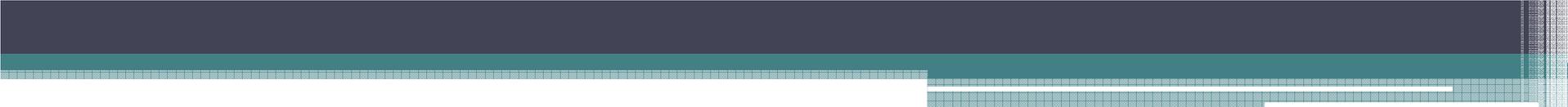


# Systeme à Base de Règles

EPITA SYBR  
SCIA

Adrien Pain - Version 1 (11)

Philippe Morignot – V2, V3, V4 (sept. 15), V5 (oct. 16)



# Plan du cours

1. Diagnostic
2. **Systemes experts**
3. Algorithme RETE
4. Travaux Pratiques en CLIPS
5. Projet

# Diagnostic et décision

- **Systemes experts**
  - Représentation de la connaissance
    - Logique formelle
    - Base de faits
    - Base de règles
  - Le moteur d'inférence
    - Chaînage avant
    - Chaînage arrière
    - Utilisation
  - Algorithme RETE

# Diagnostic et décision

- **Systemes experts**
  - Représentation de la connaissance
    - Logique formelle
    - Base de faits
    - Base de règles
  - Le moteur d'inférence
    - Chaînage avant
    - Chaînage arrière
    - Utilisation
  - Algorithme RETE

# Systeme expert

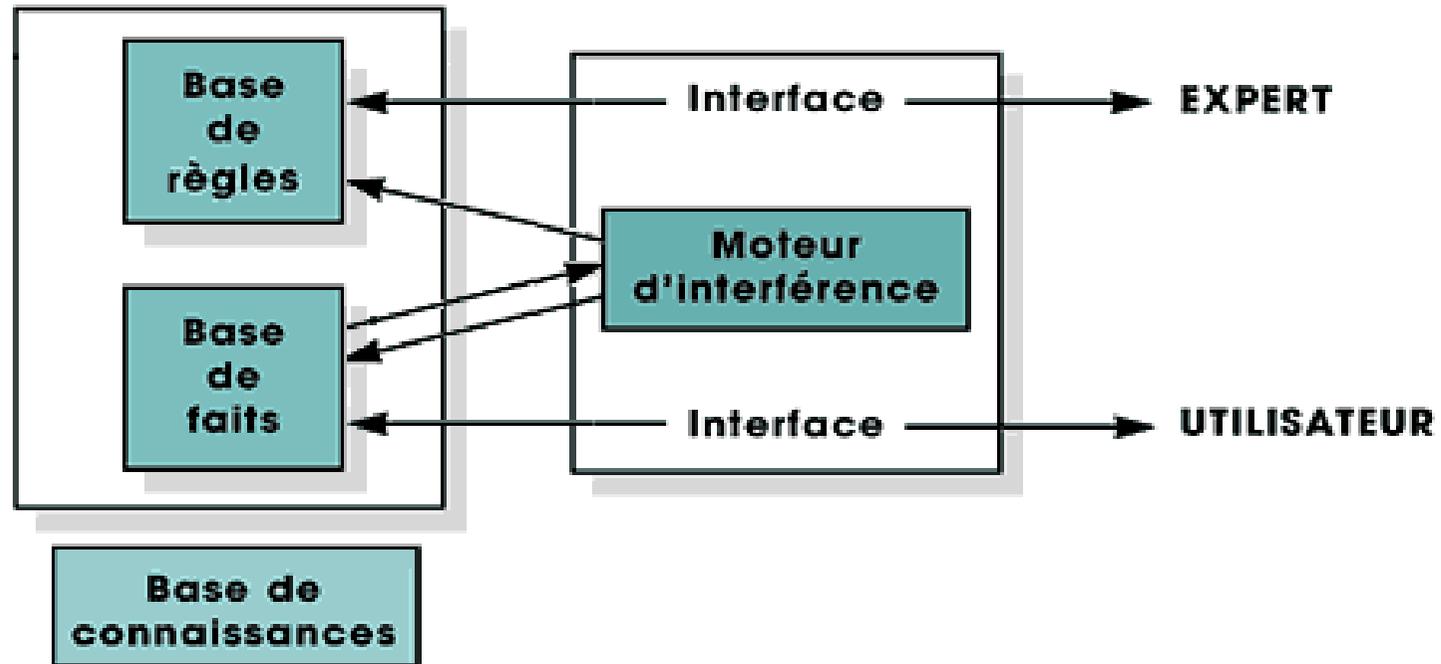
- Qu'est-ce qu'un système expert ?
  - *Aide à la décision*
    - Reproduire un **raisonnement** en tentant d'analyser un problème comme le ferait un **expert** humain dans un domaine précis.
  - *Exemples :*
    - Recherche de solution dans des domaines **peu formalisés**
      - diagnostic médical, prescription thérapeutique – MYCIN, ...
    - Détection de **pannes**
    - **Automatisation** de procédures (juridique, loi du travail, ...)
    - Régulation d'échanges boursiers
    - ...

