

## Chapitre 2

# **Formalisme des règles de production, et mode de raisonnement dans un Système à base de connaissances**

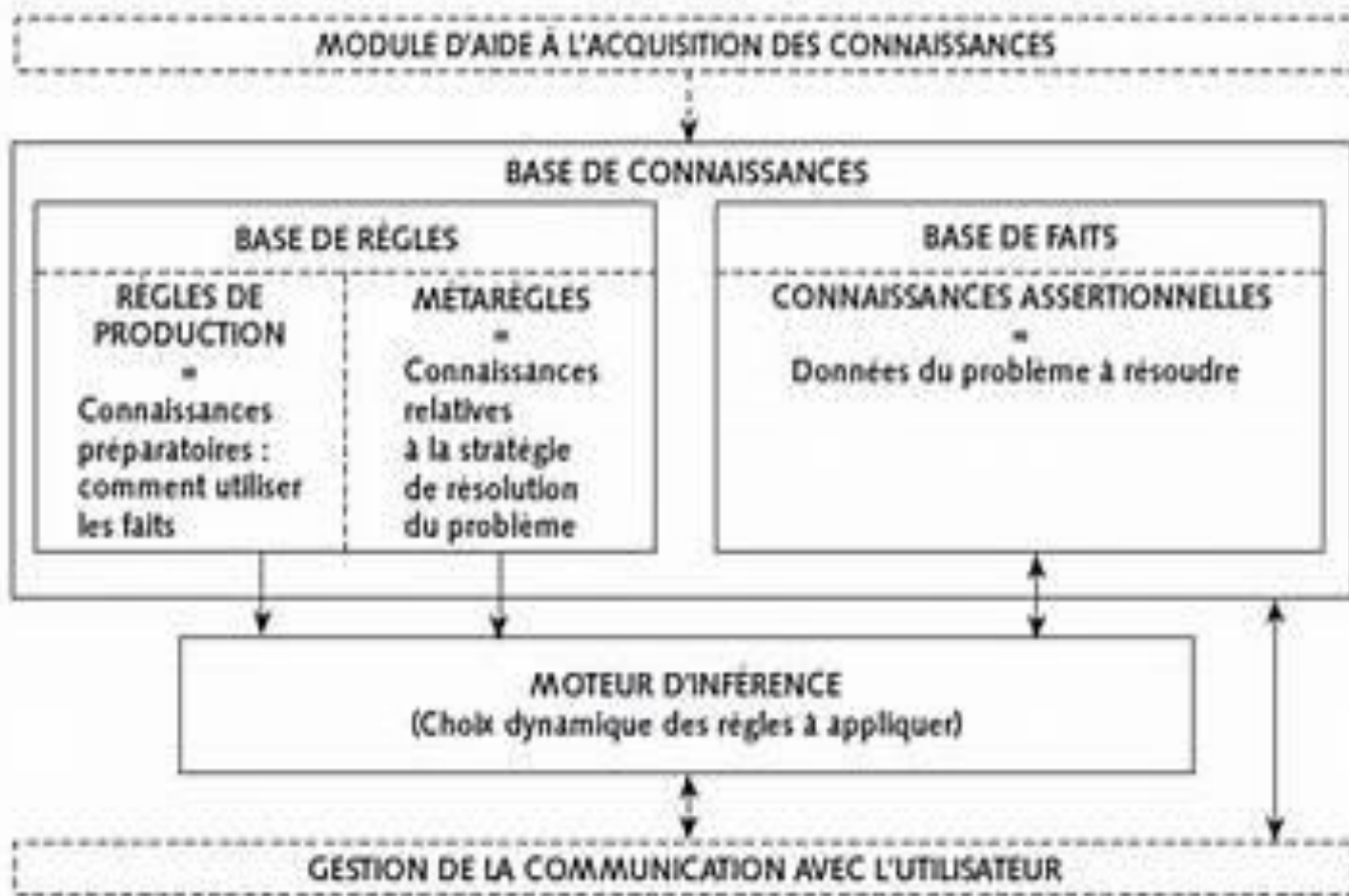
# Introduction

La conception de **systèmes experts** est fortement liée aux domaines de recherche sur **la cognition** qui étudient l'intelligence des êtres humains.

Il est nécessaire, pour concevoir des systèmes performants, d'assouplir la logique classique en intégrant des notions plus complexes **inspirées de la logique humaine**. C'est ainsi que des **logiques floues, modales ou temporelles** ont été proposées par l'IA.

# Définition d'un système expert

1. Un système expert est un logiciel qui reproduit le comportement de l'être humain dans ses activités de raisonnement afin de parvenir à un gain en temps et en efficacité dans le traitement de tâches de décision complexes.
2. Un système expert est défini comme un programme informatique qui peut conseiller, analyser, classer, communiquer, consulter, concevoir, explorer, anticiper, former, interpréter, justifier, gérer, planifier, apprendre et tester. Ces tâches constituent des problèmes complexes qui se réfèrent généralement à l'expertise humaine pour l'élaboration d'une solution.



