



Le Mans Université

# Logiques et moteurs d'inférences

## Représentation et modélisation des Connaissances

Valérie Renault

20 septembre 2017



SBRP : Systèmes à Base de Règles de Production

Les logiques

Moteurs d'inférences des SBRP

Exercice



Comprendre comment une machine peut "**raisonner**"...

- ▶ Comprendre l'architecture de type d'un **Système à Base de Connaissances** (SBC) ;
- ▶ Savoir identifier ce qui relève du **moteur de raisonnement/moteur d'inférence**, de la **base de connaissances** qu'il manipule, les **règles** reproduisant le mécanisme de réflexion, et des **faits** relatifs à un problème donné.
- ▶ Connaître les différentes **logiques** permettant de représenter des connaissances ;

# Qu'est-ce qu'un système expert ?



## Définition :

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

Programme d'Intelligence Artificielle

Système à Base de Connaissances

Système Expert

**IA symbolique** : application  
d'heuristiques  
(pas d'algorithmes exacts disponibles)

**SBC** : connaissances explicites  
(programmation déclarative)  
+ séparation des connaissances  
du reste du système

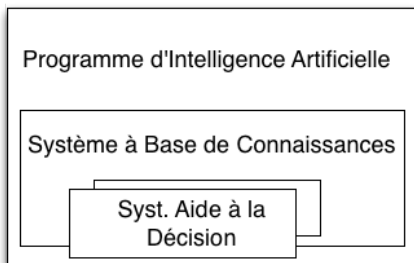
**SE** : applique les connaissances  
d'un expert à un problème réel

# Qu'est-ce qu'un système expert ?



## Définition :

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



**IA symbolique** : application d'heuristiques  
(pas d'algorithmes exacts disponibles)

**SBC** : connaissances explicites  
(programmation déclarative)  
+ séparation des connaissances du reste du système

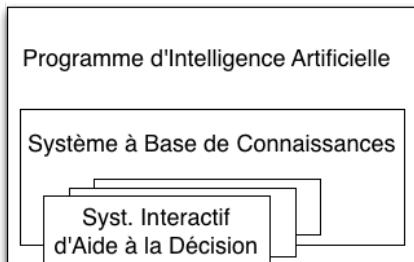
**SE** : applique les connaissances d'un expert à un problème réel

# Qu'est-ce qu'un système expert ?



## Définition :

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



**IA symbolique** : application d'heuristiques  
(pas d'algorithmes exacts disponibles)

**SBC** : connaissances explicites  
(programmation déclarative)  
+ séparation des connaissances du reste du système

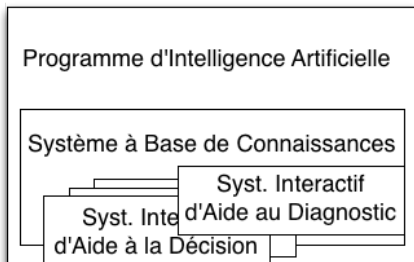
**SE** : applique les connaissances d'un expert à un problème réel

# Qu'est-ce qu'un système expert ?



## Définition :

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



**IA symbolique** : application d'heuristiques  
(pas d'algorithmes exacts disponibles)

**SBC** : connaissances explicites  
(programmation déclarative)  
+ séparation des connaissances du reste du système

**SE** : applique les connaissances d'un expert à un problème réel



- ▶ Aide au diagnostic médical (Mycin, diagnostic du diabète) ;
- ▶ Aide à la réparation de voitures et d'ordinateurs ;
- ▶ Systèmes de diagnostics de panne (centrales nucléaires) ;
- ▶ Systèmes de planification ;
- ▶ Assurances : systèmes capables d'analyser automatiquement des constats d'assurances ;
- ▶ Banques : régulations d'échanges boursiers ;
- ▶ ...