# Systeme expert

- Qu'est-ce qu'un système expert ?
  - *Une base de connaissances :*
    - **Base de Règles** : modélise la **connaissance** du domaine considéré
    - **Base de Faits** : contient les informations concernant le problème traité
  - *Un moteur d'inférences :*
    - **Raisonner** à partir des informations contenues dans la BF et dans la BR
    - Capable de faire des **déductions** ou des **inférences**
  - *Une interface utilisateur (et une interface avec l'expert) :*
    - Possibilité d'**évolution du SE** au cours de l'exécution

# Systeme expert

- Qu'est-ce qu'un système expert ?



# Diagnostic et décision

- **Systemes experts**
  - Représentation de la connaissance
    - Logique formelle
    - Base de faits
    - Base de règles
  - Le moteur d'inférence
    - Chaînage avant
    - Chaînage arrière
    - Utilisation
  - Algorithme RETE

# Systeme expert

- Représentation de la connaissance : *logique formelle*
  - *Logique d'ordre 0 : algèbre de Boole – valeurs vrai/faux*
    - SI (« patient a mal ») ET (« patient a + de 40 ans ») ALORS ...
  - *Logique d'ordre 0+ : variables valuées, discrètes*
    - SI (symptôme = douleur) ET (âge  $\geq$  40) ALORS ...
  - *Logique d'ordre 1 : logique des prédicats du premier ordre*
    - SI (?x a mal) ET (?x a l'âge ?n) ET (?n  $\geq$  40) ALORS ...

⇒ Utilisation de la logique d'ordre 1

⇒ **JESS** : <http://www.jessrules.com/jess/index.shtml>

# Diagnostic et décision

- **Systemes experts**
  - Représentation de la connaissance
    - Logique formelle
    - Base de faits
    - Base de règles
  - Le moteur d'inférence
    - Chaînage avant
    - Chaînage arrière
    - Utilisation
  - Algorithme RETE

# Systeme expert

- Représentation de la connaissance : *base de faits*
  - *Mémoire de travail du SE*
    - *Variable* au cours de l'exécution
  - *Au début*
    - Contient les *faits initiaux*
  - *Pendant l'exécution*
    - Complétée par les *faits déduits* par le moteur d'inférences
  - *Exemple*
    - Diagnostic médical

# Diagnostic et décision

- **Systemes experts**
  - Représentation de la connaissance
    - Logique formelle
    - Base de faits
    - Base de règles
  - Le moteur d'inférence
    - Chaînage avant
    - Chaînage arrière
    - Utilisation
  - Algorithme RETE

# Systeme expert

- Représentation de la connaissance : *base de règles*
  - *Connaissance et savoir-faire de l'expert*
    - *Variable* au cours de l'exécution, mais uniquement via l'interface avec l'expert
  - *Utilise une logique d'ordre 1*
  - *Composition d'une règle*
    - Un ensemble de **prémisses** : conditions d'activation
    - Un ensemble de **conclusions** : actions à exécuter

# Systeme expert

- Représentation de la connaissance : *exemple*

- *Base de règles :*

```
(defrule is-uncle
  (<x> ^father-of <y>)
  (<z> ^sex male)
  (<z> ^brother-of <x>)
=> { <z> ^uncle-of <y> }
)
```

- *Base de faits :*

```
(fact Louis ^sex male)
(fact Louis ^brother-of Alexandre)
(fact Alexandre ^father-of Jean)
```