## Plus simplement

- ▶ Les SE sont des systèmes informatiques résolvant des problèmes pour lesquels on ne dispose pas de solution algorithmique. Particulièrement dans le cas où les solutions algorithmiques sont à éviter à cause de l'explosion combinatoire.
- ▶ Ils sont des systèmes basés sur les connaissances d'une expertise humaine.

# Systeme à base de connaissances

Les premiers systèmes experts, développés dans les années 1970, se heurtent rapidement au problème de la gestion des connaissances nécessaires à leur fonctionnement.

- ▶ Les systèmes à base de connaissances, ou systèmes experts de deuxième génération, sont conçus autour de ces connaissances et intègrent dès leur conception les principes permettant leur maintenance.
- ▶ Toutefois les raisonnements des experts ne sont pas simples à cerner et sont souvent fortement liés aux détails des situations.

Dans un SBC :

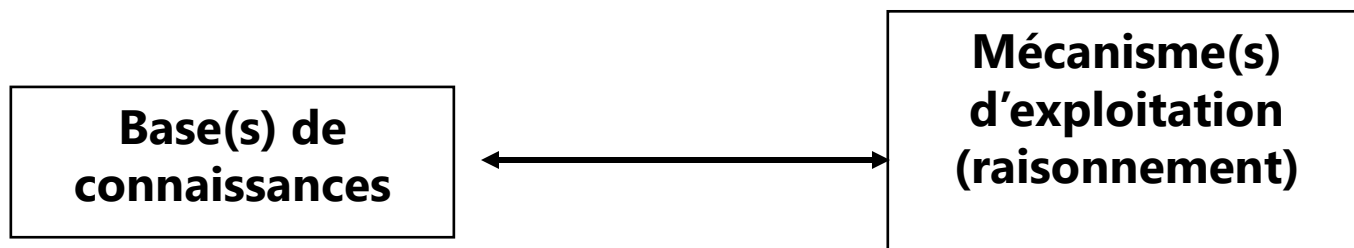
- Les règles ne sont plus **mises en vrac** dans une immense base de règles, mais elles sont organisées en **éléments de connaissance** sur le domaine.

Ces éléments **rassemblent des faits et des règles** de fonctionnement des différents objets pouvant être maniés par le système

- Les connaissances sur **le domaine** et les connaissances sur le(s) **raisonnement(s)** sont **séparées**, définissant ainsi différents **niveaux de connaissances**.

De tels systèmes nécessitent:

- une représentation adéquate des connaissances
- des mécanismes efficaces d'exploitation de ces connaissances, ou du raisonnement



**Schéma simplifié d'un système à bases de connaissances**



Pour le mécanisme de raisonnement, il faut spécifier

- les méthodes de résolution;
- les stratégies et les heuristiques: règles de comportement innées ou acquises;
- la méta connaissance : la connaissance sur la connaissance d'un problème donné.

# les catégories des connaissances

On peut distinguer 6 catégories des connaissances :

- **Connaissances de définition universelles** → un carré possède 4 cotés égaux
- **Connaissances évolutives** → Omar est de 80 cm de taille " (aujourd'hui).
- **Connaissances incertaines** → il paraît que demain il va faire beau
- **Connaissances vagues** → les jeunes élèves sont turbulents (Jeunes, turbulents ?).
- **Connaissances typiques** → habituellement, chaque jour on a au maximum 3 cours
- **Connaissances ambiguës** → il ya un avocat sur la table

# Le raisonnement

- Le raisonnement dans un système à bases de connaissances peut être schématisé comme un enchaînement de découvertes d'éléments de connaissances s'appuyant sur les informations connues, menant au but recherché.
- Un mode de représentation est en général restreint à un ou à un nombre limité de types de raisonnement.

# Processus d'ingénierie de connaissance

Pour le développement d'un SBC, doit intervenir :

