

Ch3 : Système Expert

Prof. Konan Marcellin BROU

marcellin.brou@inphb.ci

2019-2020

Sommaire

- ❑ **Introduction**
- ❑ **Architecture d'un SE**
- ❑ **Représentation de la connaissance**
- ❑ **Fonctionnement du MI**
- ❑ **Stratégie de recherche**
- ❑ **Mode d'invocation des règles**
- ❑ **Générateur de systèmes experts**

I. Introduction

□ Objectifs

- **Maîtriser les concepts de base sur les systèmes experts**
- **Comprendre le mode d'invocation des règles d'inférences**
- **Savoir créer un système expert en utilisant un générateur de système expert**

I. Introduction

□ 1.1. Rêve de l'homme

■ Créer un automate à son image :

□ Construire des machines capables de se substituer à l'homme

- Pour effectuer certaines tâches "intellectuelles" répétitives ou fastidieuses.

□ Créer une intelligence "artificielle"

□ Quelques exemples :

I. Introduction

■ Exemple 1 : le Golem

- Mythologie juive
- hébreu : "embryon", "informe" ou "inachevé"

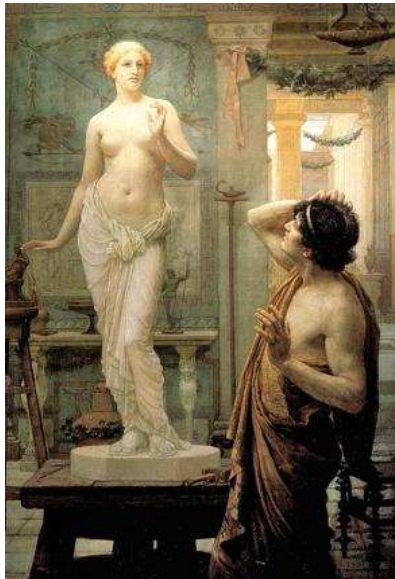


- Sorte de robot à figure humaine, fait de bois ou pétri avec de l'argile à l'imitation de Dieu créant l'homme.
 - Il ressemblait à un enfant ou à un petit homme.
 - Animé par les saints rabbins juifs

- Muet et dépourvu de libre-arbitre façonné afin d'assister ou défendre son créateur.
- Animation : inscrire sur son front le mot hébreu emet (vérité).
 - Il devenait alors un serviteur docile et muet.
- Destruction : enlever la 1ère lettre e du mot emet.
 - Il reste alors le mot met signifiant mort.

I. Introduction

- **Exemple 2 : le pygmalion**
 - Mythologie grecque
 - Pygmalion : roi sculpteur chypriote de l'antiquité.



INP-HB/K. M. BROU

- Misogyne : aucune qualité des femmes ne lui convient.
- Sculpta une statuette en ivoire représentant une femme réunissant tous ses critères de beauté appelé Galatée.
- Il demanda aux Dieux de la transformer en une femme.
- Vœux exaucé par Aphrodite : déesse grecque de la Beauté, de l'Amour, du Plaisir et de la Procréation.
- Il épousa Galatée en présence d'Aphrodite.
- Ils eurent un fils appelé Paphos et une fille du nom de Métharmè.

I. Introduction

■ Exemple 3 : les calculateurs

- **Le boulier** : première machine à calculer, inventé par les chinois Vers 3000 avant J.C.



- C'est un abaque (outil servant à calculer) formé d'un cadre rectangulaire muni de tiges sur lesquelles couissent des boules.

- Il est lié au système de numération décimale (base 10).

- chaque boule représente, selon la tige sur laquelle elle se trouve, une unité, une dizaine, une centaine....

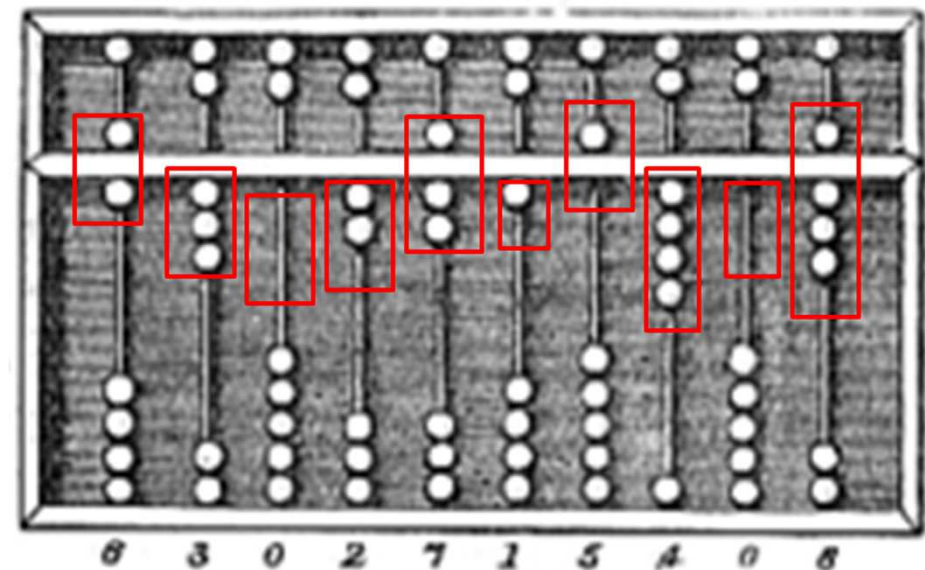
- Il permet d'effectuer des opérations élémentaires : additions, soustractions, multiplications et divisions.

I. Introduction

- **Chaque colonne représente en partant de la droite :**
 - unités, dizaines, centaines etc.
 - On ne prend en compte dans le calcul du nombre représenté que les boules activées, c'est-à-dire déplacées près de la barre centrale horizontale.
- **Les 5 boules en dessous de la barre valent chacune un,**
- **les 2 boules situées au-dessus de la barre valent chacune cinq.**

- **Exemple : Représentation du nombre : 6 3 0 2 7 1 5 4 0 8**

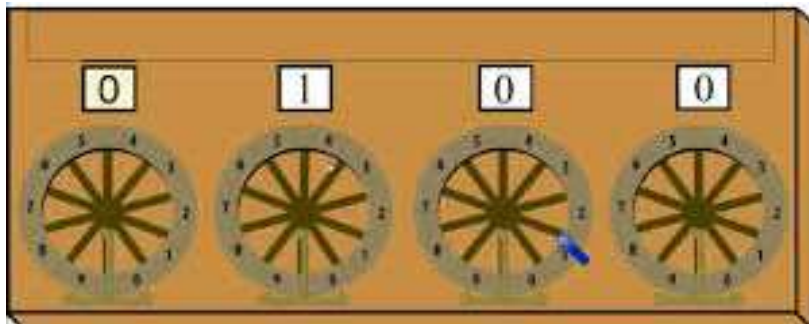
- On compte la valeur représentée par les boules dans chaque colonne.



I. Introduction

■ La machine de Pascal : la "Pascaline"

- En 1642 Blaise Pascal invente, à l'âge de 19 ans.



INP-HB/K. M. BROU

IA : Système expert

- Cette machine ne peut effectuer que des additions et soustractions.
- Les roues dentées qui la constituent comportent 10 positions (0 à 9).
A chaque fois qu'une roue passe de la position 9 à la position 0, la roue immédiatement à sa gauche, avance d'une position.
- En 1673 Gotfried Leibniz perfectionne la Pascaline, il ajoute les multiplications, les divisions et les racines carrées.

I. Introduction

□ **L'Ordinateur :**

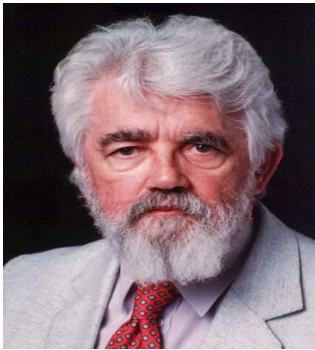
- Ordinateur : Machine électronique programmable qui permet le traitement de l'information.
- Baptisé souvent cerveau électronique.
- Commence avec les grands développements de la seconde guerre mondiale.
- Le calcul numérique remplace le calcul analogique.
- 1920 Enigma est construit par les allemands pour déchiffrer des messages pendant la seconde guerre mondiale.



I. Introduction

□ 1.2. Naissance de l'IA

- 1956 : université d'été à Dartmouth College (USA)
 - **Organisateur** : John Mc Carthy propose de créer une nouvelle discipline IA.



- **But** : reproduire des comportements intelligents à l'aide d'une machine.

- **Participants** : logiciens, électroniciens, psychologues, cybernéticiens, économistes.

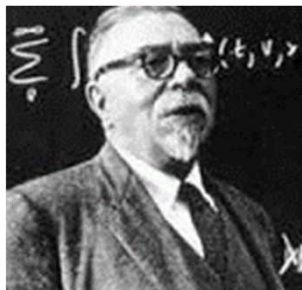
■ Pour lui :

- "Le but de l'IA est l'étude de la structure de l'information et la structure de processus de résolution de problèmes, indépendamment des applications et indépendamment d'une réalisation."

I. Introduction

■ Cybernétique

- Science permettant à un homme ou à une machine automatique (possédant un SI et un pseudo-cerveau) de gouverner (i.e. diriger, prendre des décisions).
- Fondée en 1948 par le mathématicien américain Norbert Wiener.



- C'est aussi une science du contrôle des systèmes, vivants ou non-vivants.
 - Notre monde est constitué de systèmes, vivants ou non-vivants, imbriqués et en interaction.
- Un système est un ensemble d'éléments matériels ou immatériels (homme, machine, méthode...) en interaction dynamique, organisés en fonction d'un but.

I. Introduction

- ▣ **Exemples de systèmes :**

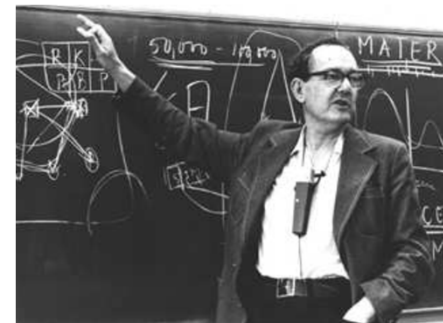
- une société, une économie, un réseau d'ordinateurs, une machine, une entreprise, une cellule, un organisme, un cerveau, un individu.

- ▣ **Les ordinateurs et toutes les machines intelligentes actuelles sont des applications de la cybernétique.**

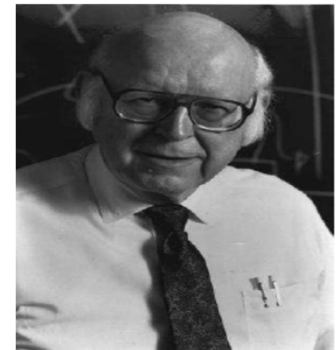
I. Introduction

- **Logic Theorist : 1^{er} programme de démonstration de théorèmes (Alan Newell et Herber Simon)**
 - **Permet de démontrer des théorèmes mathématiques basés sur le principe du syllogisme : $A \Rightarrow B$ et $B \Rightarrow C$ alors $A \Rightarrow C$.**
 - utilisation de systèmes symboliques et d'heuristiques (est une méthode empirique de résolution de problème, dont la validité ou l'efficacité n'est pas prouvée).
 - **Il a été capable de prouver 38 des 52 théorèmes des Principia Mathematica de Whitehead et Russell.**

- **Herbert Simon obtint le prix Nobel en économie**



Herber Simon



Alan Newell

I. Introduction

□ **Logic Theorist utilise :**

- les connecteurs logiques NON (noté \neg) et OU (noté \vee).
- L'implication (notée \Rightarrow) se construit à partir d'eux car $P \Rightarrow Q$ équivaut à $\neg P \vee Q$.

□ **Logic Theorist a 5 axiomes :**

- $(P \vee P) \Rightarrow P$ (idempotence du OU)
- $P \Rightarrow (Q \vee P)$
- $(P \vee Q) \Rightarrow (Q \vee P)$ (commutativité du OU)
- $(P \vee (Q \vee R)) \Rightarrow (Q \vee (P \vee R))$ (associativité du OU)
- $(P \Rightarrow Q) \Rightarrow ((P \vee R) \Rightarrow (Q \vee R))$

□ **Deux règles d'inférences sont utilisées :**

- Règle de substitution : si $A(P)$ est une formule vraie contenant la variable P , alors quelque soit la formule B , $A(B)$ est une formule vraie.
- Règle de détachement : si A est une formule vraie et $A \Rightarrow B$ est une implication vraie, alors B est une formule vraie.