- *Déduction du système expert :*

```
(add-fact Louis ^uncle-of Jean)
```

# Diagnostic et décision

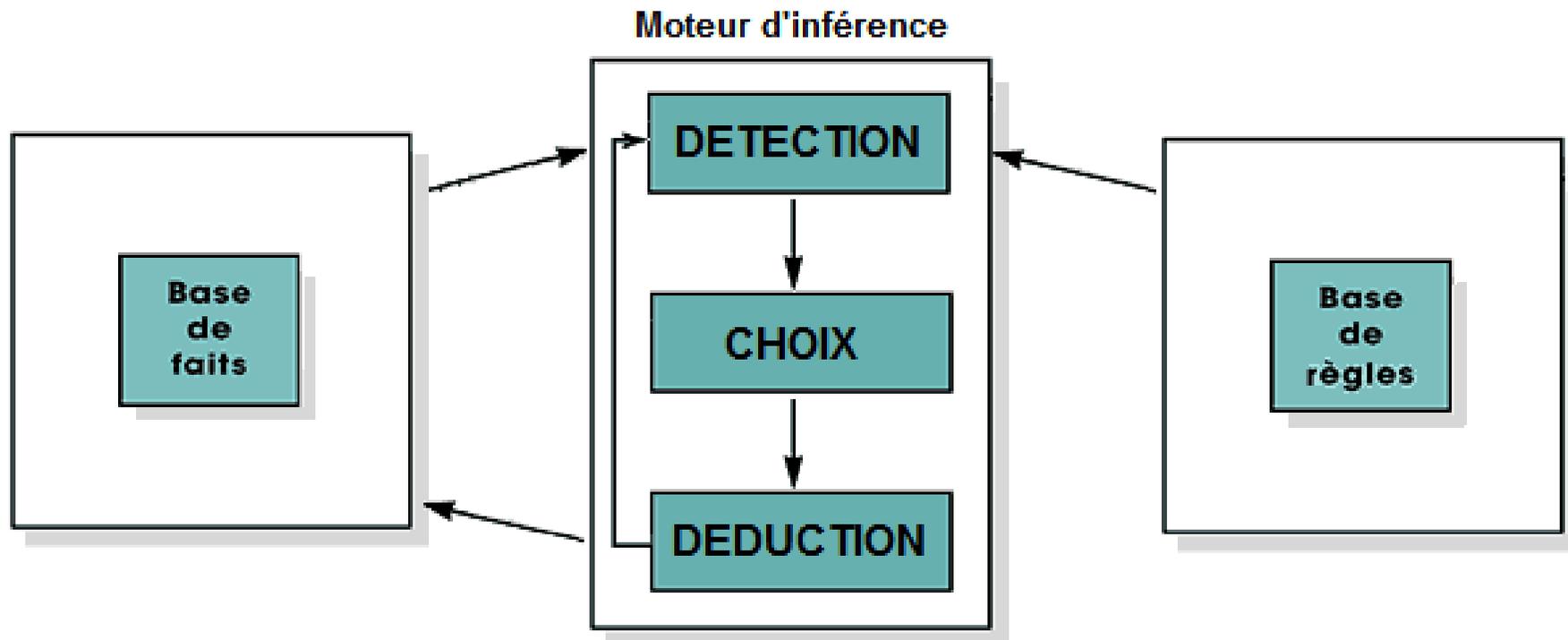
- **Systemes experts**
  - Représentation de la connaissance
    - Logique formelle
    - Base de faits
    - Base de règles
  - **Le moteur d'inférence**
    - Chaînage avant
    - Chaînage arrière
    - Utilisation
  - Algorithme RETE

# Systeme expert

- Moteur d'inférence
  - *Cycle d'inférence :*
    1. Phase de **détection** + matching
    2. Phase de **choix** + résolution de conflits
    3. Phase de **déduction** + activation
  - *Mise à jour éventuelle de la **base de faits***
  - *Boucle tant qu'il reste des règles à activer*
    - Pas de boucle infinie car une règle n'est déclenchable qu'au plus **une fois** par cycle
  - Fonctionne en **chaînage avant** ou **chaînage arrière**

# Systeme expert

- Moteur d'inférence



# Diagnostic et décision

- **Systemes experts**
  - Représentation de la connaissance
    - Logique formelle
    - Base de faits
    - Base de règles
  - Le moteur d'inférence
    - Chaînage avant
    - Chaînage arrière
    - Utilisation
  - Algorithme RETE