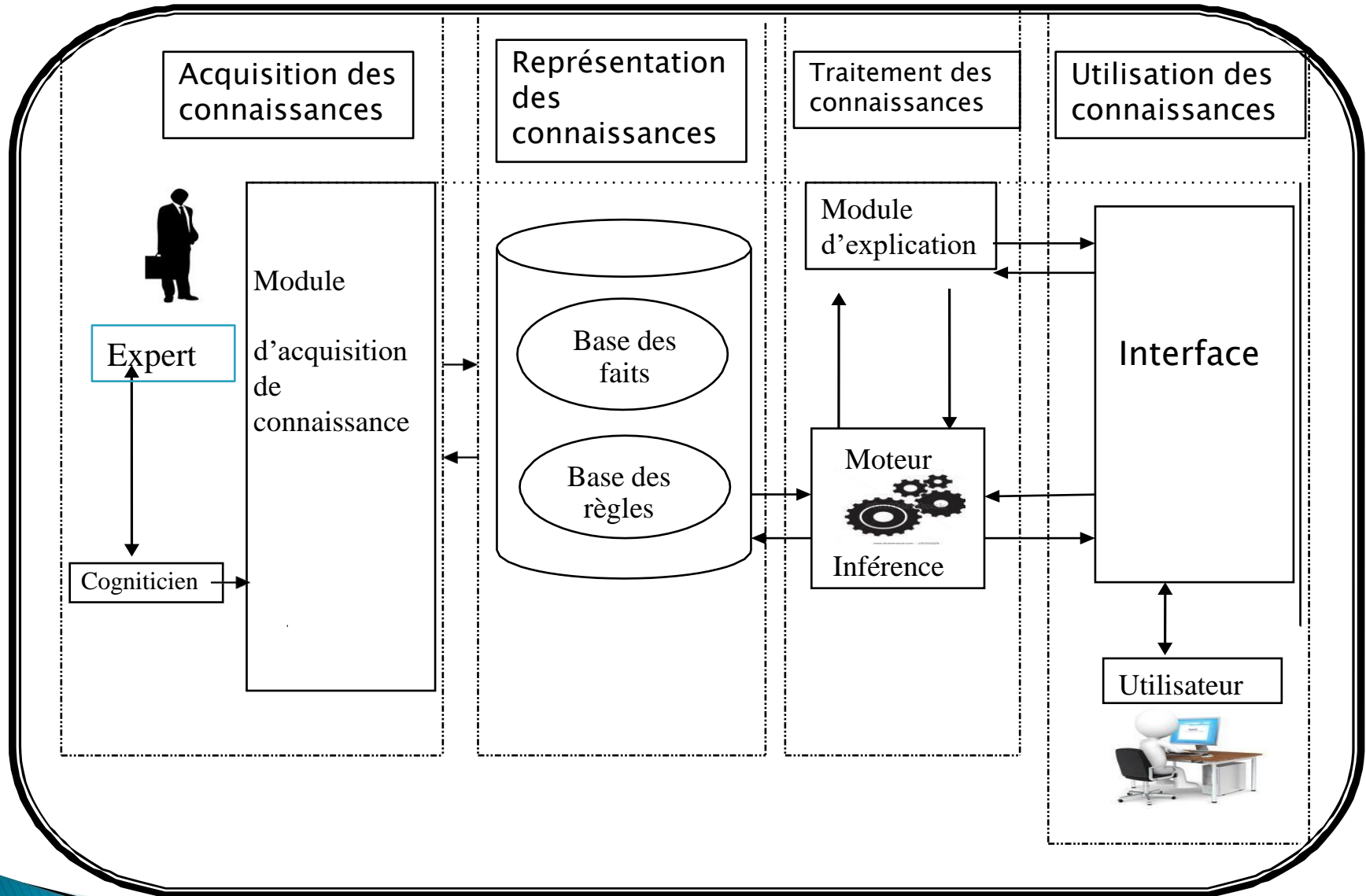
- L'ingénieur de connaissance qui est un expert en langage IA.  
Son rôle est de:
  1. trouver les outils et les logiciels nécessaires pour l'accomplissement du projet,
  2. aider l'expert du domaine à expliciter sa connaissance
  3. implanter cette connaissance dans la base de connaissances.
- l'expert du domaine qui fournit les connaissances nécessaires liées au problème.
- Le concepteur et le réalisateur du système
- l'utilisateur final dont le rôle est de spécifier l'application et de déterminer les contraintes de la conception.

## Nature de la connaissance de l'expert :

- Compilé,
- Volatile,
- Subjective
- Répartie

# Tâches Séquentielles en ingénierie des connaissances :

- Identifier la connaissance
- Expliciter la connaissance
- Structurer la connaissance
- Représenter la connaissance
- Utiliser, valider, raffiner et mettre à jour la connaissance.



## Architecture générale d'un système à base de connaissances

## La base de faits

Constitue la mémoire de travail du système, elle contient

- ▶ les données initiales
- ▶ données recueillies par les hypothèses émises
- ▶ les nouveaux faits prouvés.

**La base de faits, spécifique au problème à résoudre, s'enrichit au cours de la résolution de manière dynamique.**

Exemple :

La base de faits d'un système dédié à diagnostiquer les maladies peut contenir **au départ** les symptômes physiques et les résultats d'analyses.

**A la fin de la session le diagnostic et/ou le traitement à prendre**



# Type d'un fait

Les faits peuvent prendre des formes plus ou moins complexes.

Pour les faits élémentaires on a :

- ▶ booléennes (vrai ou faux),
- ▶ Symboliques appartenant à un domaine fini de symboles,
- ▶ réelles pour représenter les faits continus.