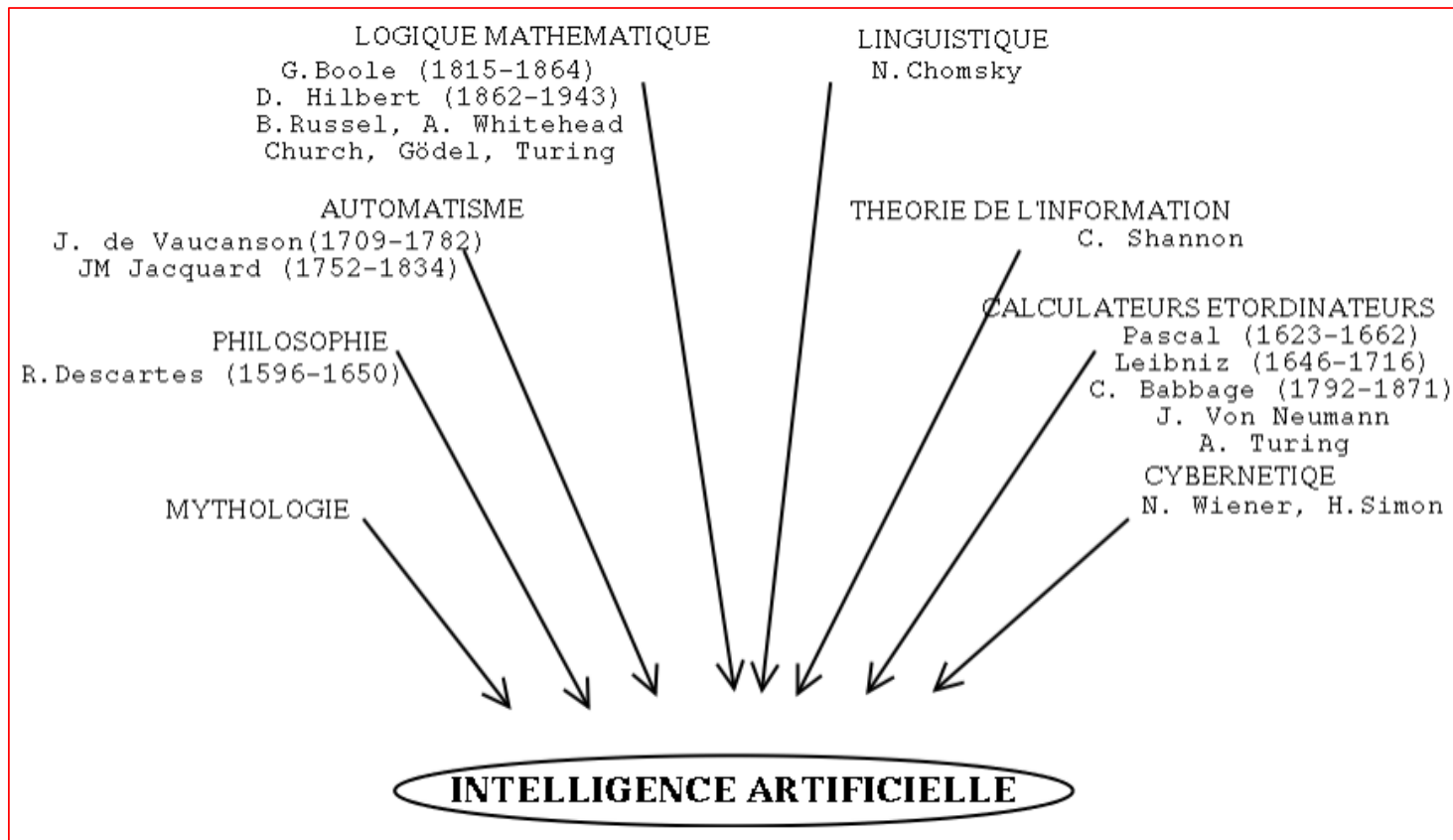
I. Introduction

■ Exemple démontrer $(P \Rightarrow \neg P) \Rightarrow \neg P$

- Il part du premier axiome : $(P \vee P) \Rightarrow P$
- Il substitue $\neg P$ à P : $(\neg P \vee \neg P) \Rightarrow \neg P$
- Il utilise la définition de l'implication : $P \Rightarrow Q$ équivaut à $\neg P \vee Q$
- **Donc $(P \Rightarrow \neg P) \Rightarrow \neg P$**

I. Introduction

■ Résumé : Origine de l'IA



I. Introduction

□ 1.3. Objectifs d'un Système Expert

- **Caractéristiques des experts d'un domaine :**
 - Rares, donc peu disponibles
 - Compétents (si possible les meilleurs)
 - Souvent incapables d'expliquer leur démarche
 - Mortels
- **Rendre une expertise accessible à tous.**
- **Approcher au mieux la perfection.**
- **Décortiquer le raisonnement expert pour l'expliquer.**
- **Rendre une connaissance experte insensible au temps.**
- **Résumé :**
 - **Simuler le comportement d'un expert humain sans avoir les défauts de la nature humaine énumérés ci-dessus.**

I. Introduction

□ 1.4. Définition

■ Plusieurs définitions possibles :

- Programme capable de mener des raisonnements intelligents et intelligibles sur des connaissances symboliques d'un domaine particulier dans un univers incertain et incomplet.
- Programme conçu pour raisonner habilement à propos de tâches dont on pense qu'elle requiert une expertise humaine considérable.
- Programme destiné à assister l'homme dans des domaines où une expertise humaine est requise.

■ Résumé

- Machine déductive relativement générale exploitant une collection séparée, sujette à évolution, d'unités de savoir-faire concernant un domaine particulier d'expertise humaine. Son but est d'apporter des solutions à des problèmes bien délimités concernant le domaine en question. Un tel programme assure aussi des fonctions complémentaires de dialogue, d'apprentissage et d'explication de son comportement.

I. Introduction

- **1.5. Terminologies équivalentes**
 - **Système déductif**
 - **Système de résolution de problèmes**
 - **Système à Base de Connaissances (SBC)**
 - **Knowledge Based System (KBS)**

I. Introduction

□ 1.6. Caractéristiques principales

- Capable de tenir des raisonnements intelligents de façon intelligible ;
- Susceptible d'accumuler des connaissances en vrac ;
- Capables d'accepter des heuristiques pour guider le raisonnement ;
- Capable de dialoguer en langage naturel.

I. Introduction

□ 1.7. Principes fondamentaux des SE

■ Connaissance donnée sous forme déclarative :

- l'expert doit pouvoir ajouter, modifier et supprimer des informations sans aucune nouvelle programmation.

■ BC indépendante du mécanisme d'exploitation :

- Ce mécanisme doit pouvoir travailler sur des connaissances exprimées selon un certain formalisme de représentation.

- Le processus d'inférence ne doit pas changer en l'utilisant dans un autre domaine.

■ Capable d'expliquer son raisonnement :

- Expliquer et justifier le raisonnement qu'il a effectué et qui lui a permis d'aboutir à une conclusion.
- Ces explications doivent correspondre à une démarche qui semble naturelle au spécialiste.

I. Introduction

■ **Apprentissage :**

□ **Enrichissement de la connaissance, deux cas possibles :**

- Enrichissement sans apprentissage : l'expert ajoute des règles à la BC.
- Enrichissement avec apprentissage : les règles sont ajoutées à la BC automatiquement à partir d'exemples. Dans le SE Meta-Dendral, un module donne de nouvelles règles à partir d'exemples (problème d'induction).

□ **Amélioration des performances du SE**

- Il s'agit de trouver des heuristiques performantes ou des méta-règles associées à certains faits de la base de faits.
- Le système LEX conçu pour calculer des primitives trouve des heuristiques à partir d'exemples traités.

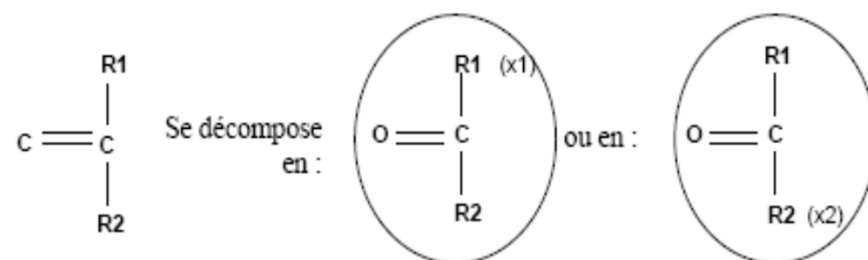
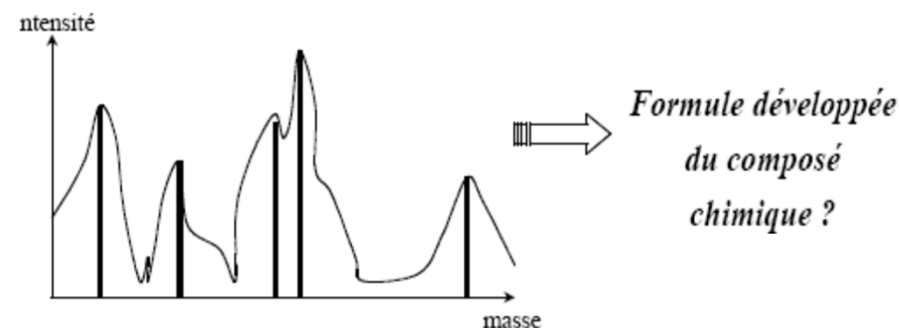
I. Introduction

□ 1.8. Quelques systèmes experts historiques

■ SE pour valider l'approche systèmes experts

□ **DENDRAL** (Université de Stamford, 1969) :

- C'est le père des systèmes experts actuel ;
- Il permet de rechercher la formule développée d'un corps organique à l'aide de la formule brute et du spectrogramme de masse du corps considéré.



I. Introduction

□ Exemples de connaissances :

▪ Règle :

Si le spectre de la molécule présente deux pics x_1 et x_2 tels que :

1. $x_1 - x_2 = M + 28$
2. $x_1 - 28$ est un pic élevé
3. $x_2 - 28$ est un pic élevé
4. au moins l'un des pics x_1 et x_2 est élevé

Alors la molécule contient un groupe cétone

□ MYCIN (Université de Stamford, 1974, USA)

- Permet d'identifier les organismes microbiens à l'origine des maladies infectieuses du sang et proposer une thérapie adaptée.
- Exemple de règle :

SI le site de culture est le sang, et que le germe est positif à la coloration de Gram, et que la porte d'entrée est l'appareil gastro-intestinal, et que (l'abdomen est le site d'infection ou le bassin est le site d'infection)

ALORS Il y a de très fortes présomptions pour que les entérobactéries soient la classe d'organismes que la thérapie devra traiter.

- Dans 90% des cas, les conclusions fournies par MYCIN furent considérées comme aussi bonnes que celles des experts.

I. Introduction

- Exemple de diagnostic

Acteur	Message
MYCIN	Mes recommandations thérapeutiques seront basées sur les infections probables et les organismes infectieux potentiels suivant : L'organisme peut être : 1° pseudomonas aeruginosa, 2° klebsiella pneumoniae, 3° escherichia coli, 4° bacteroides fragilis, 5° enterobacteriaceae, 6° proteus mirabilis Voulez-vous me donner la gravité de la maladie de M. Fred Smith sur une échelle de 0 à 4 ?
Médecin	3
MYCIN	M. Fred Smith présente-t-il des réactions allergiques à un quelconque agent microbien ?
Médecin	Non
MYCIN	Quel est son poids en kilos ?
Médecin	70

I. Introduction

- Exemple de prescription thérapeutique

Acteur	Message
MYCIN	Ma recommandation thérapeutique sera en premier choix : pour couvrir les points <1 2 3 5 6>, prescrire de la Gentamicine, dose 119 mg 98 h IV, pendant 10 jours; Pour couvrir le point 4, prescrire de la Clindamycine, dose 595 mg 96 h IV, pendant 14 jours. Désirez-vous éventuellement éliminer certains médicaments ?
Médecin	Oui, la Clindamycine.
MYCIN	Pour couvrir les points <2 3 4 5 6> donner du Chloramphénicol, dose 1 g 96h IV, pendant 14 jours; ** Surveiller le taux de globules blancs du patient. Pour couvrir le point 1, donner de la gentamicine, dose 119 mg 98 h IV, pendant 10 jours. ** Modifier la dose en cas de faiblesse rénale.

I. Introduction

- Exemple d'explication du raisonnement.