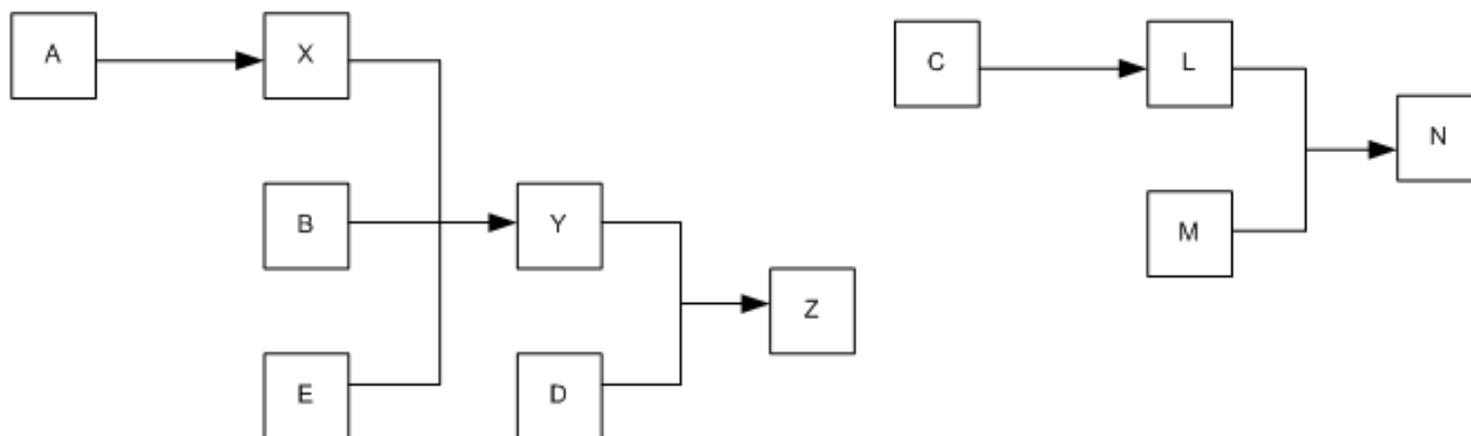
# Systeme expert

- Moteur d'inférence : *chaînage avant*
  - *Raisonnement **guidé par les données***
    - *S'appuie sur les **faits connus***
    - *Déclenche **toutes** les règles activables*
    - *Déduit des faits qu'il **rajoute** à la BF pour poursuivre l'inférence*
  - *Algorithme de fonctionnement*
    - *En **profondeur** d'abord*
    - *En **largeur** d'abord*