Exemple:

Actif est un fait booléen,  
profession est un fait symbolique  
rémunération est un fait réel.

## **Classification des SBC selon le formalisme de représentation de connaissance basé sur la logique classique**

- **Un système expert qui n'utilise que des faits booléens est dit d'ordre 0.**
- **Un système qui utilise des faits symboliques ou réels, sans utiliser de variables, est d'ordre 0+.**
- **Un système utilisant toute la puissance de la logique du premier ordre est d'ordre 1**

# Exemple de formules ou conditions

- Dans un système expert d'ordre 0, on pourra par exemple écrire des formules de la forme :

actif ou  $\neg$  actif

- Dans un système d'ordre 0+, on pourra trouver les formules :  
actif et (profession  $\neq$  medecin ou remuneration  $>$  20000)

- Dans un système d'ordre 1, on pourra trouver :  
 $\exists X$  maladie(X) et  $X \neq$  grippe et symptome(X) = forteFievre

# Méta fait et méta valeurs

Il n'est pas suffisant que le système ait des connaissances, il faut aussi qu'il ait des métas connaissances.

## Exemple

un système expert doit savoir si une valeur a été attribuée à un fait.

sinon, cette valeur pourra être demandée à l'utilisateur.

**Mais, si l'utilisateur ne peut pas répondre, il faudra que le système puisse le savoir afin de ne pas poser éternellement la même question.**

La seule manière d'attribuer une valeur à un tel fait sera alors de la déduire d'autres faits.

On dira que la valeur d'un fait est :

- **Connue** si une valeur lui a été attribuée ;
- **Inconnue** si aucune valeur ne lui a été attribuée et si aucune question à son sujet n'a été posée à l'utilisateur ;
- **Indéterminée** si le système ne lui a attribué aucune valeur et si l'utilisateur a répondu « je ne sais pas » à une question concernant sa valeur.

## La Base de Règles

Elle constituée une représentation formelle du savoir faire de(s) expert(s) c.a.d.

Contient l'ensemble des règles de raisonnement du système, Ces règles sont composées de:

1. Un ensemble de prémisses ou condition d'activation
2. Un ensemble de conclusions ou l'action à exécuter,

Une règle de production est une expression de la forme:

si Hypothèses alors Conclusions.

Où

- ▶ hypothèse est une condition ou une conjonction de conditions
- ▶ conclusion et un nouveau fait à introduire

## Exemple

Si fleur et graine alors phanérograme

Si phanérograme et graine nue alors sapin

Si phanérograme et 1-cotylédone alors monocotylédone

Si phanérograme et 2-cotylédone alors dicotylédone

Si monocotylédone et rhizome alors muguet

## Remarque

On peut facilement coder dans le formalisme règle de production

Si A ou B alors C      Si A alors B et C

Mais pas Si A alors B ou C



- ✓ La base de connaissance est modifiée en cours de temps soit par enrichissement automatique soit par ajout ou suppression de connaissance par l'expert du domaine. Ces opérations peuvent provoquer certaines anomalies.
- ✓ Pour éviter que le système soit dans un état d'inconsistance nous présentons des méthodes de vérification et de validation de bases de connaissances afin de maintenir la cohérence de la base de connaissance et du raisonnement ( Semestre2)

# Les métarègles et méta connaissances

Les méta connaissances s'expriment par :  
des méta faits et des métarègles

Il s'agit de règles destinées à guider le moteur d'inférence quant à la stratégie de résolution ou à la maintenance des bases de règles ou de faits

L'organisation d'une base de connaissances au moyen de métarègles reste essentiellement déclarative, contrairement à toute organisation basée sur une structuration à priori de l'ensemble des règles (écrire les règles dans un ordre donné,...)

## Exemple extrait des méta règle dans MYCIN

Si on cherche une thérapie alors considérer dans l'ordre les règles qui permettent de :

1. Acquérir des informations cliniques
2. Trouver quels organismes, s'ils existent sont cause de l'infection
3. Identifier les organismes les plus vraisemblables
4. Trouver tous les médicaments potentiellement utiles

# Le mécanisme d'exploitation de la base de connaissances

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 problèmes.
- La progression par avancement et retour arrière.
- L'évolution vers la solution par une progression d'étapes successives différentes.

# Le Moteur d'Inférence

- C'est un programme ou procédure qui simule le raisonnement humain.
- Il constitue la partie du système qui fournit des réponses aux questions des utilisateurs.
- Il met en œuvre le mécanisme de raisonnement chargé d'exploiter les connaissances.
- Il effectue les déductions nécessaires afin de proposer une réponse au problème posé.

# Cycles d'un Moteur d'Inférence

Le MI enchaîne une séquence de cycles au cours de son raisonnement jusqu'à aboutir au résultat désiré ou jusqu'à saturation.