Acteur	Message
MYCIN	L'infection du malade a t-elle été contractée à l'hôpital ?
Médecin	Pourquoi ?
TEIRESIAS	<p>[Pourquoi est-ce important de savoir cela ?] Cela aidera à déterminer la catégorie à laquelle appartient l'organisme responsable de l'infection :</p> <ul style="list-style-type: none">- il a déjà été établi que 1° l'organisme est à Gram négatif, 2° c'est un bâtonnet, 3° il se développe à l'air ou sans air.- Donc si l'infection par cet organisme n'a pas été contractée lors du séjour à l'hôpital <p>Alors il est presque certain (0,8) que l'organisme est de la catégorie des enterobacteriaceae; de plus, il se peut (0,1 ; règle 37) que ce soit de la pseudomonias aeruginosa.</p>
Médecin	Pourquoi n'avez-vous pas envisagé la possibilité que l'organisme soit un streptocoque ?
TEIRESIAS	La règle 33 aurait du être utilisée pour identifier l'organisme comme un streptocoque, mais la seconde clause « la morphologie est de type coque » était déjà connue comme fausse et la règle 33 n'a jamais été essayée.

I. Introduction

- **SE pour valider l'intérêt économique**

- **PROSPECTOR (SRI International Inc., Université de Stanford 1978) :**

- Système d'aide à l'évaluation de ressources géologiques en vue de la prospection minière.
 - Il a découvert en 1981 un gisement de porphyre de molybdène d'une valeur de 100 millions de dollar là où les géologues n'avaient rien décelé.

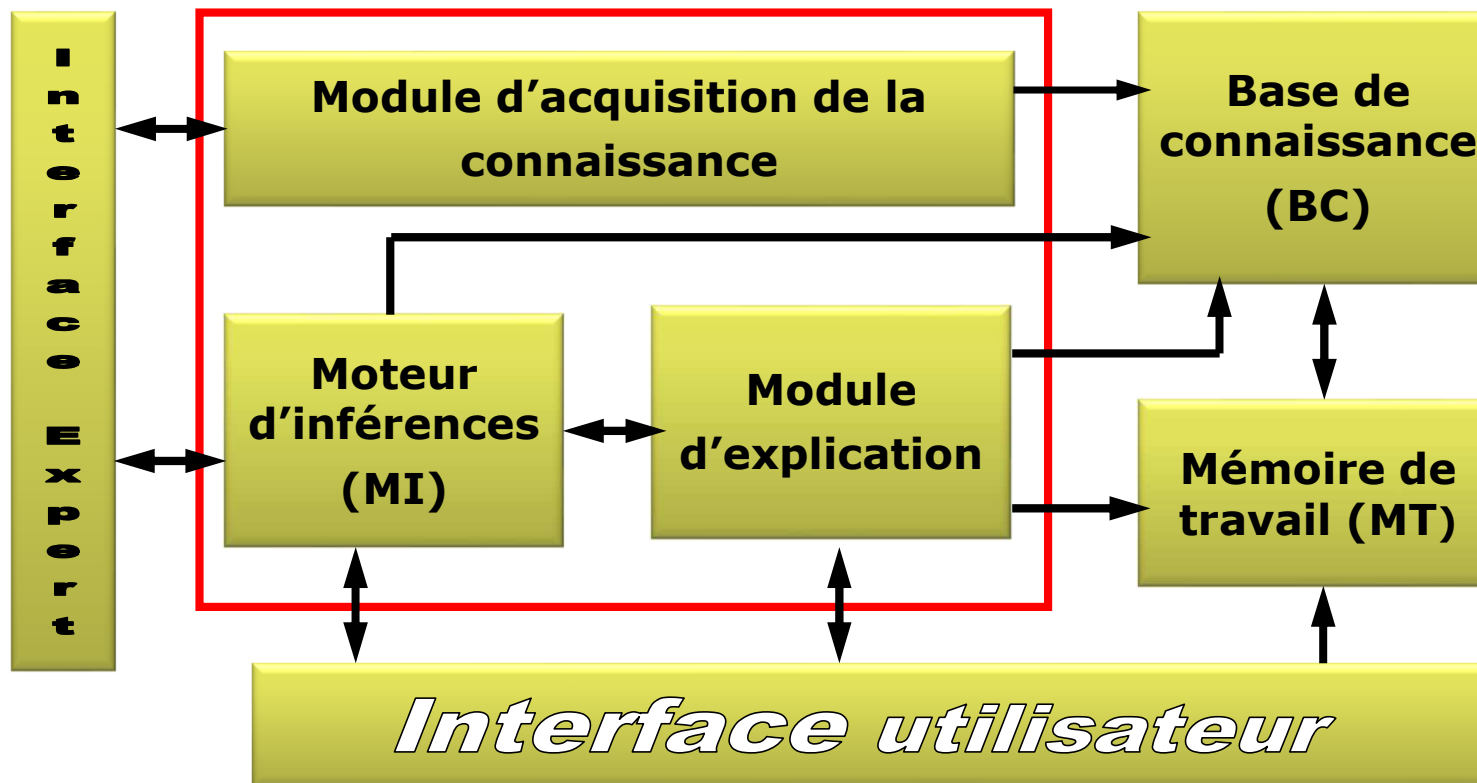
- **R1 (Carnegie Mellon University, DIGITAL EQUIPEMENT, 1980) :**

- Permet de proposer des configurations cohérentes de systèmes VAX à partir d'une commande.
 - Vérifie que l'ensemble des composants nécessaires à la cohérence du système est présent sinon il propose des modifications et une nouvelle configuration.
 - Son objectif est de répondre au mieux aux objectifs des clients et de réduire les coûts de fabrication.

I. Introduction

- **1.9. Domaines d'application**
 - **SE d'aide au diagnostic en médecine : MYCIN, DIABETO**
 - **SE pour la conception, CAO : TROPIC, R1**
 - **SE pour l'enseignement assisté par ordinateur : SOPHIE**
 - **SE pour la géologie : PROSPECTOR, ELFIN**
 - **SE pour la chimie : DENDRAL, CRYSTALIS**
 - **SE pour la robotique : ARGOS II**
 - **SE pour les jeux : PADISE, POKER, Deeper Blue**
 - **SE pour la démonstration automatique de théorème : PARI**

II. Architecture d'un SE



II. Architecture d'un SE

□ 2.1. Base de Connaissances (BC)

- Contient toute l'information dont un expert humain a besoin pour faire son travail, ceci dans un domaine donné.
- Seule composante du système qui contienne les connaissances propres au domaine.

■ Comporte deux composantes :

- **Base de Faits (BF)** : contient des faits spécifiques du domaine
 - Exemple : "maux de tête", fièvre, paludisme
 - Connaissances factuelles (savoir) de la base.

II. Architecture d'un SE

- **Base de Règles (BR) : contient des principes plus généraux, des règles, des heuristiques de résolution de problèmes**

- Représentent les modes de raisonnement propres au domaine considéré
- Connaissances déductives souvent représentées par des règles de production (savoir-faire).
- Exemple : "SI hypertension artérielle ALORS Anarcadium occidentale".

- Heuristiques peut provenir, soit de l'expert humain, soit d'une accumulation d'observations empiriques, soit de connaissances techniques propres au domaine.

II. Architecture d'un SE

- **2.2. Moteur d'inférence (MI)**
 - **Programme qui sait utiliser les connaissances pour résoudre un problème.**
 - **Modélise le raisonnement de l'expert au sein du système.**

II. Architecture d'un SE

- **2.3. Interface utilisateur**
 - **Facilite la communication entre le système et l'utilisateur en lui permettant de suivre le processus de résolution du MI.**

II. Architecture d'un SE

- **2.4. Interface expert**
 - **Permet le dialogue entre l'expert et le système.**

II. Architecture d'un SE

- **2.5. Module d'acquisition de connaissance**
 - **Outil utilisé par le cogniticien pour acquérir le savoir de l'expert.**
 - **Cf. GenRep**

II. Architecture d'un SE

- **2.6. Mémoire de travail ou base des faits**
 - **Contient les faits propres à un problème à résoudre.**

II. Architecture d'un SE

□ 2.7. Module d'explication

- Permet d'expliquer le raisonnement du MI.

III. Généralités sur les SE

□ 3.1. Éléments de logique des propositions

■ théories ayant présidés à la conception des SE :

- informatique, psychologie, logique.
- La logique fourni un langage pratique pour présenter les SE.

■ Exemple de propositions :

- P1: "les bébés pleurent quand ils ont faim"
- P2: "il pleut"
- P3: "je prends mon parapluie"
- P4: "il neige"
- P5: "les roses sont bleues"
- P6: "s'il pleut, je prends mon parapluie"
 - La validité d'une proposition P est donnée par sa "valeur de vérité" : $\text{val}(P)$.
 - La valeur de vérité est une fonction de l'ensemble des propositions sur l'ensemble à deux éléments: "vrai" et "faux".

III. Généralités sur les SE

■ Inférence

- A partir de propositions, il est possible de construire d'autres propositions à l'aide de diverses opérations.

■ Opérateurs

- **Conjonction (ET, \wedge)**
 - Exemple : $P2 \wedge P3$: "il pleut et je prends mon parapluie »
- **Négation (NON, \neg)**
 - Exemple: $\neg P4$: "il ne neige pas"
- **Disjonction (OU, \vee)**
 - Exemple: $P2 \vee P4$: "il pleut ou il neige"
- **Implication (SI ... ALORS, \rightarrow)**
 - Exemple : $P2 \rightarrow P3$ (= $P6$)
- **Règle d'inférence ou règle de déduction logique :**
 - **Modus Ponens**
 - **Modus Tollens**

III. Généralités sur les SE

■ Modus Ponens :

- $(P \wedge (p \Rightarrow q)) \Rightarrow q$
- Si $\text{val}(P) = \text{vrai}$ et $\text{val}(P \Rightarrow Q) = \text{vrai}$ on en déduit : $\text{val}(Q) = \text{vrai}$.
- C'est ce mode de raisonnement qui nous indique que si "il pleut" et que "chaque fois qu'il pleut, Toto prend son parapluie", alors "Toto doit être muni actuellement de son parapluie"

□ Preuve par table de vérité

p	q	$P \Rightarrow q$	$P \wedge (p \Rightarrow q)$	$(P \wedge (p \Rightarrow q)) \Rightarrow q$
V	V	V	V	V
V	F	F	F	V
F	V	V	F	V
F	F	V	F	V

III. Généralités sur les SE

■ Preuve directe :

- Supposons l'hypothèse soit vraie ; c.-à-d., p et $p \Rightarrow q$ sont supposés vraies.
- $p \Rightarrow q$ vraie veut dire par définition : soit (i) p est fausse, soit (ii) p et q sont vraies. Un des deux cas.
- Mais on a supposé p est vraie, donc c'est cas (ii). Ce qui implique que q est vraie aussi.
- Donc si l'hypothèse est vraie, la conclusion est vraie aussi. Cette implication est vraie.

$(p \wedge (p \rightarrow q)) \rightarrow q$ est une proposition logique toujours vraie, n'importe les propositions p et q .

En particulier, si p est remplacé par une autre proposition logique, et q aussi par une autre proposition logique alors l'implication reste vraie.

III. Généralités sur les SE

■ Modus Tollens :

- $(\neg q \wedge (p \Rightarrow q)) \Rightarrow \neg p$
- Si $\text{val}(P \Rightarrow Q) = \text{vrai}$ et $\text{val}(Q) = \text{faux}$ on en déduit: $\text{val}(P) = \text{faux}$.
- Preuve par table de vérité

p	q	$\neg q$	$p \Rightarrow q$	$\neg q \wedge (p \Rightarrow q)$	$\neg P$	$(\neg q \wedge (p \Rightarrow q)) \Rightarrow \neg P$
V	V	F	V	F	F	V
V	F	V	F	F	F	V
F	V	F	V	F	V	V
F	F	V	V	V	V	V

■ Preuve modus tollens

- En utilisant le modus ponens

$(p \wedge (p \Rightarrow q)) \Rightarrow q$ (1)
Substitution de p par $\neg q$ et de q par $\neg p$
 $(\neg q \wedge (\neg q \Rightarrow \neg p)) \Rightarrow \neg p$ (2)
Equivalence logique
 $(\neg q \Rightarrow \neg p) \Leftrightarrow (p \Rightarrow q)$ (3)
(2) et (3)
 $(\neg q \wedge (p \Rightarrow q)) \Rightarrow \neg p$ (CQFD)

III. Généralités sur les SE

- **Propositions appelées FAIT.**
 - **Implication $P \rightarrow Q$ est une REGLE.**
 - **P** : prémisses de la règle
 - **Q** : conclusion de la règle.