# Systeme expert

- Moteur d'inférence : *chaînage avant*
  - *Exemple en largeur d'abord*

- *Base de règles*

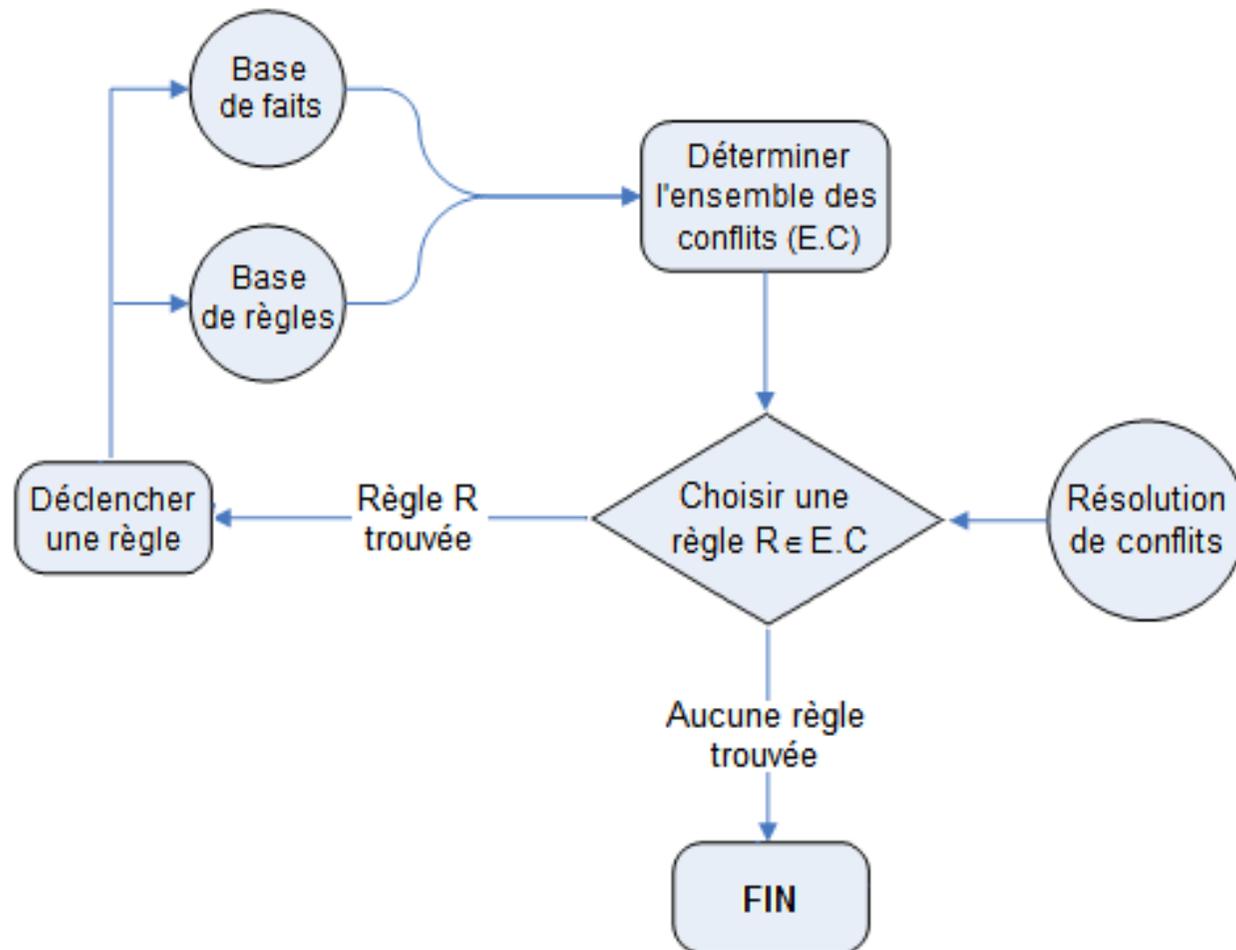


- *Base de faits : {A, B, C, D, E}*

⇒ *Résultat ?*

# Systeme expert

- Moteur d'inférence : *chaînage avant*



# Systeme expert

## Procédure CHAINAGE-AVANT(**BF**, **BR**)

```

POUR TOUT r dans BR, activable( r ) = VRAI // Initialisation
TANT QUE VRAI
  déclenchables <- ∅ // Ensemble des règles déclenchables
  POUR TOUT r dans BR
    SI activable( r ) = VRAI ALORS // Si la règle r n'est pas activable, elle n'est
      POUR TOUT p dans Prémisse(r) // pas déclenchable.
        SI !(p appartient à BF) ||
          (p contradictoire avec BF)
          ALORS continue POUR TOUT r // r non déclenchable
        FIN POUR TOUT
      AJOUTER r à déclenchables // sinon r déclenchable
    FIN POUR TOUT
  SI déclenchables = ∅ ALORS ARRET // Saturation de BF
  r <- CHOISIR(déclenchables) // Point de choix ...
  activable( r ) = FAUX // Eviter les bouclages sur les règles
  POUR TOUT c dans Conclusions(r), AJOUTER c dans BF
FIN TANT QUE

```

# Systeme expert

- Moteur d'inférence : exemple d'exécution en *chaînage avant*

- *Exemple d'exécution*

- *Base de règles :*

- $R_1 : a \vee b \Rightarrow add(e)$
    - $R_2 : b \wedge d \Rightarrow add(f)$
    - $R_3 : c \wedge d \wedge g \Rightarrow add(a)$
    - $R_4 : g \vee (d \wedge f) \Rightarrow add(c), remove(b)$
    - $R_5 : a \wedge d \wedge e \Rightarrow add(\neg b)$

- *Base de faits : { b, d, a }*

- *Ordre de résolution des conflits :  $R_4 > R_3 > R_5 > R_1 > R_2$*

$\Rightarrow$  *Résultat ?*

# Diagnostic et décision

- **Systemes experts**
  - Représentation de la connaissance
    - Logique formelle
    - Base de faits
    - Base de règles
  - **Le moteur d'inférence**
    - Chaînage avant
    - Chaînage arrière
    - Utilisation
  - Algorithme RETE