Un cycle du MI est composé de deux phases :

- ▶ Une phase d'évaluation
- ▶ Une phase d'exécution

**Phase d'évaluation** est effectuée en 3 étapes :

**1. La restriction :**

C'est une étape optionnelle, le MI détermine un sous ensemble de règles et de faits dans lequel il est probable de trouver la solution.

La base de connaissance est supposée être décomposée en sous ensembles ou groupes de domaines distincts.

**2. Filtrage ou génération de conflits :**

Sélectionne les règles qui peuvent effectivement être déclenchées. Production d'un sous ensemble de règles  
Appelé ensemble conflits

### **3. Sélection ou Résolution de conflits :**

Choisir la règle à déclencher.

- ✓ Sans utiliser des heuristiques ou méta règle
- ✓ En utilisant des heuristique ou des méta règles (générique ou spécifique)



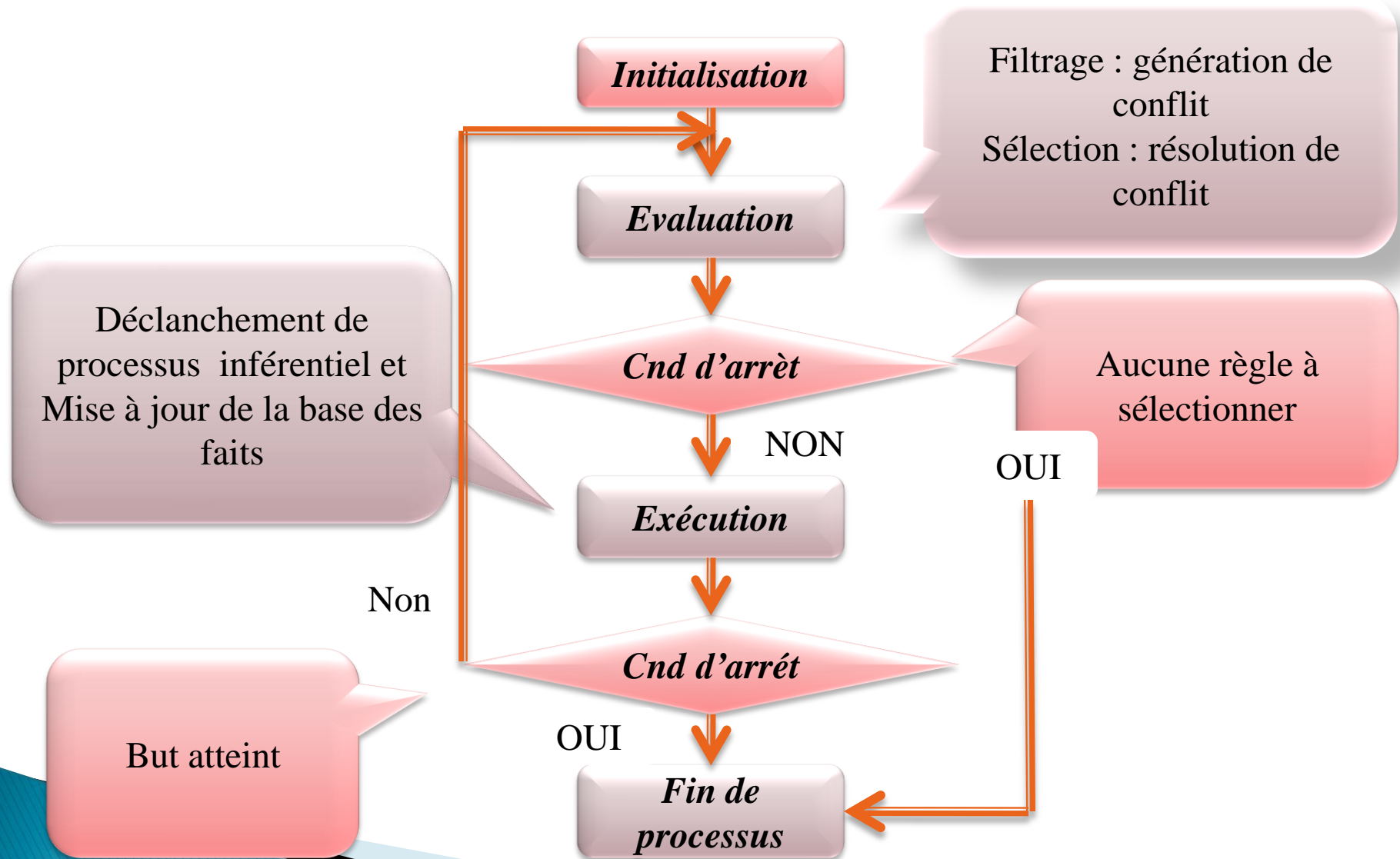
# Phase d'exécution

- Déclenchement de la règle sélectionnée.
- Les nouveaux faits résultants seront ajoutés à la base de faits.
- Si le but est atteint la recherche est arrêtée, sinon, un nouveau cycle est déclenché.