III. Généralités sur les SE

■ Examen d'un exemple

□ Instructions qui permettant de déterminer la nature d'une panne de démarreur.

□ Premier cas :

- le démarreur ne fonctionne pas ou fonctionne au ralenti. Dans ce cas, il est proposé d'allumer les grands phares. Si ceux-ci sont faibles, c'est un problème de batterie.
- Si les phares fonctionnent normalement, on propose alors d'essayer de faire démarrer le moteur avec les phares allumés.

- Si la lumière baisse, il y a un court-circuit dans le démarreur. Par contre, si la lumière des phares n'est pas modifiée, on propose de relier alors les deux grandes bobines. Si le moteur part, il y a un problème de bobine. Sinon la panne est autre.

□ Deuxième cas :

- le démarreur fonctionne, mais n'entraîne pas le moteur. C'est alors un problème de pignon.

III. Généralités sur les SE

- **On trouve dans cet exemple les faits, par exemple :**

- F1: "le démarreur ne fonctionne pas"
- F2: "les grands phares sont faibles"
- F3: "problème de batterie"

- **Une règle relie ces trois faits :**

- $R1: (F1 \wedge F2) \rightarrow F3$

- **Modus ponens permet de déduire :**

- $\text{val}(F3) = \text{vrai}$ lorsque $\text{val}(F1 \wedge F2) = \text{vrai}$, $\text{val}(R1)$ étant vrai

- **Réécriture de la règle R1 :**

- SI le démarreur ne fonctionne pas
- ET grands phares faibles
- **ALORS** problème de batterie

- **Exercice :**

- **Distinguer dans ce texte les autres faits, règles et informations complémentaires.**

III. Généralités sur les SE

- **Correction**
- **Faits**

F1: "le démarreur ne fonctionne pas"
F2: "les grands phares sont faibles"
F3: "problème de batterie"
F4: "démarrer le moteur avec les phare allumés"
F5: "la lumière baisse"
F6: "court circuit dans le démarreur"
F7: "relier les deux bobines"
F8: "moteur part"
F9: "problème de bobine"
F10: "autre panne"
F11: "problème de pignon"
F12: "le démarreur fonctionne au ralenti"

III. Généralités sur les SE

■ Règles :

R1 : $(F1 \vee F12) \wedge F2 \rightarrow F3$

SI le démarreur ne fonctionne pas ET grands phares faibles
ALORS problème de batterie

R2 : $(F1 \vee F12) \wedge \neg F2 \wedge F4 \wedge F5 \rightarrow F6$

SI le démarreur ne fonctionne pas ET NON les grands phares sont faibles ET démarrer le moteur avec les phare allumés ET la lumière baisse

ALORS court circuit dans le démarreur

R3 : $(F1 \vee F12) \wedge \neg F2 \wedge F4 \wedge \neg F5 \wedge F7 \wedge F8 \rightarrow F9$

SI le démarreur ne fonctionne pas ET NON les grands phares sont faibles ET démarrer le moteur avec les phare allumés ET la lumière baisse ET relier les deux bobines ET moteur part

ALORS problème de bobine

R4 : $(F1 \vee F12) \wedge \neg F2 \wedge F4 \wedge \neg F5 \wedge F7 \wedge \neg F8 \rightarrow F10$

SI le démarreur ne fonctionne pas ET NON les grands phares sont faibles ET démarrer le moteur avec les phare allumés ET la lumière baisse ET relier les deux bobines ET NON moteur part

ALORS autre panne

R5 : $\neg F1 \wedge \neg F8 \rightarrow F11$

SI NON le démarreur ne fonctionne pas ET NON moteur part
ALORS problème de pignon

III. Généralités sur les SE

□ 3.2. Logique d'ordre 0 et logique d'ordre 1

- Prenons le syllogisme classique suivant :
 - P1 : "Toto est un homme"
 - P2 : "les hommes sont mortels"
- **Déduction : P3 : "Toto est mortel".**

■ Mécanisme de déduction : Modus ponens

- Ces propositions font intervenir l'ensembles des hommes et celui des créatures mortelles.
- De la proposition universelle P2(X) :
 - "Si est X est un homme alors X est mortel",
 - on en déduit la proposition P3 à l'aide de P1, de P2(Toto) et du Modus ponens.
- **La logique utilisée introduit donc des variables.**

III. Généralités sur les SE

- **Logique sans variable : logique d'ordre 0.**
- **Logique qui admet des variables : logique d'ordre 1.**
- **SE sont d'ordre 0 ou 1 : selon qu'ils peuvent utiliser ou non des variables.**
- **Ordre 0 : logique des propositions**
 - **SI Ferrari ET Michael ALORS rapide**
- **Ordre 0⁺ : logique des propositions typées (attribut-valeur)**
 - **SI voiture = Ferrari ET pilote = Michael ALORS vitesse = rapide**
- **Ordre 1 : logique des prédicats**
 - **$\forall X, Y$: SI voiture(X) ET X=Ferrari ET pilote(X, Y) ET Y=Michael ALORS rapide(X)**
- **Ordre 2 : logique d'ordre 2**
 - **On peut affecter à un prédicat une variable.**
 - **$\forall R, X, Y$: SI type(R) = symétrique ET R(X, Y) ALORS R(Y, X)**

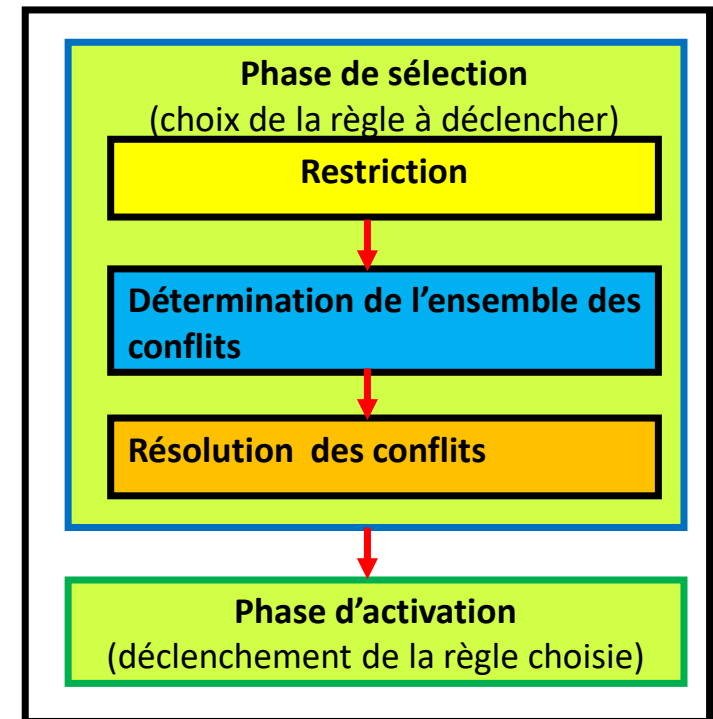
IV. Fonctionnement du MI

□ 4.1. Cycle de résolution du MI

■ Composés de deux phases :

- Phase de sélection
- Phase d'activation

Cycle de résolution de base



IV. Fonctionnement du MI

□ 4.2. Phase de sélection

- Composée de trois phases :

- **Phase 1 : Restriction**

- Réduction du domaine de connaissance (faits et règles) afin d'avoir les faits et règles pertinents.
- Deux sortes de restriction :
 - Restriction statique : séparation des règles en groupe de règles
 - Restriction dynamique : méta-règles permettent de travailler que sur certaines règles.

- **Phase 2 : Détermination de l'ensemble de conflits**

- Ou ensemble de règles candidates.
- sélection d'un ensemble des règles candidates à partir des faits et règles pertinent par filtrage (pattern-matching) avec les éléments de la mémoire de travail (ou MT).

IV. Fonctionnement du MI

■ Hypothèses de travail :

- Les règles sont des propositions vraies
- Les faits de la BF sont des faits vrais
- Les règles se présentent sous la forme : **SI A ET B ALORS C**
 - A ET B = déclencheur
 - C = Conclusion

□ Le système scrute alors la partie $A \wedge B$ de la règle :

- Il regarde si toutes les propositions contenues dans le déclencheur (ici A et B) sont aussi dans la BF.
- Si oui, la règle $A \wedge B \rightarrow C$ devient une règle applicable.

□ Supposons que le système décide d'exécuter cette règle

- la règle de détachement ou modus ponens entre en jeu et la proposition C est considérée comme vraie, le système l'ajoute alors à la BF.

IV. Fonctionnement du MI

■ Mécanismes de filtrage appelés modes de raisonnement :

- Le chaînage avant ou forward chaining
- Le chaînage arrière ou backward chaining
- Le chaînage mixte ou inférence bi-directionnelle

■ Phase 3 : résolution de conflits

■ Nombre de règles candidates important

- Choisir laquelle ou lesquelles appliquer en priorité.

■ Stratégies de contrôle :

- ensemble des méthodes permettant au MI d'enchaîner son raisonnement et résoudre les différents conflits se posant à lui en cours de son raisonnement.

IV. Fonctionnement du MI

□ Trois familles de stratégies de contrôle :

▪ **Choix par évaluation**

- Première règle devenue candidate, dernière règle devenue candidate
- Règle la plus informante (i.e. comportant plus de fait dans la partie droite), la règle la plus utilisée, la règle qui contient le fait le plus récemment utilisé

▪ **Choix par recherche exhaustive**

- Utilisation de coefficients d'atténuation pour les règles et coefficient de vraisemblance pour les faits.

▪ **Choix par méta-règles : ou règle sur les règles**

- Elles indiquent quelle règle il faut essayer en priorité dans chaque situation particulière. Elles sont transmises par l'expert.
- Chaque système possède ses propres heuristiques de choix de déclenchement de règles, ainsi certains MI sont dits paramétrables. 56

IV. Fonctionnement du MI

□ 4.3. Phase d'activation

- Appliquer la ou les règles élues sur la BF.
 - L'inférence de la règle active la partie action de la règle. Ceci a pour conséquence :
 - l'insertion de nouveaux faits dans la BF (chaînage avant)
 - ou la vérification d'hypothèse (chaînage arrière).
 - Si le but est atteint \Rightarrow arrêt des déductions
- Si aucune règle n'est déclenchée :
 - \Rightarrow Arrêt du MI : démarche irrévocable ou
 - \Rightarrow Reconsidération de l'étape précédente : démarche avec tentative
 - Test des conditions d'arrêt du moteur au cours de la phase de déduction.

IV. Fonctionnement du MI

□ 4.4. Mode de fonctionnement du MI

■ Fonctionnement monotone :

- Aucun fait ne peut être supprimé de la BF.
- Aucun fait ajouté n'introduit de contradictions dans la BF.

■ Fonctionnement non monotone :

- Les faits ne sont pas acquis une fois pour toutes.
- Retrait des règles ou faits considérés comme faux.
- Problème complexe.

V. Stratégie de recherche

□ 5.1. Présentation

- **Joue un rôle clé dans la phase de sélection (restriction) du cycle de base du MI.**

V. Stratégie de recherche

□ 5.2. Représentation par espace d'états

■ Espace d'états :

- Ensemble des états possibles d'un problème.
- Graphe orienté fait par le MI.
- Fonctionne en général en chaînage avant
- Représente l'ensemble des états que l'on peut obtenir avec les règles pour un problème donné.