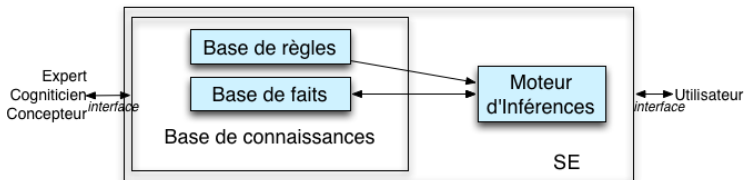
=> tous les problèmes d'**aide** à la décision



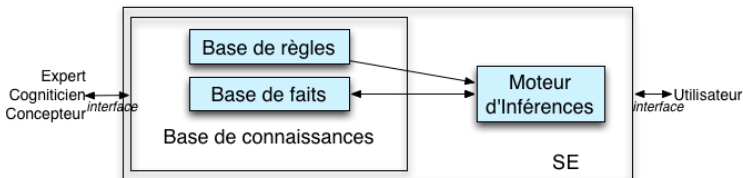


## Remarque :

Un système expert n'apportera jamais une réponse dont la logique n'a pas été prévue en amont lors de la programmation... ce qui ne veut pas dire qu'un système expert ne soit pas capable de produire une réponse surprenante ou inédite à partir de ces mécanismes.



- **Base de règles** : connaissances 'permanentes' du domaine, fournies par l'expert ;
- **Base de faits** : une expertise donnée (mémoire de travail) = les données initiales d'un problème à traiter ;
- **Moteur d'inférences** : mécanisme de 'raisonnement' du système :



## Moteurs disponibles :

- ▶ CLIPS : <http://clipsrules.sourceforge.net>
- ▶ JESS : <http://www.jessrules.com>
- ▶ PROLOG : <http://www.swi-prolog.org>



## Définition

La base de connaissances est définie à partir du **domaine** ciblé, elle rassemble toutes les connaissances utilisées par un **expert** du domaine en question pour résoudre les questions qui lui sont posées.

- ▶ Description des objets ou concepts et leurs relations ;
- ▶ Recensement des cas particuliers et des exceptions ;
- ▶ Stratégies de résolution ;
- ▶ Conflits d'application, etc.

# La base de connaissances : les faits



Les **faits** : énoncés de base, utilisés pour décrire des situations de problème et les informations déduites.

**<attribut><valeur>**

Boutons FAUX ; température 38

# La base de connaissances : les faits



Les **faits** : énoncés de base, utilisés pour décrire des situations de problème et les informations déduites.

**<attribut><valeur>**

Boutons FAUX ; température 38

**<attribut> <objet> <valeur>**

couleur voiture verte

# La base de connaissances : les faits



Les **faits** : énoncés de base, utilisés pour décrire des situations de problème et les informations déduites.

**<attribut><valeur>**

Boutons FAUX ; température 38

**<attribut> <objet> <valeur>**

couleur voiture verte

**<attribut> <objet> <valeur> <coef de vraisemb.>**

(MYCIN) identité organisme1 E-Coli 0,6

# La base de connaissances : les faits



Les **faits** : énoncés de base, utilisés pour décrire des situations de problème et les informations déduites.

**<attribut><valeur>**

Boutons FAUX ; température 38

**<attribut> <objet> <valeur>**

couleur voiture verte

**<attribut> <objet> <valeur> <coef de vraisemb.>**

(MYCIN) identité organisme1 E-Coli 0,6

**<objet> <relation> <valeur>**

X1 valeur 10 ; X1 rang 5 ;



# La base de connaissances : les faits



Les **faits** : énoncés de base, utilisés pour décrire des situations de problème et les informations déduites.

<attribut><valeur>

Boutons FAUX ; température 38

<attribut> <objet> <valeur>

couleur voiture verte

<attribut> <objet> <valeur> <coef de vraisemb.>

(MYCIN) identité organisme1 E-Coli 0,6

<objet> <relation> <valeur>

X1 valeur 10 ; X1 rang 5 ;

<prédicats> <arguments>

rouge(voiture1)



## Valeur des faits

- **Connue** : Valeur attribuée ;



## Valeur des faits

- ▶ **Connue** : Valeur attribuée ;
- ▶ **Inconnue** : Pas de valeur attribuée - Aucune question sur le sujet ;



## Valeur des faits

- ▶ **Connue** : Valeur attribuée ;
- ▶ **Inconnue** : Pas de valeur attribuée - Aucune question sur le sujet ;
- ▶ **Indéterminée** : Aucune valeur attribuée et impossible d'établir une valeur ;



## Valeur des faits

- ▶ **Connue** : Valeur attribuée ;
- ▶ **Inconnue** : Pas de valeur attribuée - Aucune question sur le sujet ;
- ▶ **Indéterminée** : Aucune valeur attribuée et impossible d'établir une valeur ;
- ▶ **Déductible** : Fait pouvant être déduit de la BF et de BR