## Remarque

**Généralement cette phase consiste à ajouter des faits à la base de fait, mais il est aussi possible d'en retirer d'autres ou de déclencher des réactions ou exécuter des tâches supplémentaires.**

## Le moteur d'inférence :



## Algorithme de filtrage

$E \{ \}$  = ensemble conflit ,  $BR$  = base des règles

Début :

Pour toute règle  $R$  de la base des règles

Pour toute hypothèse de  $R$  qui appartient à la  
base des faits faire

    UPG (hypothèse, fait)

    Si unification alors

$E := E + R$

$BR = BR - R$

    Fin si

Fin pour

Fin pour

Fin

Algorithme de sélection ()

$E = \{\text{ensemble des règles applicables}\}$

Début :

Pour toute règle de l'ensemble E

Sélectionner une règle selon l'heuristique proposée

Fin pour

Fin

# Caractéristiques d'un Moteur d'inférence

1. Le mode de raisonnement (mode d'invocation des règles).
2. La stratégie de recherche qu'il utilise.
3. Le régime de contrôle.
4. Le critère de monotonie.

# Raisonnement en Chaînage Avant

- Saisie des faits initiaux
- Début
  - Phase de filtrage => Détermination des règles applicables
  - Tant que ensemble de règles applicables n'est pas vide ET que le problème n'est pas résolu Faire
    - Phase de choix => Résolution des conflits
    - Appliquer la règle choisie (exécution)
    - Modifier (éventuellement) l'ensemble des règles applicables
  - Fin faire
- Fin

## Remarques Importantes:

- ▶ Une règle n'est déclenchée qu'une seule fois
- ▶ L'algorithme termine toujours
- ▶ Les conclusions des règles peuvent être des faits négatifs
- ▶ Différentes situations en cas d'utilisation de faits négatifs
  - \*  $F \in BF \rightarrow$  fait établi
  - \*  $\neg F \in BF \rightarrow$  la négation du fait est établi
  - \*  $F \notin BF$  et  $\neg F \notin BF$  Aucune information ni le fait ni sa négation est établi
  - \*  $F \in BF$  et  $\neg F \in BF$  la base de faits est incohérente

# ALGORITHME DU CHAINAGE AVANT

ENTREE : BF, BR, F

DEBUT

TQ F n'est pas dans BF ET QU'il existe dans BR une règle applicable  
FAIRE

- choisir une règle applicable R (étape de résolution de conflits, utilisation d'heuristiques, de métarègles)
- $BR = BR - R$  (désactivation de R)
- $BF = BF \cup \text{concl}(R)$  (déclenchement de la règle R, sa conclusion est rajoutée à la base de faits)

FINTQ

SI F appartient à BF ALORS

- F est établi

SINON

- F n'est pas établi

► FIN



# Raisonnement en chaînage arrière

Le principe est le suivant :

- Le moteur recherche les règles qui concluent sur le but à vérifier, et s'assurent que ces règles sont "déclenchables".
- La règle est déclenchable si ses prémisses sont vérifiées.
- Si parmi les règles sélectionnées, une règle est déclenchable , alors le but est vérifié.
- Si ce n'est pas le cas, alors les prémisses à vérifier deviennent de nouveaux buts, appelés sous-buts, et le processus est réitéré.

Les principales conditions d'arrêt :

- L'ensemble des sous-buts est vide (succès) = tous les sous-buts ont été vérifiés et le problème est résolu
- Impasse ou échec : Soit un des sous - buts n'est pas vérifiable avec la règle courante et il faut choisir une nouvelle règle pour le vérifier, et si cela n'est pas possible, alors il y a échec.

## Principe de l'Algorithme chaînage\_arrière ()

But initial placé au sommet d'une pile

Détection des règles qui concluent sur ce but

Résolution de conflits

Exécution de la règle : les prémisses de la règle deviennent de nouveaux sous-buts à atteindre (empiler dans la pile)

Arrêt : Pile vide (succée) ou aucune règle applicable (echec)

## chaînage arrière

**Entrée :** F base de faits, R base de règles, P ensemble de faits à prouver

**Début**

**Tant que**  $P \neq \emptyset$

Extraire (en l'effaçant) de P un fait f

**Si** le fait f n'appartient pas à F alors

S  $\leftarrow$  ensemble des règles concluant sur f

**Si**  $S \neq \emptyset$  alors A  $\leftarrow$  choix d'une règle de S

Prouver toutes les hypothèses en prémisses de A

(une hypothèse est

soit un fait  $\in$  ou  $\notin$  à F

soit un sous but appeler alors chaînage arrière)

**Si** A est applicable alors Appliquer la règle A

**Sinon** Choisir une autre règle de S

**Fin si**

**Fin si**

**Fin si**

**Fin Tant que**

**Fin**

ChainageArriere(BF, BR, F)