■ On définit toujours :

- Un état objectif
 - succès de la recherche : but à prouver, BF saturée...
- Des sommets = des faits
- Des arcs = des transitions, des règles

V. Stratégie de recherche

■ Exemple de BC

□ La BR

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

□ La BF initiale : **H, K**

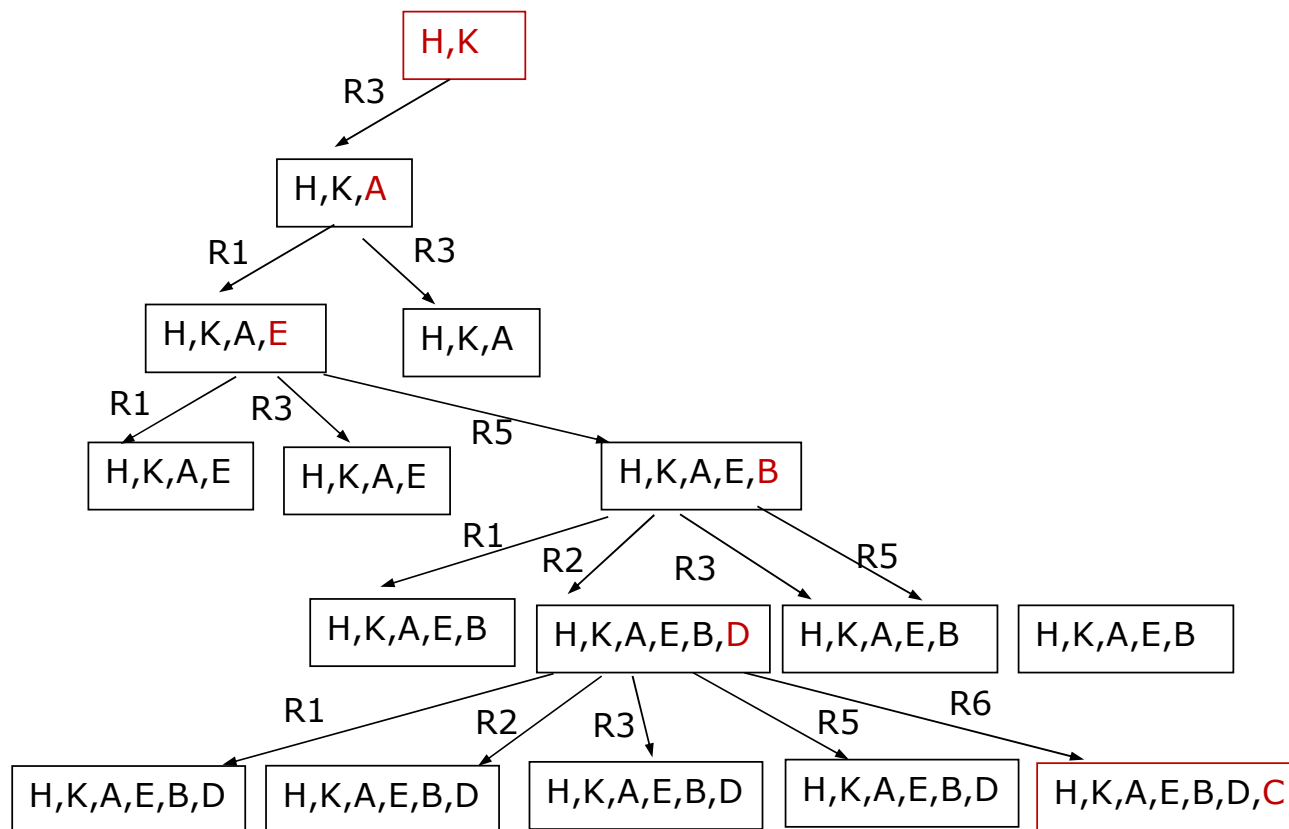
□ Le but à prouver : **C**

- Stratégie d'arrêt du moteur : l'on s'arrête lorsque le but recherché est prouvé ou lorsqu'il n'y a plus rien à déduire.

■ Stratégie de choix :

- La règle qui donne le but si elle est sélectionnable
- Une règle non encore utilisée sinon

V. Stratégie de recherche



Détection	Choix	Déclenchement
R3	R3	H, K, A
R1 R3	R1	H, K, A, E
R1 R3 R5	R5	H, K, A, E, B
R1 R2 R3 R5	R2	H, K, A, E, B, D
R1 R2 R3 R5 R6	R6	H, K, A, E, B, D, C

V. Stratégie de recherche

□ 5.3. Représentation par espace de sous-problèmes

■ En chaînage arrière :

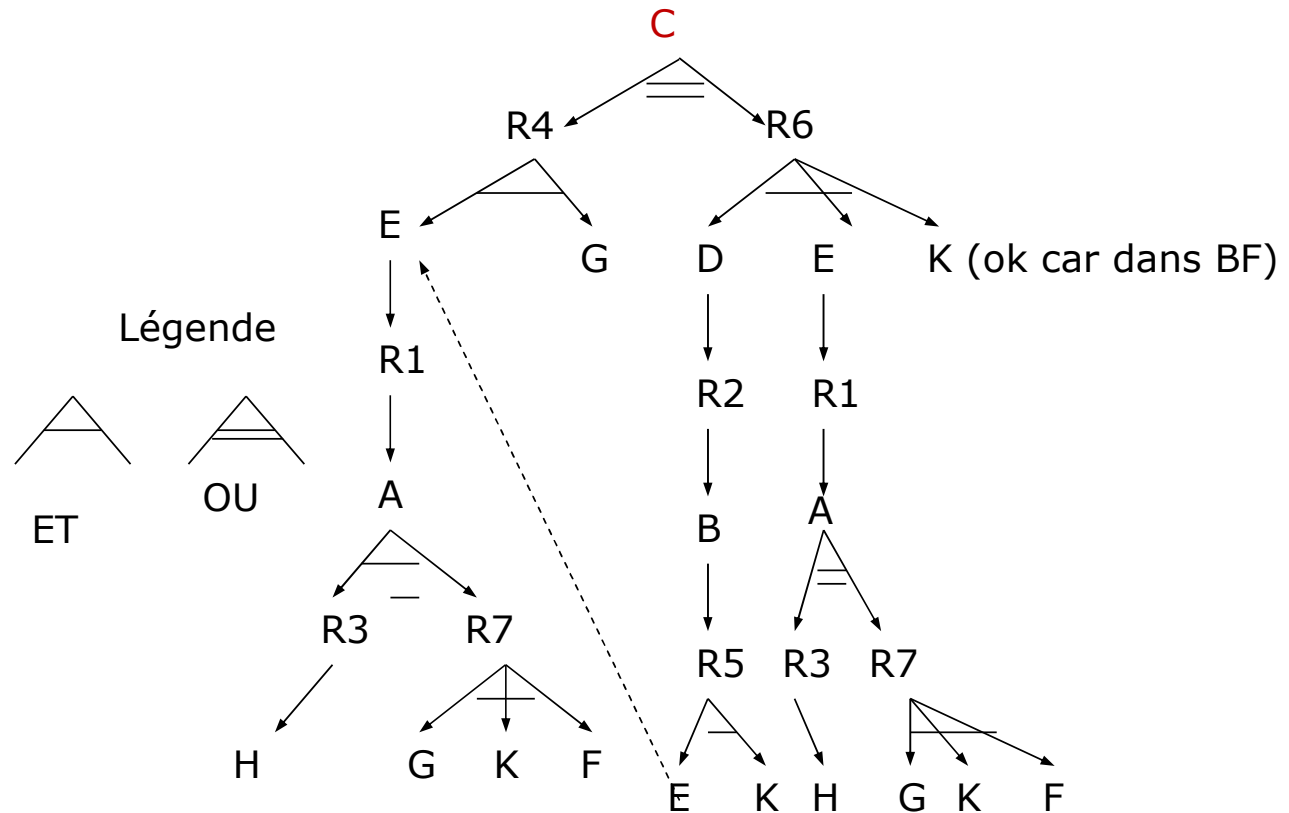
- On considère les règles comme des opérateurs de décomposition de problèmes (faits à établir) en sous-problème (autres faits à établir).
- L'espace des sous-problèmes est un graphe orienté du type ET/OU représentant l'ensemble des décompositions possibles.

■ Sur l'exemple précédent :

- **Stratégies d'arrêt du moteur :** l'on s'arrête lorsque les sous-buts traités sont des faits présents dans la BF.
- **Stratégies de choix :**
 - La règle qui a le plus de prémisses vérifiées
 - La règle qui donne le moins de sous-buts
 - Une règle non encore utilisée

V. Stratégie de recherche

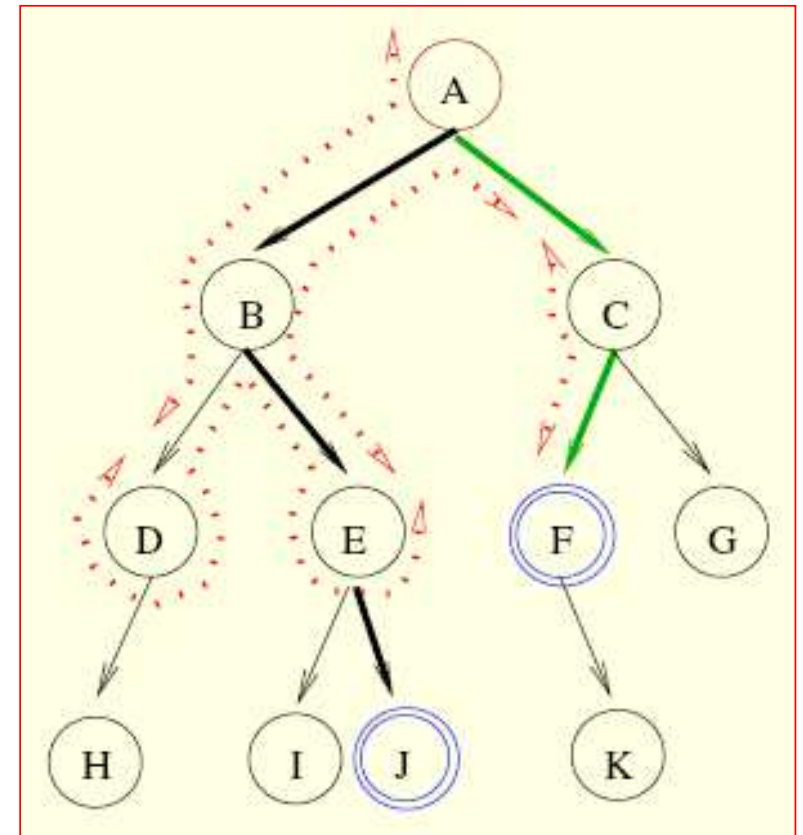
R1 : $A \rightarrow E$
R2 : $B \rightarrow D$
R3 : $H \rightarrow A$
R4 : $E, G \rightarrow C$
R5 : $E, K \rightarrow B$
R6 : $D, E, K \rightarrow C$
R7 : $G, K, F \rightarrow A$



V. Stratégie de recherche

□ 5.4. Mécanisme de recherche dans les graphes

- **Sélection des règles dans les graphes.**
- **Profondeur d'abord :**
 - Explorer une branche jusqu'au bout.



V. Stratégie de recherche

□ **Avantages :**

- On descend rapidement dans un arbre profond
- Facile à implémenter
- Peu cher, car on peut oublier une branche après parcours

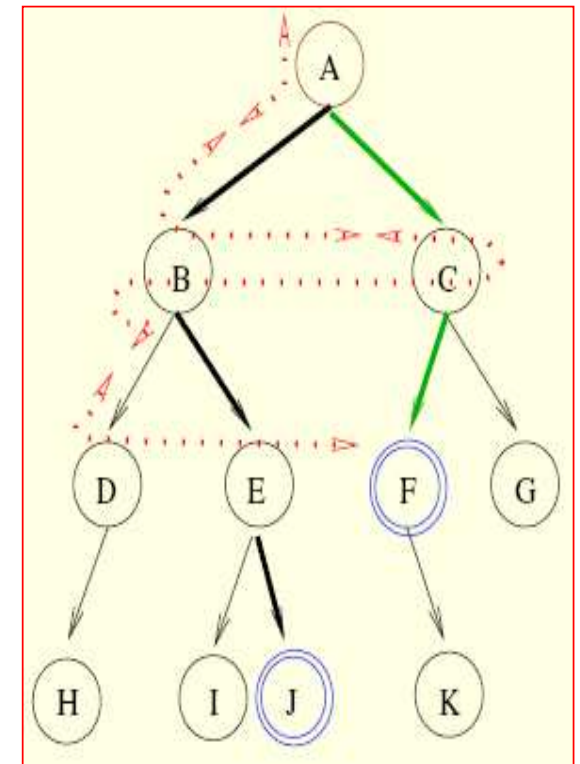
□ **Inconvénients :**

- Si parmi les premières branches explorées, figurent des branches infinies, on ne trouvera rien
- Si une solution simple existe dans une branche, elle peut être longue à trouver

V. Stratégie de recherche

■ Largeur d'abord :

- On parcourt par niveau
- On examine tous les chemins en même temps
- **Avantage :**
 - Les solutions triviales sont trouvées rapidement
- **Inconvénients :**
 - Il faut conserver tout l'arbre en mémoire centrale
 - Ne résout pas le problème d'explosion combinatoire



VI. Modes d'invocation des règles

□ 6.1. Présentation

- Deux types de règles d'inférences :
- Modus ponens ou règle de détachement
 - Si **P** est vrai et **$P \rightarrow Q$** est vraie alors on déduit **Q**
- Modus tollens ou raisonnement par l'absurde
 - Si **NON Q** est vraie et **$P \rightarrow Q$** est vraie alors on déduit **NON P**