# Systeme expert

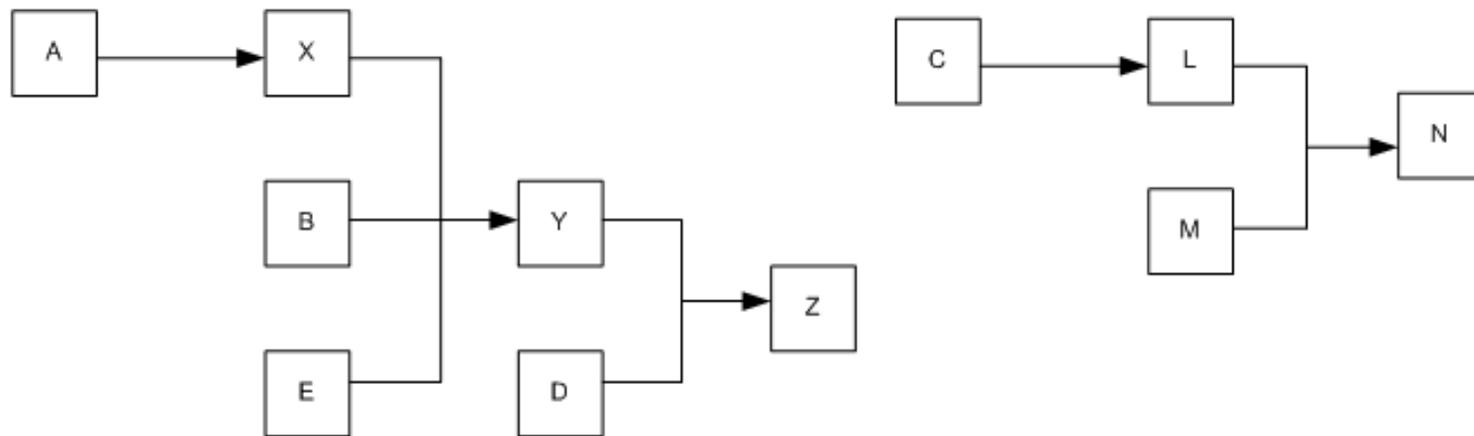
- Moteur d'inférence : *chaînage arrière*
  - *Raisonnement guidé par les buts à atteindre*
    - *Part du but à démontrer*
    - *Reconstruit un arbre d'induction (« ET/OU ») pour remonter aux causes*

# Systeme expert

- Moteur d'inférence : *chaînage arrière*
  - *Méthode de fonctionnement pour établir le fait  $f$*  :
    - Analyser quelles règles  $R_i$  le déclenchent
    - Déterminer la liste des faits à établir pour déclencher les règles  $R_i$
    - Pour chaque règle  $R_k$  de  $R_i$ 
      - si la règle est déclenchable : le fait  $f$  est démontré.
      - sinon : relancer le chaînage arrière récursivement sur chaque fait de la partie « conditions » de  $R_k$  qui n'est pas établi
    - Le moteur s'arrête :
      - Lorsque  $f$  est démontré OU
      - Lorsqu'il reste des faits requis non établis et qu'il n'y a plus de règles déclenchables. Dans ce cas, c'est un échec : le fait  $f$  n'est pas démontré.

# Systeme expert

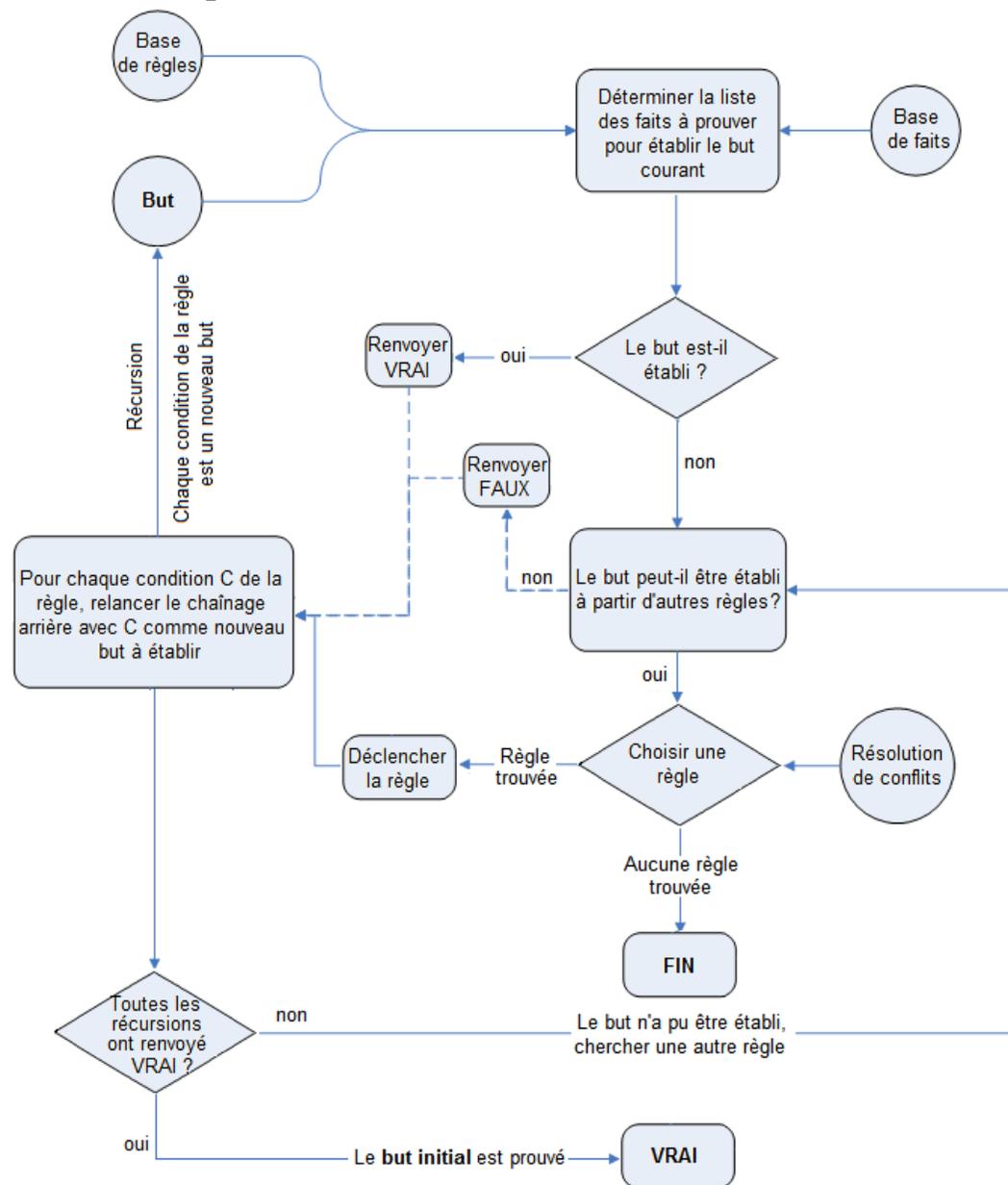
- Moteur d'inférence : exemple de *chaînage arrière*
  - *Base de règles*