Si  $F \subset BF$  alors

F est observable

Sinon

F n'est pas observable

Fin Si

ChainageArriere(BF, BR, F)

DEBUT TRAITEMENT

Soit R les règles R(i) de BR telles que  $F \subset \text{Conclusion}(R(i))$

Si  $R = \{\emptyset\}$  alors

Retourner( $\emptyset$ )

Sinon

Pour toutes les règles R(i) de R telles que  $F \subset \text{Conclusion}(R(i))$  Faire

Si  $\text{Condition}(R(i)) \subset BF$  alors

$BR \leftarrow BR - \{R(i)\}$

Pour toutes les règles R(j) telles que  $\text{Conditions}(R(j)) \subset BF$  Faire

$R \leftarrow R - \{R(j)\}$

Fin Pour

Retourner( $\text{Conclusion}(R(i))$ )

Sinon

Pour tout F de  $\text{Conditions}(R(i))$  Faire

$BF \leftarrow BF \cup \text{ChainageArriere}(BF, BR, F)$

Fin Pour

Fin Si

Fin Pour

Fin Si

FIN TRAITEMENT

## Question 1

Traduire les énoncés suivants en formules logiques

1. Bien que personne ne fasse de bruit, Ghofrane n'arrive pas à se concentrer
2. Si personne ne fait de bruit, Ghofrane répondra au moins à une question
3. Tout le monde a lu un livre de logique
4. Ce livre de logique a été lu par tous le monde

### Question2 Soit la clause suivante

$$C = R((f(g(x), x), g(x)) \vee R(f(y, g(z)), y)$$

Peut on trouver une clause facteur de C ? justifier votre réponse .  
Si oui, donner la clause facteur

### Question3

Etablir la validité de la déduction suivante.

Tout homme est un primate. Les dauphins ne sont pas des primates. Il y a des dauphins qui sont intelligents. Par conséquent, on peut ne pas être un homme et être intelligent.

### Question4

Soit F la formule suivante :  $F : \exists x (p(x) \rightarrow (q(x) \vee \forall y p(y))) \wedge \forall x \exists y (q(y) \rightarrow p(x))$

Donner la forme prénexe de F , la forme de Skolème puis mettre sous forme clausale.



## Exercice

A partir des énoncés suivants :

1. Pour tout crime, il y a quelqu'un qui l'a commis
2. Seuls les gens malhonnêtes commettent des crimes
3. On n'arrête que les gens malhonnêtes
4. Les gens malhonnêtes arrêtés ne commettent pas de crimes
5. Il y a des crimes

On veut déduire ; en utilisant le raisonnement par réfutation que  
➤ il y a des gens malhonnêtes en liberté

En notant

- $C(x)$  :  $x$  est un crime
- $M(y)$  :  $y$  est une personne malhonnête
- $CC(x,y)$  : la personne  $y$  a commis le crime  $x$
- $A(y)$  :  $y$  est une personne arrêtée

## Exemple

Soit la base de règles suivantes :

R1 :  $(\neg A \rightarrow B) \rightarrow P$

R2 :  $(P \text{ et } Q) \rightarrow F$

R3 :  $(C \rightarrow A) \rightarrow Q$

R4 :  $F \rightarrow (D \rightarrow K)$

R5 :  $K \rightarrow (M \text{ et } L)$

La base initiale de faits est : (A, D).

Prouvez le fait M par chaînage avant. Puis par chaînage arrière



## Exercice

Soient la base de règles suivante :

R1 : B et D et E  $\rightarrow$  C

R2 : P et C et S  $\rightarrow$  T

R3 : S et B  $\rightarrow$  T

R4 : M et N et S  $\rightarrow$  K

R5 : I et J  $\rightarrow$  B

R6 : A et B  $\rightarrow$  C

R7 : G  $\rightarrow$  F

R8 : T  $\rightarrow$  K

R9 : H  $\rightarrow$  A

R10 : H et S et B et I  $\rightarrow$  G

Les faits : I, S, B, H.

Les faits qui ne sont pas dans la base et ne sont pas des conclusions de règles sont supposés demandables.

1. dans quel ordre sont conclus les nouveaux faits lorsque l'on utilise une stratégie de recherche par chaînage avant.
2. dans quel ordre sont conclus les nouveaux faits lorsque l'on utilise une stratégie de recherche par chaînage arrière en largeur d'abord. Pour le but T.
3. dans quel ordre sont conclus les nouveaux faits lorsque l'on utilise une stratégie de recherche par chaînage arrière en profondeur d'abord. Pour le but T.

