

# Intelligence Artificielle



Dorra BEN AYED

# Chapitre 4

## Systemes Experts et Moteurs d'Inférence





# Définition

- ❖ Un **système expert** est un logiciel qui reproduit le comportement d'un expert humain accomplissant une tâche intellectuelle dans un domaine précis.

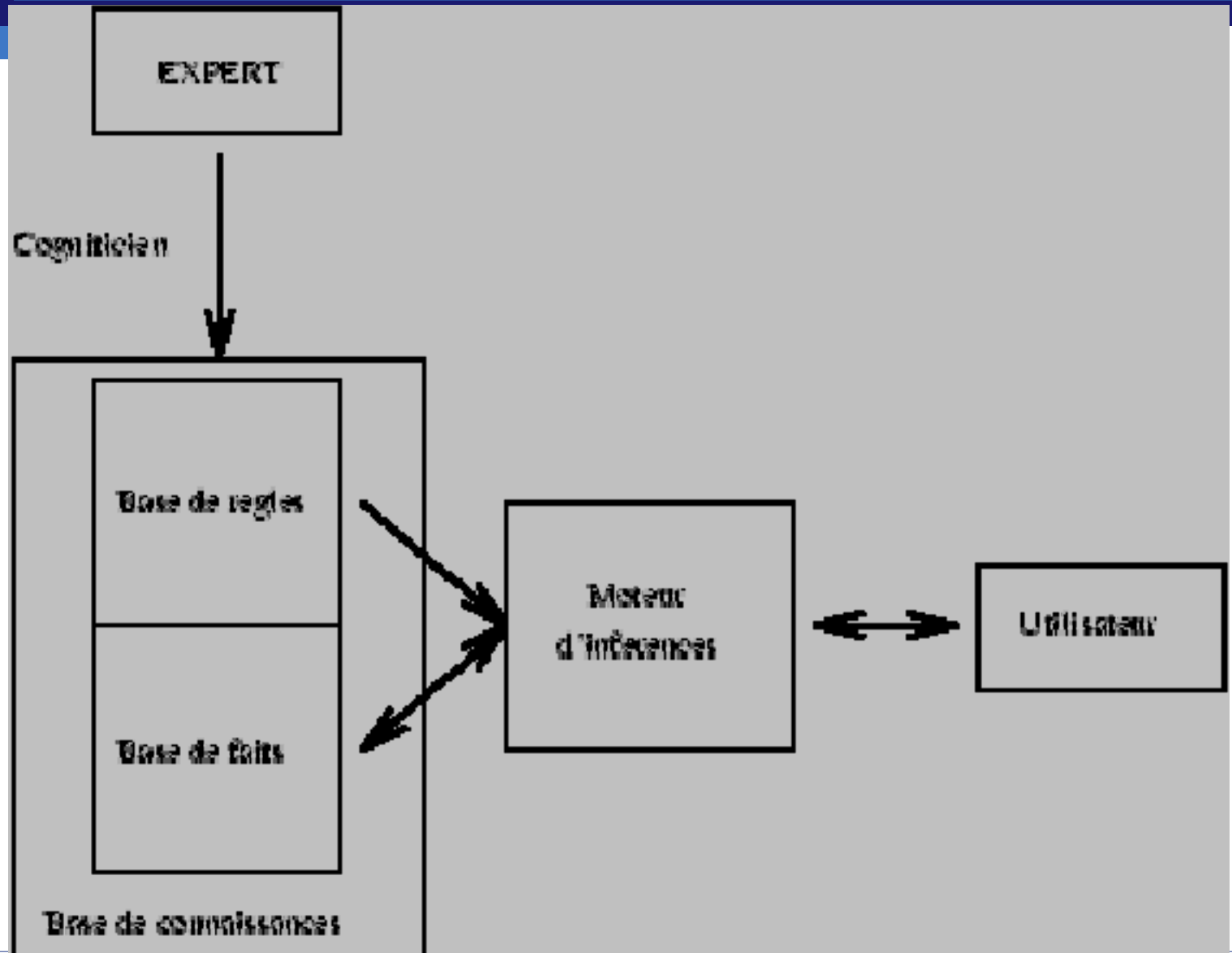


# Structure d'un SE

Un système expert ( SE) est composé de deux parties indépendantes :

- une **base de connaissances** (mémoire d'un SE)
  - **base de faits (BDF)** qui contient les séquences de faits établis et ayant une valeur de vérité vraie (constitue la partie statique)
  - **base de règles (BDR)** qui contient l'ensemble des règles de production pouvant être appliqués aux faits. (il s'agit de la partie dynamique)
- un **moteur d'inférences (MI)** Il s'agit du cerveau du SE. C'est un programme qui simule le raisonnement humain

# Structure d'un SE





# Cycle d'un Moteur d'inférence

Le moteur d'inférence fonctionne en deux phases:

- **Phase d'évaluation**

- **1/ Etape de Filtrage ou détection** : Elle consiste à définir pour l'ensemble des règles de BC, les règles potentiellement applicables → résultat ensemble de règles
- **2/ Etape de Sélection avec réduction de conflits**: Elle consiste à choisir parmi l'ensemble des règles applicables, la règle à appliquer effectivement → résultat une règle

- **Phase d'exécution**

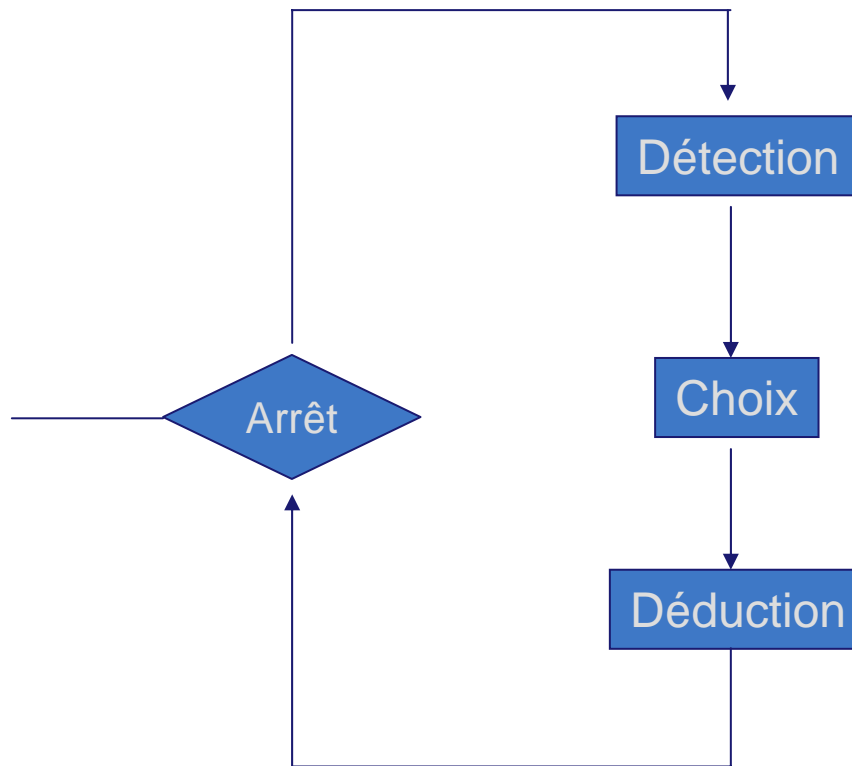
- Elle consiste à appliquer la règle choisie et mettre à jour la base de faits BDF

Le moteur d'inférence exécute ces différentes phases de façon cyclique jusqu'à une condition d'arrêt soit vérifiée:

- Un objectif atteint
- Epuisement de toutes les connaissances
  - Etape de filtrage ne fournit aucune règles potentiellement applicable
  - Epuisement normale des règles



# Cycle d'un Moteur d'inférence





# Caractéristiques d'un Moteur d'Inférence

La programmation d'un MI nécessite la sélection d'un ensemble de critères:

- ❖ 1/ Mode d'invocation des règles
- ❖ 2/ La stratégie de recherche
- ❖ 3/ Régime de contrôle
- ❖ 4/ Critère de monotonie





# 1/Mode d'invocation des règles

La programmation d'un MI nécessite la sélection d'un ensemble de critères:

## ❖ 1/Mode d'invocation des règles

## ❖ 2/ La stratégie

## ❖ 3/ Régime

## ❖ 4/ Critère

- MI à approche de chainage avant
- MI à approche de chainage arrière
- MI à approche de chainage mixte



# MI à chainage avant

1) Les MI à chaînage avant : un MI à chaînage avant (forward chaining) détermine le résultat (ou but) à partir de la BDF. Les règles à *déclencher* à chaque cycle sont celles dont les prémisses appartiennent à la BDF. L'exécution de ces règles modifie la BDF et d'autres règles peuvent alors être déclenchées au cycle suivant. Ce principe sera répété jusqu'à ce que le but soit dans la BDF.

Exemple :

soit  $BDF = \{A, B, C, D\}$

et soit la règle :  $A, C \rightarrow G$

cette règle est déclenchable et on obtient  $BDF = \{A, B, C, D, G\}$

L'étape de génération de conflit correspond à rassembler toutes les règles dont les prémisses sont dans la BDF.



# MI à chaînage arrière

2) Les MI à chaînage arrière : un MI à chaînage arrière (backward chaining) détermine l'ensemble des règles qu'il faut invoquer pour aboutir au but (fait à établir). Le but sera alors remplacé par les prémisses qui vont constituer les nouveaux faits à établir. Les règles à tirer à chaque cycle sont alors celles dont les conclusions sont égales au (ou contiennent le) fait à établir. L'exécution de ces règles ne modifie pas la BDF mais elle remplace le problème initial par d'autres. Ce type de raisonnement correspond à une décomposition du problème en un ensemble de sous problèmes.