- *Base de faits : { A, B, C, D, E }*

⇒ *Le fait Z peut-il être démontré ? Et le fait N ?*

# Systeme expert



# Systeme expert

*bool* CHAINAGE-ARRIERE(*BF*, *BR*, *f*)

```

SI f appartient-à BF ALORS renvoyer VRAI
POUR TOUT r dans BR, activable (r) = VRAI // Initialisation
conflit <- ∅ // Chercher les règles déclenchables
POUR TOUT r dans BR // Recherche des règles déclenchables
  SI activable (r) = VRAI &&
    f appartient-à Conclusions (r) &&
    POUR TOUT p dans Prémisses (r)
      p appartient-à BF &&
      p non contradictoire avec BF
    ALORS AJOUTER r à conflit
FIN POUR TOUT
SI conflit = ∅ ALORS renvoyer FAUX // Pas de règles pour prouver f
r <- CHOISIR(conflit) // Point de choix
activable (r) = FAUX // Eviter les bouclages sur les règles
SOIT p1, ..., pn les Prémisses(r) // Appel récursif sur les prémisses de r
renvoyer CHAINAGE-ARRIERE(BF, BR, p1) && ... && CHAINAGE-ARRIERE(BF, BR, pn)

```

# Systeme expert

- Moteur d'inférence : exemple d'exécution en *chaînage arrière*

▫ *Base de règles :*

$$\bullet R_1 : a \vee (e \wedge b) \quad \Rightarrow \quad f$$

$$\bullet R_2 : c \wedge d \quad \Rightarrow \quad f$$

$$\bullet R_3 : d \wedge h \quad \Rightarrow \quad c$$

$$\bullet R_4 : c \wedge g \quad \Rightarrow \quad e$$

▫ *Base de faits : { d, g, h }*

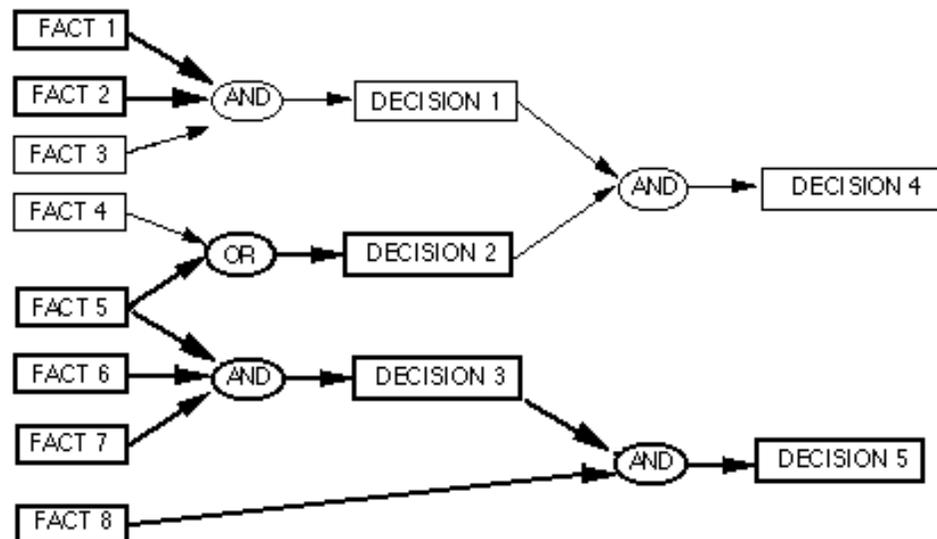
$\Rightarrow$  *Le fait f est-il vrai ?*