## Valeur des faits

- ▶ **Connue** : Valeur attribuée ;
- ▶ **Inconnue** : Pas de valeur attribuée - Aucune question sur le sujet ;
- ▶ **Indéterminée** : Aucune valeur attribuée et impossible d'établir une valeur ;
- ▶ **Déductible** : Fait pouvant être déduit de la BF et de BR
- ▶ **'Demandable'** : Peut faire l'objet d'une question à l'utilisateur



## Valeur des faits

- ▶ **Connue** : Valeur attribuée ;
- ▶ **Inconnue** : Pas de valeur attribuée - Aucune question sur le sujet ;
- ▶ **Indéterminée** : Aucune valeur attribuée et impossible d'établir une valeur ;
- ▶ **Déductible** : Fait pouvant être déduit de la BF et de BR
- ▶ **'Demandable'** : Peut faire l'objet d'une question à l'utilisateur
- ▶ **Affichable** : Le fait doit être signalé à l'utilisateur



Jean et Alain sont deux personnages dont l'humeur est régie par ce principe général assez réaliste :

« Jean et Alain sont de bonne humeur s'ils ont de l'argent et s'ils sont en vacance au soleil, ou bien s'ils réussissent à la fois dans le travail et dans leurs familles respectives ».

Par ailleurs, on sait que :

- ▶ Jean et Alain ont tout deux de l'argent. Jean et Alain réussissent dans leur travail.
- ▶ Jean part en vacances en août et Alain en juillet. Il y a du soleil en août mais on est en juillet.
- ▶ Alain réussit dans sa famille

Question : qui est heureux ?

**Quels sont les faits ?**

attribut ? objet ? valeur ? argument ? déductible ?





Ces connaissances sont utilisées par le système expert au moyen de règles reproduisant le **mécanisme de réflexion** de l'expert.

Ces règles ne doivent pas être complexes en elles-mêmes, mais au contraire le plus simple possible : il s'agit de **décomposer le raisonnement global** en un nombre maximum de sous-raisonnements logiques, qui serviront de «briques» pour reconstituer une multitude de raisonnements globalement complexes.



Une règle : un couple ayant une "partie gauche" et une "partie droite"

si <...> et ... et <...>	alors <...>et ... et <...>
condition	conclusion
prémisse	conséquence
antécédent	action
<i>détermine l'applicabilité de la règle</i>	<i>décrit l'action à accomplir si la règle s'applique</i>



Différents formalismes pour les <prémisses>

**NON(<attribut boolean>)**

NON(Boutons FAUX) ; NON (Acheter-un-billet VRAI) ;



Différents formalismes pour les <prémisses>

**NON(<attribut boolean>)**

NON(Boutons FAUX) ; NON (Acheter-un-billet VRAI) ;

**INCONNU <attribut>**

INCONNU(distanceA-B)



Différents formalismes pour les <prémisses>

**NON(<attribut boolean>)**

NON(Boutons FAUX) ; NON (Acheter-un-billet VRAI) ;

**INCONNU <attribut>**

INCONNU(distanceA-B)

**<attribut> <objet> <comparateur> <valeur>**

prix voiture < 30 000



Différents formalismes pour les <prémisses>

**NON(<attribut boolean>)**

NON(Boutons FAUX) ; NON (Acheter-un-billet VRAI) ;

**INCONNU <attribut>**

INCONNU(distanceA-B)

**<attribut> <objet> <comparateur> <valeur>**

prix voiture < 30 000

**<attribut> <variable> <comparateur> <valeur>**

prix X <200 avec X un objet quelconque

...



Différents formalismes pour les <conclusions>

<attribut booléen>



## Différents formalismes pour les <conclusions>

<attribut booléen>

NON <attribut booléen>





## Différents formalismes pour les <conclusions>

<attribut booléen>

NON <attribut booléen>

<attribut>  $\Leftarrow$  <valeur>



## Différents formalismes pour les <conclusions>

<attribut booléen>

NON <attribut booléen>

<attribut>  $\Leftarrow$  <valeur>

<attribut> <objet>  $\Leftarrow$  <valeur>



## Différents formalismes pour les <conclusions>

<attribut booléen>

NON <attribut booléen>

<attribut>  $\Leftarrow$  <valeur>

<attribut> <objet>  $\Leftarrow$  <valeur>

<action>



## Différents formalismes pour les <conclusions>

<attribut booléen>

NON <attribut booléen>

<attribut>  $\Leftarrow$  <valeur>

<attribut> <objet>  $\Leftarrow$  <valeur>

<action>

<predicat> <argument>



- ▶ R1 : `est_de_bonne_humeur(X)` si `a_de_l_argent(X)` et `est_en_vacances(X)` et `il_y_a_du_soleil`.
- ▶ R2 : ...
- ▶ R3 : `a_de_l_argent(jean)`.