■ Considérons la BC suivante :

□ Base des règles

R1 :	SI B ET D ET F	ALORS E
R2 :	SI D ET G	ALORS A
R3 :	SI C ET E	ALORS A
R4 :	SI B	ALORS K
R5 :	SI D	ALORS F
R6 :	SI A ET K	ALORS S
R7 :	SI C	ALORS D
R8 :	SI K ET C	ALORS A ET Z
R9 :	SI B ET K	ALORS D
R10 :	SI X	ALORS Z

□ Mémoire de travail ou BF

- Fait connus (ou initiaux) : B, C

□ Hypothèse à vérifier : S

VI. Modes d'invocation des règles

□ 6.2. Chaînage avant

- **Le MI fonctionne en chaînage avant lorsque les faits de la BF, sur lesquels portent les déclencheurs des règles, représentent des informations dont la valeur de vérité est déjà établie.**
- **Règle d'inférences utilisées**
 - **Mode d'invocation de règle est dirigé par la donnée.**

□ Utilisation du modus ponens :

- La règle **SI A ALORS B** ne peut être déclenchée que si toutes les conditions exprimées dans **A** sont vérifiées.
- le fait **B** peut alors être inféré et ajouté à la BF.
- **S'il manque des données pour progresser, le MI a la possibilité de demander des informations supplémentaires à l'utilisateur.**
- **Le processus se poursuit jusqu'à ce qu'une condition d'arrêt du MI soit vérifiée (saturation de la BC ou hypothèse vérifiée).**

VI. Modes d'invocation des règles

■ **Fonctionnement**

- **La stratégie de résolution de conflit : choix de la 1^{ère} règle rencontrée.**
- **MT = Mémoire de Travail**
- **EC = Ensemble de conflits (toutes les règles dont la partie gauche est dans la MT)**
- **RE = Règle Elue (après résolution de conflits)**

VI. Modes d'invocation des règles

R1 : SI B ET D ET F ALORS E
 R2 : SI D ET G ALORS A
 R3 : SI C ET E ALORS A
 R4 : SI B ALORS K
 R5 : SI D ALORS F
 R6 : SI A ET K ALORS S
 R7 : SI C ALORS D
 R8 : SI K ET C ALORS A ET Z
 R9 : SI B ET K ALORS D
 R10 : SI X ALORS Z

Mémoire de travail ou BF
Fait connus : B, C
Hypothèse à vérifier : S

Cycle 1 : MT = {B, C}

Ensemble de conflits par filtrage

EC = {R4, R7}

Résolution de conflits :

RE = {R4}

Activation de la règle élue

MT = {B, C, **K**}

S est non déduit

Cycle 2 : MT = {B, C, K}

EC = {R7, R8, R9}

RE = {R7}

MT = {B, C, K, **D**}

S est non déduit

Cycle 3 : MT = {B, C, K, D}

EC = {R5, R8, R9}

RE = {R5}

MT = {B, C, K, D, **F**}

S est non déduit

Cycle 4 : MT = {B, C, K, D, F}

EC = {R1, R8, R9}

RE = {R1}

MT = {B, C, K, D, F, **E**}

S est non déduit

Cycle 5 : MT = {B, C, K, D, F, E}

EC = {R3, R8, R9}

RE = {R3}

MT = {B, C, K, D, F, E, **A**,}

S est non déduit

Cycle 6 : MT = {B, C, K, D, F, E, A}

EC = {R6, R8, R9}

RE = {R6}

MT = {B, C, K, D, F, E, A, **S**}

S est déduit

⇒ Succès

⇒ Arrêt du processus de déduction

Chemin = R4 → R7 → R5 → R1 → R3 → R6

VI. Modes d'invocation des règles

■ Changeons la stratégie de choix de résolution de conflits :

- Règle élue = règle la plus informante (plus de faits dans la partie droite de la règle) ; en cas d'égalité, on choisit la 1^{ère} règle.

R1 : SI B ET D ET F	ALORS E
R2 : SI D ET G	ALORS A
R3 : SI C ET E	ALORS A
R4 : SI B	ALORS K
R5 : SI D	ALORS F
R6 : SI A ET K	ALORS S
R7 : SI C	ALORS D
R8 : SI K ET C	ALORS A ET Z
R9 : SI B ET K	ALORS D
R10 : SI X	ALORS Z

Cycle 1 : MT = {B, C}
EC = {R4, R7}
RE = {R4}
MT = {B, C, **K**}

S est non déduit

Cycle 2 : MT = {B, C, K}
EC = {R7, R8, R9}
RE = {R8}
MT = {B, C, K, **A, Z**}

S est non déduit

Cycle 3 : MT = {B, C, K, A, Z}
EC = {R6, R7, R9}
RE = {R6}
MT = {B, C, K, A, Z, **S**}

S est déduit

⇒ Succès

⇒ Arrêt du processus de déduction

Chemin = R4 → R8 → R6

■ Conclusion

- la stratégie de choix de résolution de conflits influe sur le temps de recherche d'une solution.

VI. Modes d'invocation des règles

■ Algorithme

□ Évaluation d'une règle

- soit BF une base de faits,
- et la règle R : **SI A_1 ET A_2 ET ... ET A_n ALORS A**
- $\text{eval}(R, BF) = \text{vrai}$ ssi pour tout i , $\text{match}(A_i, BF) = \text{vrai}$
- $\text{match}(A, BF) = \text{vrai}$ ssi $A \in BF$
- H= hypothèse

```
FONCTION chainageAvant(BFinit, BR, H):liste
DEBUT
  listeF  $\leftarrow$  BFinit
  REPETER
    BF  $\leftarrow$  listeF
    POUR toute règle R non encore déclenchée
      SI eval(R, BF) ALORS
        listeF  $\leftarrow$  listeF  $\cup$  conclusion(R)
    JUSQUA(listeF = BF OU H  $\in$  listeF)
  chainageAvant  $\leftarrow$  listeF
FIN
```

□ Complexité

- l'algorithme ChainageAvant(BFinit; BR) est polynomial par rapport à la taille de BFinit, à la taille de BR, et à la taille d'une règle.

VI. Modes d'invocation des règles

□ Évaluation d'une requête Q par chainage-avant

- Q est satisfaite ssi Q appartient au résultat de
ChainageAvant(Bfinit, BR, H)
- Résultat : l'algorithme
ChainageAvant(Bfinit, BR) est
polynomial par rapport à la taille
de Bfinit, à la taille de BR, et à la
taille d'une règle

□ Base de règles

R1	SI E ET C	ALORS B
R2	SI A ET D	ALORS E
R3	SI A ET B ET F	ALORS G
R4	SI C ET G	ALORS F

□ MT ou BF) : BF = {A, C, D}

□ Hypothèse à vérifier : B

```
Bfinit = {A, C, D}
listeF = {A, C, D}
Repeter1
  BF = {A, C, D}
  R2 : listeF = {A, C, D, E}

Repeter2
  BF = {A, C, D, E}
  R1 : listeF = {A, C, D, E, B}

  B ∈ ListeF
  Arrêt du MI
```

VI. Modes d'invocation des règles

■ Exercice 1

- Soit la BC suivante ;
- R1 SI personne aisée ET temps libre ALORS peut voyager
- R2 SI temps libre ALORS peut faire de la poterie
- R3 SI peut peindre ALORS peut faire un produit artisanal
- R4 SI peut faire de la poterie ALORS peut faire un produit artisanal
- R5 SI peut faire un produit artisanal ET artiste ALORS peut faire une œuvre d'art

□ La base de fait est :

- personne aisée
 - temps libre
 - artiste
- Hypothèse : peut faire une œuvre d'art

VI. Modes d'invocation des règles

■ Exercice 2

□ Base des règles

- R1 SI $\neg P$ ALORS B ET $\neg D$ ET $\neg E$
- R2 SI $\neg A$ ET B ALORS $\neg D$ ET $\neg G$
- R3 SI $\neg A$ ALORS C ET $\neg P$ ET K
- R4 SI $\neg D$ ET L ALORS $\neg C$ ET Y
- R5 SI $\neg E$ ALORS M ET $\neg D$
- R6 SI $\neg H$ ALORS L ET $\neg A$
- R7 SI $\neg H$ ET W ALORS Z
- R8 SI $\neg P$ ET L ALORS W

□ Mémoire de travail

- BF = $\{\neg H, B, C, X\}$
- Hypothèse : saturation (plus de faits à déduire)

□ Remarque :

- Tentative, Univers non monotone (remise en cause des faits déduits).
- $\neg H$ = fait à établir (NON H), On ne peut avoir dans la BF H et $\neg H$

VI. Modes d'invocation des règles

■ Correction Exercice 1

□ La base de fait est : {pa, tl, a},
Hypothèse : pfoa

□ Cycle 1 : MT = {pa, tl, a}

- EC = {R1, R2}
- RE = {R1}
- MT = {pa, tl, a, pv}
- peut pfoa est non déduit

□ Cycle 2 : MT = {pa, tl, a, pv}

- EC = {R2}
- RE = {R2}
- MT = {pa, tl, a, pv, pfp}
- pfoa est non déduit

□ Cycle 3 : MT = {pa, tl, a, pv, pfp}

- EC = {R4}
- RE = {R4}
- MT = {pa, tl, a, pv, pfp, pfpa}
- Pfoa est non déduit

□ Cycle 4 : MT = {pa, tl, a, pv, pfp, pfpa}

- EC = {R5}
- RE = {R5}
- MT = {pa, tl, a, pv, pfp, pfpa, pfoa}
- pfoa est déduit
- succès

VI. Modes d'invocation des règles

■ Remarque :

- Même si on n'obtenait pas un succès le processus de déduction se serait arrêté car on a saturation de la BR (toute les règles ont été utilisées).

VI. Modes d'invocation des règles

■ Solution Exercice 2

Cycle 1 : MT = {¬H, B, C, X}

EC = {R6}

RE = {R6}, ¬H est remplacé par ¬A

MT = {L, ¬A, B, C, X}

Cycle 2 : MT = {L, ¬A, B, C, X}

EC = {R2, R3}

RE = {R2}, ¬A est remplacé par ¬D, ¬G

MT = {¬D, ¬G, L, B, C, X}

Cycle 3 : MT = {¬D, ¬G, L, B, C, X}

EC = {R4}

RE = {R4}, ¬D est remplacé par ¬C, Y,
Comme on ne peut avoir dans la BF C et ¬C,
on n'ajoute pas ¬C

MT = {Y, ¬G, L, B, C, X}

Cycle 4 : MT = {Y, ¬G, L, B, C, X}

Échec

Retour au cycle 2 pour choisir R3

Univers non monotone, donc remise en
cause des faits déduits.