# Diagnostic et décision

- **Systemes experts**
  - Représentation de la connaissance
    - Logique formelle
    - Base de faits
    - Base de règles
  - **Le moteur d'inférence**
    - Chaînage avant
    - Chaînage arrière
    - Utilisation
  - Algorithme RETE

# Systeme expert

- Moteur d'inférence : *utilisation*
    - *Chaînage avant*
      - *Part des faits et déduit le plus de nouveaux faits possibles*
- ⇒ *Utilisé quand on veut découvrir quelles décisions peuvent être prises à partir des données dont on dispose*



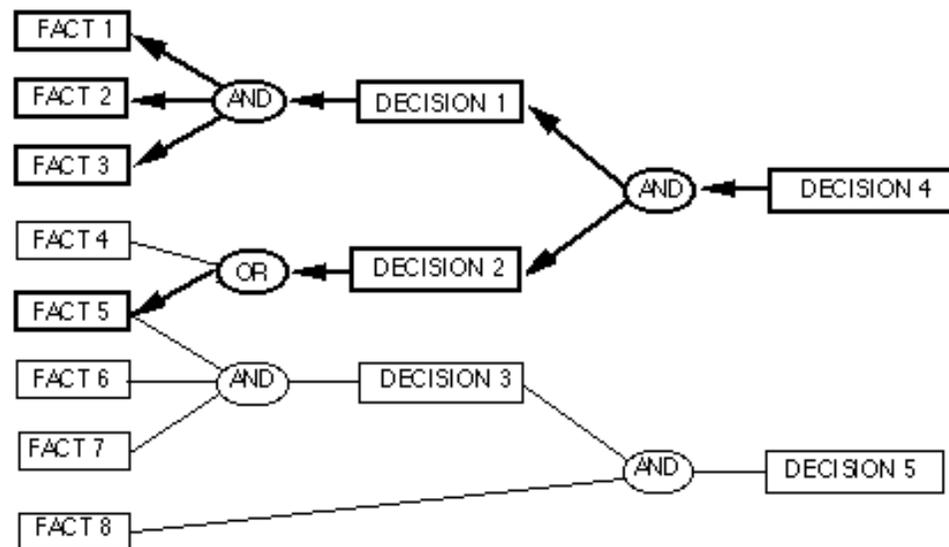
# Systeme expert

- Moteur d'inférence : *utilisation*

- *Chaînage arrière*

- *Part du **but** et **recherche** des justifications à ce but*

⇒ *Utilisé quand on veut **prendre une décision**, pour savoir si les conditions de la décision sont satisfaites*



# Systeme expert

- Moteur d'inférence : *utilisation*
  - *Choix de la stratégie de raisonnement*
    - *Si l'on cherche à connaître un but (conclusion) en particulier :*
      - *Chaînage arrière* ⇒ *pas de règle déclenchée pour rien*
      - *Chaînage avant* ⇒ *déclencherait toutes les règles possibles, sans se focaliser sur le but à démontrer*
    - *S'il y a beaucoup de manières de démontrer un fait :*
      - *Chaînage arrière* ⇒ *risque d'être trop coûteux (trop de règles)*
      - *Chaînage avant* ⇒ *si de nombreuses règles amènent à la conclusion recherchée, plus rapide à exécuter*
- ⇒ **Chaînage avant** : *si beaucoup de conclusions et/ou un petit nombre de faits initiaux*
- ⇒ **Chaînage arrière** : *si peu de conclusions à prouver et/ou que la BF initiale contient de nombreux faits*

# Diagnostic et décision

- **Systemes experts**
  - Représentation de la connaissance
    - Logique formelle
    - Base de faits
    - Base de règles
  - Le moteur d'inférence
    - Chaînage avant
    - Chaînage arrière
    - Utilisation
  - **Algorithme RETE**