## Un langage

- ▶ **Une syntaxe** : ensemble de symboles et de règles ;
- ▶ **Une sémantique** : donne un sens aux symboles et aux formules ;

## Proposition : unité d'information élémentaire

- ▶ Une proposition peut être vraie ou fausse, mais elle est toujours correcte syntaxiquement et sémantiquement ;
- ▶ **Un merle est noir. Une pie est rouge.**

# Proposition = Formule bien formée



Une FBF comporte :

**Instances** : représentées par des symboles ou des lettres minuscules

Jacques, Vert, Bloc, a, b, c, ...

**Prédicats** : symboles prenant un nombre fixe d'arguments (0-n)

il-pleut, Etudiant(Jacques), Age(Jacques, 25)

**Fonctions** : retournant les propriétés d'un symbole

Oncle(Jacques) :Pierre, Age(Jacques) :25



- ▶  $\text{P\grave{e}se}(\text{Jacques}, 78)$
- ▶  $\neg \text{Ecrivain}(\text{Mozart})$
- ▶  $\text{Cousin}(\text{Fils}(\text{Jean}), \text{Neveu}(\text{Jean}))$
- ▶  $\text{Lit}(\text{Pierre}, \text{Candide}) \wedge \text{Auteur}(\text{Voltaire}, \text{Candide})$
- ▶  $\text{Humain}(\text{Socrate}) \Rightarrow \text{Mortel}(\text{Socrate})$





On peut donner une valeur à une FBF par rapport à une interprétation donnée, en utilisant les tables de vérité des différents connecteurs :

- ▶ ou :  $\vee$
- ▶ et :  $\wedge$
- ▶ not :  $\neg$
- ▶ implication :  $\Rightarrow$

Il existe aussi des quantificateurs :

- ▶ il existe :  $\exists$
- ▶ quelque soit :  $\forall$



- ▶  $\neg(\neg X)$  est équivalent à  $X$
- ▶ Lois associatives
  - ▶  $(X \vee Y) \vee Z$  équivalent à  $X \vee (Y \vee Z)$
  - ▶  $(X \wedge Y) \wedge Z$  équivalent à  $X \wedge (Y \wedge Z)$
- ▶ Lois commutatives
  - ▶  $X \vee Y$  équivalent à  $Y \vee X$
  - ▶  $X \wedge Y$  équivalent à  $Y \wedge X$
- ▶ Lois de distribution
  - ▶  $X \vee (Y \wedge Z)$  équivalent à  $(X \vee Y) \wedge (X \vee Z)$
  - ▶  $X \wedge (Y \vee Z)$  équivalent à  $(X \wedge Y) \vee (X \wedge Z)$



- ▶  $(X \vee Y)$  équivalent à  $\neg X \Rightarrow Y$
- ▶ Loi de Morgan
  - ▶  $\neg (X \wedge Y)$  équivalent à  $\neg X \vee \neg Y$
  - ▶  $\neg (X \vee Y)$  équivalent à  $\neg X \wedge \neg Y$
- ▶ Loi de contraposée
  - ▶  $X \Rightarrow Y$  équivalent à  $\neg Y \Rightarrow \neg X$

## Exemple : Loi de Morgan

$\neg (\text{il-pleut} \wedge \text{je-prends-mon-parapluie})$  équivalent à  
 $\neg \text{il-pleut} \vee \neg \text{je-prends-mon-parapluie}$



- ▶ **Valide** : vraie dans toute interprétation (tautologie) ;
- ▶ **Inconsistante** : fausse dans toutes les interprétations ;
- ▶ **Consistante** : il existe une interprétation pour laquelle elle est vraie ;



Règles sans variable :

- ▶ Ordre 0 - logique des propositions
- ▶ Ordre 0+ - logique des propositions étendue
- ▶ Ordre 1/2

Règles avec variables :

- ▶ **Logique des prédicats du 1er ordre**
- ▶ Ordre > 1

"Les" logiques :

- ▶ La logique floue : Rajout de 'nuances' : la vitesse du vent sera *un peu* élevée ;
- ▶ Logique monotone : Ajout de faits uniquement ;
- ▶ Logique non monotone : Ajout, modification, retrait ;
- ▶ ...