Cycle 5 : MT = {L, ¬A, B, C, X}

EC = {R2, R3}

RE = {R3}, ¬A est remplacé par C, ¬P, K

MT = {¬P, K, L, B, C, X}

Cycle 6 : MT = {¬P, K, L, B, C, X}

EC = {R1, R8}

RE = {R1}, ¬P est remplacé par ¬D, ¬E

MT = {¬D, ¬E, K, L, B, C, X}

Cycle 7 : MT = {¬D, ¬E, K, L, B, C, X}

EC = {R4, R5}

RE = {R4}, ¬D est remplacé par ¬C ET Y

MT = {Y, ¬E, K, L, B, C, X}

Cycle 8 : MT = {Y, ¬E, K, L, B, C, X}

EC = {R5}

RE = {R5}, ¬E est remplacé par M, ¬D

MT = {M, ¬D, Y, K, L, B, C, X}

Cycle 9 : MT = {Y, M, ¬D, K, L, B, C, X}

EC = {R4}

RE = {R4}, ¬D est remplacé par ¬C, Y

MT = {Y, M, K, L, B, C, X}

Cycle 10 : MT = {Y, M, K, L, B, C, X}

EC = {}

RE = {}, ¬E est remplacé par M, ¬D

Arrêt, plus de fait à déduire

VI. Modes d'invocation des règles

▪ Autre présentation

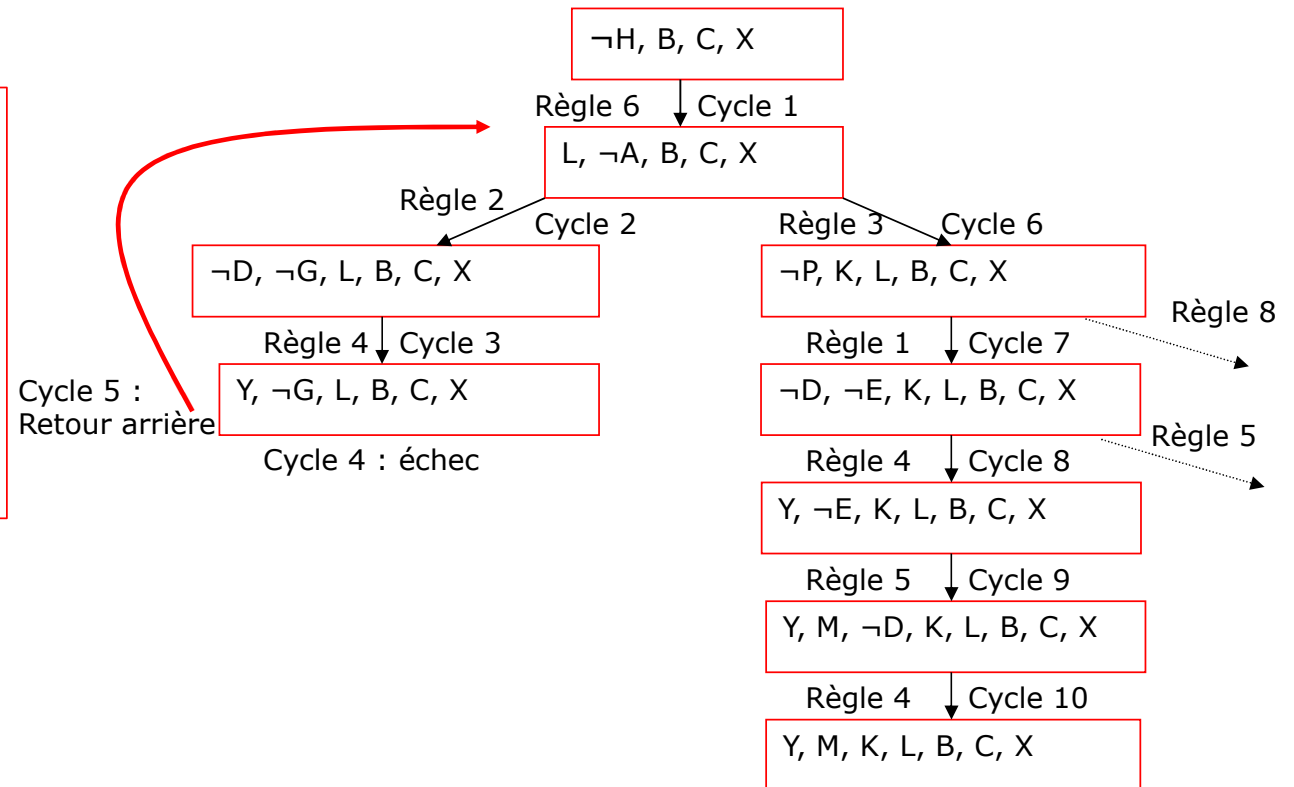
Base des règles

R1 SI $\neg P$ ALORS B ET $\neg D$ ET $\neg E$
 R2 SI $\neg A$ ET B ALORS $\neg D$ ET $\neg G$
 R3 SI $\neg A$ ALORS C ET $\neg P$ ET K
 R4 SI $\neg D$ ET L ALORS $\neg C$ ET Y
 R5 SI $\neg E$ ALORS M ET $\neg D$
 R6 SI $\neg H$ ALORS L ET $\neg A$
 R7 SI $\neg H$ ET W ALORS Z
 R8 SI $\neg P$ ET L ALORS W

Mémoire de travail

BF = { $\neg H$, B, C, X}

Hypothèse : saturation (plus de faits à déduire)



VI. Modes d'invocation des règles

□ 6.3. Chaînage arrière

- **Le MI fonctionne en chaînage arrière lorsque certains faits de la BF sont considérés comme étant à établir ou à évaluer.**
 - On les appelle souvent problèmes (en suspens, à résoudre) ou hypothèses (à vérifier) ou buts (à atteindre).
- **Quand l'utiliser ?**
 - Pour guider les questions à poser à l'utilisateur en cas d'information incomplète dans la BF pour satisfaire la requête.

■ Généralement, il est combiné avec le chaînage-avant :

- chaînage-avant : pour saturer la BF
- chaînage-arrière : pour guider l'interaction avec l'utilisateur, exploration d'un arbre ET/OU.
- **Règle d'inférences utilisée**
 - Modus tollens.
 - Mode de raisonnement est guidé par le but à atteindre.
 - La BF est constituée de faits établis et de faits non encore établis que l'on cherche à établir.

VI. Modes d'invocation des règles

- ▣ **Le MI choisit une règle permettant de vérifier l'hypothèse et essaye d'en vérifier chaque prémisse (partie gauche de la règle).**

- Si les prémisses sont vérifiées le problème est résolu.
- Sinon les conditions non vérifiées deviennent des sous-buts à prouver.
- On réitère le processus qui se poursuit jusqu'à la vérification des prémisses de la règle initialement considérée.
- **Si une règle n'a pu être déclenchée, le système en cherche une autre permettant d'aboutir au but fixé.**

- ▣ **Le MI à un raisonnement à rebours, c'est-à-dire qu'il part des buts pour arriver aux données.**

- ▣ **Le système explore un arbre ET/OU :**

- Nœuds OU : ensemble des règles pouvant déduire un même fait.
- Nœuds ET : la conjonction des prémisses d'une règle.

- **La formalisation des connaissances par les experts est plus délicate que pour le chaînage avant.**

VI. Modes d'invocation des règles

■ **Fonctionnement**

□ **Phase de sélection**

- Recherche des règles (candidates) dont la partie droite est le but b que l'on s'est fixé : soit R cet ensemble.
- Choix d'une règle dans R

□ **Phase d'activation**

- Les éléments de la partie gauche de la règle déclenchée vont remplacer le but b comme un des sous-buts.
- Ce processus s'arrêtera quand le but initial sera réduit à un ensemble de sous-buts tous présents dans la mémoire de travail
- ou lorsqu'aucune règle ne puisse être déclenchée.

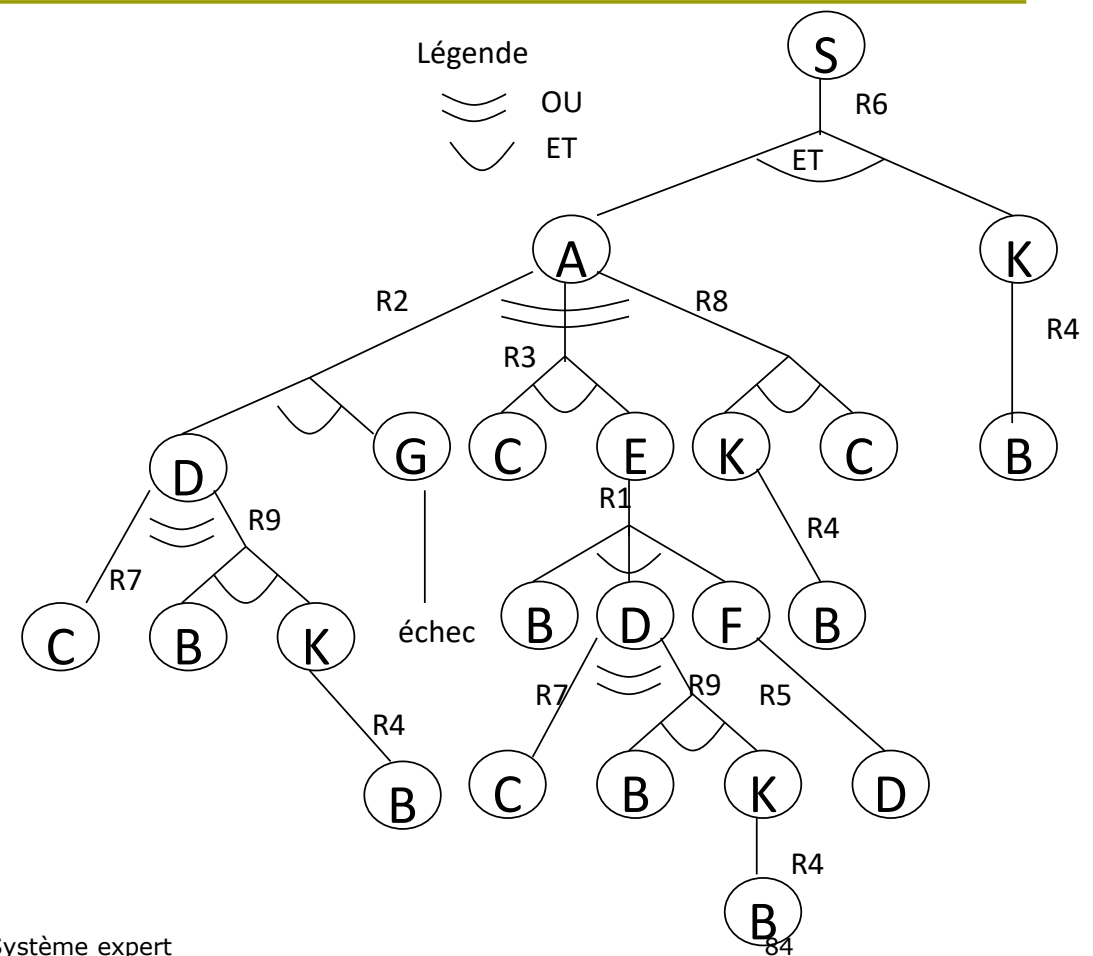
VI. Modes d'invocation des règles

■ **Considérons la BC suivante :**

□ Base des règles

R1 : SI B ET D ET F	ALORS E
R2 : SI D ET G	ALORS A
R3 : SI C ET E	ALORS A
R4 : SI B	ALORS K
R5 : SI D	ALORS F
R6 : SI A ET K	ALORS S
R7 : SI C	ALORS D
R8 : SI K ET C	ALORS A ET Z
R9 : SI B ET K	ALORS D
R10 : SI X	ALORS Z

- **Mémoire de travail ou BF**
- **Fait connus (ou initiaux) : B, C**
- **Hypothèse à vérifier : S**



VI. Modes d'invocation des règles

■ **Au cycle 2**

- le choix de la règle R2 conduit à un échec
- le choix de la règle R3 conduit à un succès
- le choix de la règle R8 conduit à un succès

VI. Modes d'invocation des règles

■ Autre méthode

<p>Cycle 1 : MT = {B, C, ¬S}</p> <p>Ensemble de conflits par filtrage EC = {R6}</p> <p>Résolution de conflits : RE = {R6}</p> <p>Activation de la règle élue MT = {B, C, ¬A, ¬K}</p> <p>S est non déduit</p> <p>Cycle 2 : MT = {B, C, ¬A, ¬K}</p> <p>EC = {R2, R3, R8} RE = {R2} MT = {B, C, ¬D, ¬G, ¬K}</p> <p>S est non déduit</p> <p>Cycle 3 : MT = {B, C, ¬D, ¬G, ¬K}</p> <p>EC = {R7, R9} RE = {R7} MT = {B, C, ¬C, ¬G, ¬K}</p> <p>On ne peut avoir C et ¬C C étant prouvé donc D l'est MT = {B, C, D, ¬G, ¬K}</p> <p>S est non déduit</p>	<p>Cycle 4 : MT = {B, C, D, ¬G, ¬K}</p> <p>EC = {} pour ¬G RE = {} MT = {B, C, D, ¬G, ¬K}</p> <p>Retour au cycle 2 pour envisager R3</p> <p>S est non déduit</p> <p>Cycle 5 : MT = {B, C, D, ¬K}</p> <p>EC = {R3, R8} RE = {R3} MT = {B, C, D, ¬C, ¬E, ¬K}</p> <p>MT = {B, C, D, ¬E, ¬K}</p> <p>S est non déduit</p> <p>Cycle 6 : MT = {B, C, D, ¬E, ¬K}</p> <p>EC = {R1} RE = {R1} MT = {B, C, D, ¬B, ¬D, ¬F, ¬K}</p> <p>MT = {B, C, D, ¬F, ¬K}</p> <p>S est non déduit</p>	<p>Cycle 7 : MT = {B, C, D, E, ¬F, ¬K}</p> <p>EC = {R5} RE = {R5} MT = {B, C, D, ¬D, ¬K}</p> <p>MT = {B, C, D, ¬G, E, F, E, A, ¬K}</p> <p>S est non déduit</p> <p>Cycle 8 : MT = {B, C, D, E, F, E, A, ¬K}</p> <p>EC = {R4} RE = {R4} MT = {B, C, D, E, F, E, A, ¬B}</p> <p>MT = {B, C, D, E, F, E, A, K, S}</p> <p>S déduit</p>
---	---	--

