R2) et on définit B comme nouveau but.
- On considère alors la règle (R5) et on définit E comme nouveau but (puisque K est dans la base des faits).
- On considère alors les règles (R3) et (R7). (R3) permet de conclure.

### Exemple 1.7 :

Dans le domaine de la botanique, considérons la base de règles suivante :

1. Si fleur et graine alors phanérogame
2. Si phanérogame et graine nue alors sapin
3. Si phanérogame et 1-cotylédone alors monocotylédone
4. Si phanérogame et 2-cotylédone alors dicotylédone
5. Si monocotylédone et rhizome alors muguet
6. Si dicotylédone alors anémone
7. Si monocotylédone et non rhizome alors lilas
8. Si feuille et fleur alors cryptogame
9. Si cryptogame et non racine alors mousse
10. Si cryptogame et racine alors fougère
11. Si non feuilles et plante alors thallophyte
12. Si thallophyte et chlorophylle alors algue
13. Si thallophyte et non chlorophylle alors champignon
14. Si non feuille et non fleur et non plante alors colibacille

Ainsi, 1. signifie que "si la plante possède une fleur et une graine, alors la plante est un phanérogame".

Le problème posé est alors de déterminer une plante ayant les caractéristiques suivantes :

{rhizome, fleur, graine, 1-cotylédone}.

Ce sera la base de faits initiale.

En chaînage avant, on obtient la chaîne de dérivation suivante :

- 1.  $\rightarrow$  phanérogame
- 3.  $\rightarrow$  monocotylédone
- 5.  $\rightarrow$  muguet

D'où la solution "muguet", les autres règles ne se déclenchant pas.



## Comment enrichir et maintenir la base de connaissances :

Une fois que le choix d'une représentation est définie pour une base de connaissance donnée il faut l'alimenter puis par la suite la maintenir : c'est le rôle du module d'acquisition des données. La base de donnée doit être facilement manipulable et extensible pour ajouter, modifier ou supprimer les connaissances stockées grâce à son symbolisme de représentation.

Suivant les systèmes, l'acquisition des nouvelles connaissances se fait par :

- Les ingénieurs cogniticiens.
- L'expert.
- Le système lui-même grâce à son système d'exploitation..

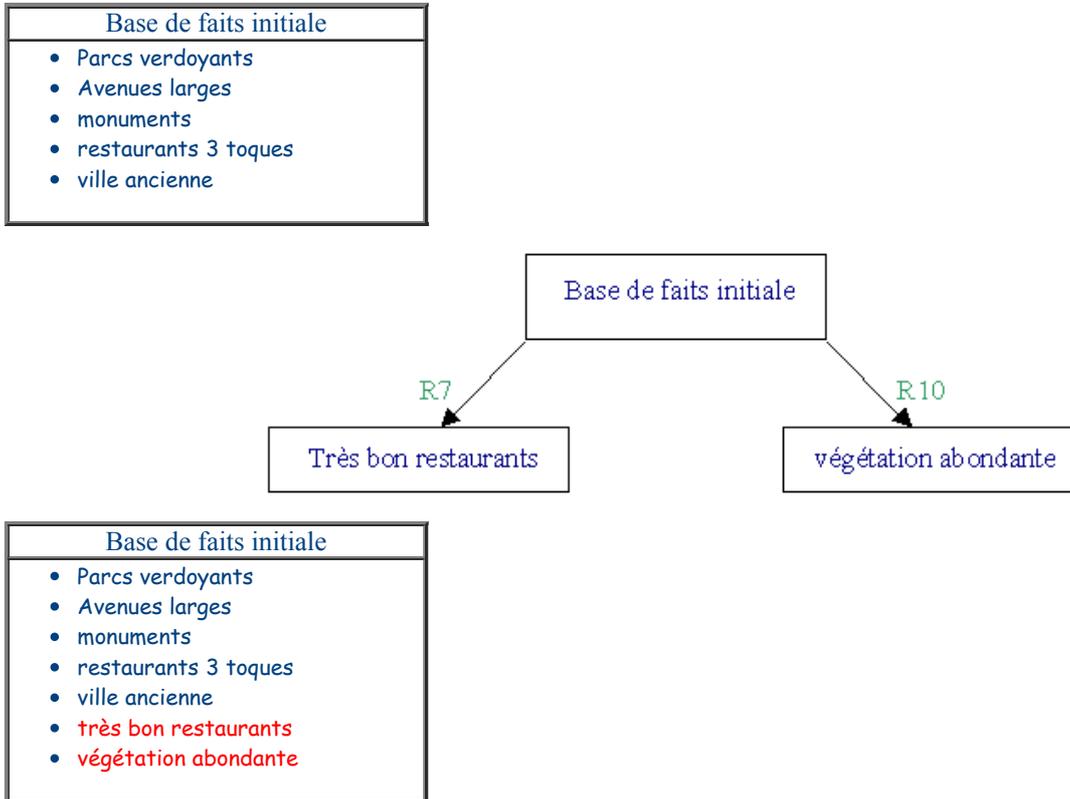
- L'apprentissage automatique.
- L'acquisition automatique, pour des compteurs physiques notamment.



## Exercices :

### exercice 1 :

A partir de la base de règles "ville" ainsi que de la base de faits associée, on procède aux déductions suivantes :



Compléter la base de faits en procédant aux déductions suivantes.

[consulter la solution](#)

### exercice 2 :

A partir de la base de règles "ville" ainsi que de la base de faits associée, on cherche à vérifier si la ville est une ville qui mérite le voyage. Construire la solution sous forme d'un graphe.

[consulter la solution](#)