Les symboles sont des booléens :

## Propositions :

"possède\_appartement", "crise\_financiere"  
"immobilier\_valeur\_refuge", "être\_riche"

## Règles :

SI "crise\_financiere" ALORS "immobilier\_valeur\_refuge"

## Faits initiaux :

"crise\_financiere"



Introduction d'attributs numériques et de comparateurs



Introduction d'attributs numériques et de comparateurs

## Attributs (type)

possède(énuméré), crise\_financière(boolean),  
valeur\_refuge(énuméré)





Introduction d'attributs numériques et de comparateurs

## Attributs (type)

possède(énuméré), crise\_financière(boolean),  
valeur\_refuge(énuméré)

## Règles

Si (crise\_financière = oui) Alors (valeur\_refuge = immobilier)



Introduction d'attributs numériques et de comparateurs

## Attributs (type)

possède(énuméré), crise\_financière(boolean),  
valeur\_refuge(énuméré)

## Règles

Si (crise\_financière = oui) Alors (valeur\_refuge = immobilier)

## Condition terminale

situation\_financière



Introduction de variables : ?var



Introduction de variables : ?var

## Prédicats (fonctions logiques) :

possède( ?individu, ?chose), crise\_financiere(),  
situation\_financiere( ?individu, ?situation)



Introduction de variables : ?var

## Prédicats (fonctions logiques) :

possède( ?individu, ?chose), crise\_financiere(),  
situation\_financiere( ?individu, ?situation)

## Règles :

SI crise\_financiere ALORS valeur\_refuge(immobilier) ;  
SI valeur\_refuge(immobilier) & possede( ?l, appartement)  
ALORS situation\_financiere( ?l, riche)



Introduction de variables : ?var

## Prédicats (fonctions logiques) :

possède( ?individu, ?chose), crise\_financiere(),  
situation\_financiere( ?individu, ?situation)

## Règles :

SI crise\_financiere ALORS valeur\_refuge(immobilier) ;  
SI valeur\_refuge(immobilier) & possede( ?l, appartement)  
ALORS situation\_financiere( ?l, riche)

## Faits initiaux :

crise\_financiere() ; possède(Jean, appartement)



Introduction de variables : ?var

## Prédicats (fonctions logiques) :

possède( ?individu, ?chose), crise\_financiere(),  
situation\_financiere( ?individu, ?situation)

## Règles :

SI crise\_financiere ALORS valeur\_refuge(immobilier) ;  
SI valeur\_refuge(immobilier) & possède( ?I, appartement)  
ALORS situation\_financiere( ?I, riche)

## Faits initiaux :

crise\_financiere() ; possède(Jean, appartement)

## Condition terminale :

situation\_financiere( ?I, ?S)



- ▶ Une **règle d'inférence** (RI) est un procédé pour dériver à partir d'un ensemble de FBF donnés (les axiomes) un ensemble d'autres FBF (les théorèmes).
- ▶ Une chaîne de dérivation s'appelle une preuve.
- ▶ Exemples de règles d'inférences :
  - ▶ **Modus Ponens** : De  $G$  et  $(G \Rightarrow H)$  on dérive  $H$ .
  - ▶ **Modus Tollens** : De  $\neg H$  et  $(G \Rightarrow H)$  on dérive  $\neg G$ .





La logique des prédicats du 1er ordre est indécidable :

## Théorème de Church

Il n'y a pas d'algorithme permettant de décider en un nombre fini d'opérations qu'une formule quelconque est valide ou invalide

Mais la logique des prédicats du 1er ordre est semi-décidable :  
décidable pour certains types de FBF.



D'où :

- ▶ Utilisation de la logique des prédicats du 1er ordre pour formaliser les problèmes en un ensemble de FBF ;
- ▶ Puis, transformation de cet ensemble de FBF en un autre ensemble de FBF sur lesquelles on va pouvoir appliquer des algorithmes de démonstration automatique.

## Principe de Résolution par Réfutation - Principe de Robinson

On va devoir prouver qu'un ensemble de clauses est inconsistant afin de valider une hypothèse.