## Exemple :

Soit le but à établir  $P$ , et soit la règle :

$$A, B, C \rightarrow P$$

Pour résoudre le problème  $P$ , il suffit de résoudre les sous problèmes  $A$ ,  $B$  et  $C$ .

Ce mécanisme sera répété jusqu'à ce que la dernière liste de faits à établir soit entièrement dans la BDF.

L'étape de génération de conflit consiste à rassembler toutes les règles dont les conclusions contiennent le fait à établir.



# MI à Chainage Mixte

3) Les MI à chaînage mixte : Ce type de moteur d'inférence peut être choisi lorsqu'une partie des faits du problème est à établir, l'autre est considérée déjà établie. Les conditions de déclenchement des règles dans ce cas peuvent porter sur les deux types de faits. Donc, pour résoudre le problème on sera amené à prendre en considération les faits déjà établis et de remplacer le problème à résoudre par une liste de sous problèmes. Ceci, correspond à un mode de raisonnement mixte faisant intervenir simultanément le chaînage avant et le chaînage arrière.



## 2/ La stratégie de recherche

La programmation d'un MI nécessite la sélection d'un ensemble de critères:

❖ 1/ Mode d'invocation des règles

❖ 2/ La stratégie de recherche

❖ 3/ Règles

❖ 4/ ...

Au cours des cycles de recherche d'un MI, on développe un **arbre de recherche** dans lequel chaque niveau correspond à l'ensemble des règles applicables (ensemble de conflits) Chaque règle déclenchée crée une nouvelle situation et de nouvelles règles à invoquer.

**Deux principales stratégies de recherche se présentent:**

- soit on développe toutes les règles d'un même niveau l'un après l'autre avant de passer au niveau suivant (**stratégie en largeur d'abord**)
- soit d'un niveau à un autre à chaque fois qu'on déclenche une règle et on ne revient aux règles restantes que si on épuise toutes les règles en **profondeur (stratégie en profondeur d'abord)**  
→ Le retour arrière dans le cas où la recherche en profondeur échoue sera appelé: **backtracking**



## 3/ régime de contrôle

La programmation d'un MI nécessite la sélection d'un ensemble de critères:

- ❖ 1/ Mode d'invocation des règles
- ❖ 2/ La stratégie de recherche
- ❖ 3/ Régime de contrôle
- ❖ 4/ Critère de monotonie

Le régime de contrôle d'un MI peut être:

- **Irrévocable:**

- L'application d'une règle dans un cycle du MI n'est jamais remise en cause et on n'opère pas de backtracking. S'il n'y a plus de règles à appliquer. Le MI s'arrête et signale un échec sans faire retour en arrière

- **ou par tentative:**

- Ce régime peut remettre en cause des règles déjà appliquées si elles n'ont pas abouti, et faire un backtracking en retirant aussi les faits qui en étaient déduits



## 4/ Critère de monotonie

La programmation d'un MI nécessite la sélection d'un ensemble de critères:

- ❖ 1/ Mode d'invocation des règles
- ❖ 2/ La stratégie de recherche
- ❖ 3/ Régime de contrôle
- ❖ 4/ Critère de monotonie

Un moteur d'inférence MI peut être :

- monotone:
  - En régime monotone: le MI ne fait qu'ajouter des faits à la BDF et n'élimine jamais une règle de BDR
- non monotone
  - En régime non monotone: le MI peut en cas de retour arrière par exemple retrancher de la BDF un fait précédemment ajouté



# Exercice d'application

❖ Base de Fait (BDF):  $\{ B, C \}$

❖ But : H

❖ Base des règles (BDR):

- R1:  $B \wedge D \wedge E \rightarrow F$
- R2:  $D \wedge G \rightarrow A$
- R3:  $C \wedge F \rightarrow A$
- R4:  $B \rightarrow X$
- R5:  $D \rightarrow E$
- R6:  $A \wedge X \rightarrow H$
- R7:  $C \rightarrow D$
- R8:  $X \wedge C \rightarrow A$
- R9:  $X \wedge B \rightarrow D$

Résoudre ce problème en appliquant l'approche de **chainage avant** selon une stratégie en **profondeur d'abord monotone** avec régime irrévocable :

- La 1ère règle qui apparaît dans l'ensemble des règles détectées est celle à appliquer
- Une règle appliquée est définitivement écartée de BDR





# Solution

## Solution 1

**Sous forme d'un tableau**

Numéro d'infé- rence	Filtrage	sélection	déduction

## Solution 2

**Sous forme d'un arbre  
de recherche en profondeur d'abord**

# Question?