## Exemple 2

R1 : Si B et D et E alors F

R2 : Si D et G alors A

R3 : Si C et F alors A

R4 : Si B alors X

R5 : Si D alors E

R6 : Si A et X alors H

R7 : Si C alors D

R8 : Si X et C alors A

R9 : Si X et B alors D

BF = {B, C}

But à démontrer : H

- Faire le raisonnement par chaînage avant puis chaînage arrière

## La propagation de la certitude dans le moteur d'inférence se fait ainsi :

- Lorsqu'une règle de certitude CA est déclenchée par des prémisses de certitudes  $(CF_1, \dots, CF_n)$ , sa conclusion prend la certitude  $CA * \min(CF_i)$
- Si un fait a déjà une certitude CF1 et qu'il est inféré à nouveau avec la certitude CF2, on lui attribue la certitude :  $CF_1 + CF_2 - CF_1 * CF_2$

On considère système expert avec un moteur d'inférence en chaînage arrière

**base de connaissances :**

- **R1** : Si B et D alors F (1)
- **R2** : Si E et D alors A (0,8)
- **R3** : Si C et F alors A (0,5)
- **R4** : Si B alors X (1)
- **R5** : Si X et A alors H
- **R6** : Si F alors D (0,7)
- **R7** : Si X et C alors A (0,9)
- **R8** : Si E et G alors D (0,9)

**Base des faits :**

- E
- G
- B et C sont demandables

Questions

1. Construire le graphe et/ou pour le but F. Présente t il une anomalie ??
2. Peut on déduire le fait H ? Expliquer (trace d'exécution), envisager les cas de dialogues possibles

## Exercice 2

Soit la base de règles suivante :

R1 : s'il fait beau aujourd'hui alors il fera beau demain (0,5)

R2 : s'il pleut aujourd'hui alors il pleuvra demain (0,5)

R3 : si le vent est d'ouest alors le temps est à la pluie (0,6)

R4 : s'il y a des nuages alors le temps est à la pluie (0,4)

R5 : si les araignées tissent des toiles lâches alors les animaux pressentent la pluie (0,4)

R6 : si les hirondelles volent bas alors les animaux pressentent la pluie (0,6)

R7 : si les animaux pressentent la pluie alors il pleuvra demain (0,4)

R8 : s'il pleut en Bretagne et le vent est d'ouest alors le temps est à la pluie (0,9)

R10 : s'il fait beau demain alors il pleuvra demain (-1)

avec les faits :

il fait beau aujourd'hui (0,7) ; il y a des nuages (0,3)

les araignées font des toiles lâches (0,4) ; le vent est d'ouest (1)

1. Donner la base de faits qui résulte d'un raisonnement en chainage avant
2. Donner la trace d'exécution (expliquer clairement) pour un raisonnement en chainage arrière pour le but :  
il pleuvra demain

# Les interfaces

Les interfaces sont des modules programmés qui tiennent lieu d'unité d'entrée et d'unité de sortie d'un système expert, elles permettent l'interaction avec les deux types d'opérateurs :

- l'expert et le cognitifien qui construisent la base de connaissances
- les utilisateurs qui demande des services du système.

# L'interface Expert

l'interface expert comporte les modules suivantes :

- Un éditeur de règles pour initialiser ou modifier le contenu de la base.
- Un éditeur de dialogue avec l'utilisateur permettant aux experts de prévoir les questions que le système à base des règles devra poser à l'utilisateur pour le guider dans sa recherche de solution.
- Un analyseur ou un compilateur de base pour traduire les connaissances des experts en structures de données reconnues par le moteur d'inférence.

# L'interface Utilisateur

l'interface utilisateur assure la communication entre l'utilisateur et la base de connaissance, il s'agit d'un module subordonné au moteur d'inférence, le module pose des questions à l'utilisateur au besoin, et recueille les réponses de ce dernier pour les transmettre au moteur d'inférence.

L'interface utilisateur est décomposée de deux modules :

- un module de dialogue avec la base de connaissance
- un module d'explication par lequel le moteur d'inférence explique comment il utilise le contenu de la base de connaissance pour répondre aux questions de l'utilisateur,



# Avantages

- Permanence
- Expérience multiple
- Grande disponibilité.
- Cout réduit

## Limites des SBC:

Ces systèmes apportent des solutions nouvelles ou parfois plus rapides mais soulèvent plusieurs difficultés.

**Acquisition des connaissances :** l'acquisition des connaissances requiert le codage des connaissances dans la base mais l'expert du domaine a souvent des difficultés pour exprimer explicitement ses connaissances, ce qui nuit à l'élaboration du système.

**Cohérence de la base de connaissances:** la vérification, la correction et la complétude de la base de connaissances est une tâche complexe. selon le domaine d'application, cette tâche peut même s'avérer impossible dans la mesure où des incohérences et des contradictions sont parfois inhérentes du domaine, par exemple dans le cadre juridique.

**La rigidité de la base de connaissances.** Toutes les connaissances sont représentées dans un formalisme unique. Il en résulte la difficulté de construire la base pour des problèmes qui s'expriment naturellement dans des formalismes différents.