## Définition

Une clause de Horn possède un et un seul littéral positif.

Exemples :

- ▶ femme(victoria)
- ▶ homme(edward)
- ▶  $\text{soeur}(X,Y) \vee \neg \text{femme}(X) \vee \neg \text{parents}(X, \text{Pere}, \text{Mere}) \vee \neg \text{parents}(Y, \text{Pere}, \text{Mere})$

Forme générale (en écriture Prolog) :

$F :- F1, F2, \dots, F_n$

si  $F1, F2$  et  $F_n$  sont vraies alors le tête est aussi vraie.



Ecritures logiques	Clauses de Horn	Ecritures Prolog
$\Rightarrow D$	$D$	$D.$
$A \Rightarrow$	$\neg A$	$:-A.$
$A \wedge B \Rightarrow$	$\neg A \vee \neg B$	$:-A,B.$
$A \wedge B \Rightarrow C$	$\neg A \vee \neg B \vee C$	$C :-A,B.$



## Ecritures logiques :

$$\text{homme}(X) \wedge \text{parents}(X, \text{Mere}, \text{Pere}) \wedge \text{parents}(Y, \text{Mere}, \text{Pere}) \Rightarrow \text{frere}(X, Y)$$

## Clauses de Horn :

$$\text{frere}(X, Y) \vee \neg \text{homme}(X) \vee \neg \text{parents}(X, \text{Mere}, \text{Pere}) \vee \neg \text{parents}(Y, \text{Mere}, \text{Pere})$$

## Ecritures Prolog :

$$\text{frere}(X, Y) \text{ :- homme}(X), \text{parents}(X, \text{Mere}, \text{Pere}) \text{ parents}(Y, \text{Mere}, \text{Pere}).$$



C'est un mécanisme qui permet d'inférer des connaissances nouvelles à partir de la base de connaissances du système.

Phase **évaluation** :

- ▶ **Sélection** (ou restriction) : privilégier tel groupe de règles BR1 ou de faits BF1 dans la BC (BR, BF) ;
- ▶ **Filtrage** (ou pattern-matching) : comparer les déclencheurs de règles BR2 par rapport aux faits BF1. BR2 : l'ensemble des conflits ;
- ▶ **Résolution des conflits** (ou matching) : déterminer les règles BR2 qui doivent être effectivement déclenchées ;

Phase **exécution** des règles : application de la règle choisie.



- ▶ Règles simplificatrices au début ;
- ▶ Règles expansives (qui augmentent la longueur de la formule) à la fin ;
- ▶ Règles qui s'appliquent souvent au début ou à la fin ;
- ▶ ...

Les règles que l'on veut appliquer en dernier doivent être rejetées à la fin de la base



## Algorithme Chaînage Avant (BF, BR, Fait)

Début

Tant que Fait n'est pas dans BF  
et qu'il existe dans BR une règle applicable

Faire:

Choisir une règle applicable R (heuristiques)

$BR = BR - R$  /\* désactivation de R\*/

$BF = BF \cup \text{conclusion}(R)$

Fin tant que

Si F appartient à BF Alors F est établi

Sinon F n'est pas établi

Fin





Soit BR :

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

exemple 1 : BF initiale  $\{H, K\}$

exemple 2 : BF initiale  $\{G, F, H, K\}$



- ▶ Il s'arrête toujours.
- ▶ Si on utilise des règles dont les conclusions peuvent être des faits négatifs, pour tout fait  $F$ , il peut se produire 4 situations :
  - ▶  $F \in BF$  : le fait est établi ;
  - ▶  $\neg F \in BF$  : la négation du fait est établie ;
  - ▶ ni  $F$  ni  $\neg F$  ne sont dans la  $BF$  : le système ne peut rien déduire à partir de ce fait - l'interprétation peut être diverse ;
  - ▶  $\neg F$  et  $F \in BF$  : la base de fait est incohérente. Pour cela, on peut prévoir un fait : BaseFait-Incohérente et une Méta-Règle :  
si  $\exists F$  tq  $F \in BF$  et  $\neg F \in BF$  alors BaseFait-Incohérente



- ▶ Déclenche toutes les règles ;
- ▶ Demande beaucoup de faits initiaux ;
- ▶ Explosion combinatoire possible ;



## Algorithme Chaînage Arrière(BF, BR, Fait)

Début

- Phase de filtrage
- Si ensemble des règles sélectionnées est vide  
alors questionner utilisateur
- Sinon :
  - Tant que F n'est pas dans la BF  
et qu'il reste des règles sélectionnées
  - Faire
    - Phase de choix = résolution des conflits
    - Ajouter les sous-buts correspondant à la partie  
gauche de la règle élue
    - Si un sous-but n'est pas résolu Alors le résoudre

Fin

Fin



Soit BR :

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

exemple 1 : BF initiale  $\{H, K\}$

exemple 2 : BF initiale  $\{G, F, H, K\}$

But : D vrai ?



- ▶ Algorithme un peu plus compliqué (construction d'un arbre ET/OU) ;
- ▶ Système interactif à la demande (distinguer les faits demandables et les faits non demandables) ;
- ▶ Arbre de recherche est plus petit ;
- ▶ Risque de bouclage ;



## Algorithme Chaînage Mixte(BF, BR, Fait)

Début

Tant que F n'est pas déduit mais peut encore l'être

Faire

- saturer la base de faits par chaînage avant
- chercher quels sont les faits encore éventuellement déductibles
- déterminer une question pertinente à poser à l'utilisateur et ajouter sa réponse à la BF

Fin Tant que

Fin



- ▶ **Monde clos** : en absence de fait, on en déduit son contraire (interprétation de la négation par l'absence) ;
- ▶ **Monde ouvert** : on ne tient compte que des faits explicitement connus (déduits, donnés) ;

## Monde clos

Si maux-de-tête  $\wedge$  frisons  $\Rightarrow$  Etat-grippal

par défaut on suppose  $\neg$ vomissement

Si maux-de-tête  $\wedge$  frisons  $\wedge$  vomissement  $\Rightarrow$  Gastro





Les langages qui considèrent le calcul en tant que déduction dans un formalisme logique :

- ▶ la programmation logique : PROLOG, la programmation l'emporte sur la logique (emploi du **is** arithémique) ;
- ▶ la programmation logique **avec contraintes** : accéder à des domaines **numériques** ;
- ▶ la programmation **par ensembles réponses (ASP - answer set programming)** : la logique **non monotone** l'emporte sur l'aspect programmation ;

## Un programme

```
entree(crudites).  
entree(terrine).  
entree(melon).  
viande(steack).  
viande(poulet).  
viande(gigot).  
poisson(bar).  
poisson(saumon). dessert(sorbet).  
dessert(creme).  
dessert(tarte).  
  
menu_simple(E, P, D) :- entree(E), plat(P), dessert(D).  
plat(P) :- viande(P).  
plat(P) :- poisson(P).
```

## Des requêtes : construction d'un arbre de résolution

```
?- poisson(P), menu_simple(melon, P, D).  
?- menu_simple(melon, P, D), poisson(P).
```



Nemo est un gentil petit poisson et il vit dans le Pacifique.  
Tous les requins aiment les petits poissons.  
Si Shark, le requin, a faim alors il mangera les petits poissons.  
Aucun poisson clown ne s'éloigne de sa maison.  
Quelques poissons clowns aiment le pain.

- ▶ Formalisez les énoncés suivants en logique des prédicats du 1er ordre.
- ▶ Exprimez, quand c'est possible, les connaissances sous forme de clauses de Horn.
- ▶ Exprimez, quand c'est possible, les connaissances en Prolog.



[Marquis 2014] Pierre Marquis, Odile Papini, Henry Prade, "Panorama de l'Intelligence Artificielle, ses bases méthodologiques, ses développements", Cépaduès-Éditions - 2014

Volume 1 : "Représentation des connaissances et formalisation des raisonnements"

Volume 2 : "Algorithmes pour l'intelligence artificielle"

Volume 3 : "L'intelligence artificielle, frontières et applications"

[Laurière 1987] Jean-Louis Laurière, "Intelligence artificielle : résolution de problème par l'homme et la machine", Eyrolles, 1987.

[Laurière 1987] Jean-Louis Laurière, "Intelligence artificielle, tome 2, Représentation des connaissances", Eyrolles, 1988.

[Russel 2003] Stuart Russell, Peter Norvig, "Artificial Intelligence, A modern Approach" Prentice Hall Series, 2003.

<http://www.cin.ufpe.br/~tfl2/artificial-intelligence-modern-approach.9780131038059.25368.pdf>