VI. Modes d'invocation des règles

■ Algorithmme

```
Fonction chainageArriere(H, BFininit, BR) : BOOLEEN
DEBUT
  SI match(H, BFininit) ALORS
    | ChainageArriere ← VRAI
  SINON
    DEBUT
      POUR Toute règle R : SI P1 ET P2 ET ... ET Pn ALORS C t.q. match(H, C)
        DEBUT
          bool ← VRAI
          i ← 1
          TANTQUE(bool ET i ≤ n)
            DEBUT
              | bool ← chainageArriere(Pi, BFininit, BR)
              | i ← i + 1
            FIN
          SI bool ALORS
            ChainageArriere ← VRAI
        FIN
      ChainageArriere ← FAUX
    FIN
  FIN
```

□ Base de règles

R1	SI E ET C	ALORS B
R2	SI A ET D	ALORS E
R3	SI A ET B ET F	ALORS G
R4	SI C ET G	ALORS F

□ **Mémoire de travail ou BF) : BF = {A, C, D}**

□ **Hypothèse à vérifier : B**

VI. Modes d'invocation des règles

■ Exécution

```
BFinit = {A, C, D} BF = {A, B, C, D, E, F} BR = {R1, R2, R3, R4,}
resultat = chainageAr(B, BFinit, BR)
  match(B, BFinit) = FAUX
  POUR
    R1 : SI E ET C ALORS B  bool = VRAI  i = 1
    TANTQUE
      E : bool = chainageAr(E, BFinit, BR)  i = 2
      match(E, BFinit) = FAUX
      POUR
        R2 : SI A ET D ALORS E  bool = VRAI  i = 1
        TANTQUE
          A : bool = chainageAr(A, BFinit, BR)  i = 2
          match(A, BFinit) = VRAI
          D : bool = chainageAr(D, BFinit, BR)  i = 3
          match(D, BFinit) = VRAI
        bool = VRAI
      Arrêt du MI
  resultat = VRAI
```


VI. Modes d'invocation des règles

■ Exercice 1

□ Base des règles

R1 SI A ALORS B
R2 SI B ALORS C
R3 SI D ALORS NON C
R4 SI B OU NON C ALORS E
R5 SI F ALORS B
R6 SI NON G ALORS A
R7 SI E OU D ALORS F
R8 SI H ALORS NON B
R9 SI G ALORS H

□ **BF = {D, E, G}**

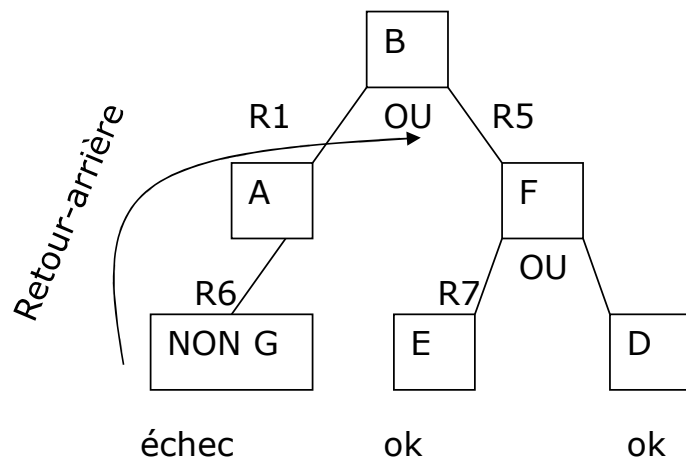
□ **Hypothèse : B**

□ **tentative**

□ **Univers non monotone (remise en cause des faits déduits)**

VI. Modes d'invocation des règles

■ Correction



VI. Modes d'invocation des règles

■ Chainage arrière avec questions à l'utilisateur

□ Il faut déterminer quelles questions on peut poser à l'utilisateur.

- **demandable(H)** : On peut poser une question sur H. si H n'apparaît pas en conclusion de règles, il est demandable.
- **question(H)** : booléen renvoyant la réponse de l'utilisateur à la question demandant si H est vrai.

□ Principes des modifications :

- Si on n'arrive pas à prouver un fait, on le demande à l'utilisateur en dernier recours.

□ Algorithme

VI. Modes d'invocation des règles

```
Fonction chainageArriere(H, BFinet, BR) : BOOLEEN
DEBUT
  SI match(H, BFinet) ALORS
    | ChainageArriere ← VRAI
  SINON
    DEBUT
      POUR Toute règle r : SI l1 ET l2 ET ... ET ln ALORS Conc t.q. match(H, Conc)
        DEBUT
          bool ← VRAI
          i ← 1
          TANTQUE(bool ET i ≤ n)
            DEBUT
              | bool ← chainageArriere(li, BFinet, BR)
              | i ← i + 1
            FIN
          SI bool ALORS
            | ChainageArriere ← VRAI
          FIN
        Si demandable(H) Alors
          ChainageArriere ← question(H)
        SINON
          ChainageArriere ← FAUX
      FIN
    FIN
  FIN
```

VI. Modes d'invocation des règles

■ Exercice 2

□ BC

R1 : VH → VMV
R2 : VA → VMV
R3 : NA → VMV
R4 : VA, TG → VMV
R5 : BV, NM → VA
R6 : VAn, NM → VH
R7 : NC, NT → NA
R8 : AS, TF → NA
R9 : EV, CA → VAg
R10 : EV, NM → BV
R11 : NR, BR → TG

VH : ville historique
VMV : ville méritant le voyage
VA : ville artistique
NA : nombreuses animations
~~VA : ville agréable~~
TG : tradition gastronomique
BV : belle ville
NM : nombreux monuments
VAn : ville ancienne
NM : nombreux monuments

NC : nombreux concerts
NT : nombreux théâtres
AS : activités sportives
TF : traditions folkloriques
EV : espaces verts
CA : climat agréable
VAg : ville agréable
EV : espaces verts
NR : nombreux restaurants
BR : bons restaurants

INP-HB/K. M. BROU

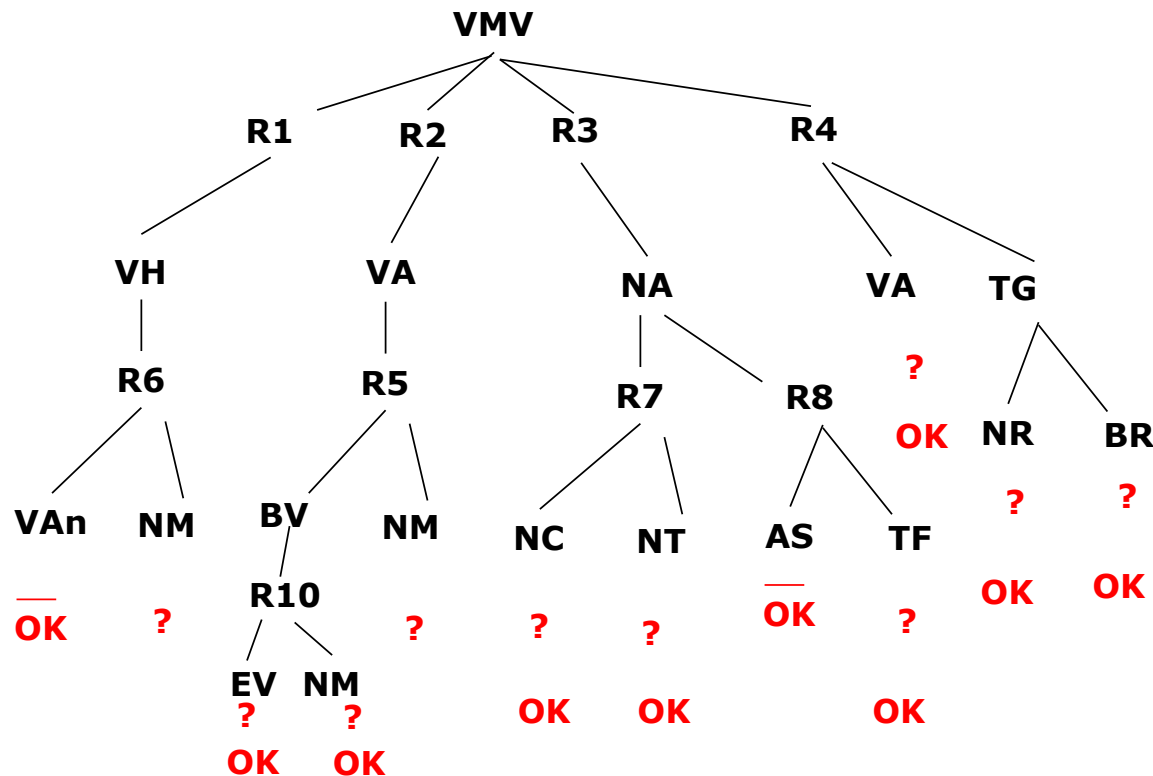
IA : Système expert

□ Question

- Construisez l'arbre ET/OU qui correspond au fait à prouver "ville méritant le voyage" (VMV).
- Si le MI n'arrive pas à prouver un fait, il pose un certain nombre de question à l'utilisateur pour pouvoir décider en dernier recours.
- Faits demandables : NM, VA, NC, NT, AC, TF, EV, CA, NR, BR

VI. Modes d'invocation des règles

■ Correction



R1 : VH → VMV
 R2 : VA → VMV
 R3 : NA → VMV
 R4 : VA, TG → VMV
 R5 : BV, NM → VA
 R6 : VAn, NM → VH
 R7 : NC, NT → NA
 R8 : AS, TF → NA
 R9 : EV, CA → VAg
 R10 : EV, NM → BV
 R11 : NR, BR → TG

■ Problème :

□ Risques de bouclages

■ Solution :

□ Ne pas relancer l'algorithme sur un sous-but déjà examiné.

Faits demandables : NM, VA, NC, NT, AS, TF, EV, CA, NR, BR

VI. Modes d'invocation des règles

■ Algorithme

```
Fonction chainageArriere(H, BFinet, BR) : BOOLEEN
DEBUT
  SI match(H, BFinet) ALORS
    | ChainageArriere ← VRAI
  SINON
    DEBUT
      dejaEssaye ← ∅
      POUR Toute règle r : SI l1 ET l2 ET ... ET ln ALORS Conc t.q. match(H, Conc)
      DEBUT
        bool ← VRAI
        i ← 1
        TANTQUE(bool ET i ≤ n)
          DEBUT
            SI(l1 ∉ dejaEssaye) ALORS
              DEBUT
                dejaEssaye ← dejaEssaye ∪ {li}
                bool ← chainageArriere(li, BFinet, BR)
              FIN
            i ← i + 1
          FIN
        SI bool ALORS
          ChainageArriere ← VRAI
        FIN
      Si demandable(H) Alors
        | ChainageArriere ← question(H)
      SINON
        ChainageArriere ← FAUX
    FIN
  FIN
```

VI. Modes d'invocation des règles

□ 6.4. Chaînage mixte

■ Idée de base

□ Alternier les deux mécanismes :

- Utiliser l'information fournie par l'utilisateur pour choisir un but.
- Ensuite demander à l'utilisateur plus d'informations alors que le processus évolue vers la résolution de ce but.

■ Une partie des faits de la BF est considérée comme étant à établir:

□ Ce sont des problèmes.

■ D'autres sont considérés comme établis :

□ Ce sont des faits proprement dits

■ Chaînage mixte :

- Car les conditions des règles peuvent porter simultanément sur des faits établis (chainage avant) ou sur les faits à établir (chainage arrière).

VI. Modes d'invocation des règles

- **Exemple 1 :**

- **BC**

R1 : SI Tropiques ALORS Les_Saintes
R2 : SI Saint-Bart et hôtel ALORS Hôtel Paradisio
R3 : SI dépressif ALORS Tourisme chaud
R4 : SI tourisme chaud ALORS tropiques
R5 : SI Les_Saintes ALORS Hôtel Paradisio
R6 : SI Les_Saintes ALORS tourisme chaud
R7 : SI P.D.G. ALORS tourisme chaud
R8 : SI tourisme chaud et Les_Saintes ALORS
tourisme chaud et voilier
R9 : SI Hôtel Paradisio ALORS Caraïbes

- **Base de faits état initial :**

- **Les_Saintes**

VI. Modes d'invocation des règles

■ Correction :

Cycle 1 : MT = {Les_Saintes}

Règle déclenchée R5.

MT = {Les_Saintes, Hôtel Paradisio}

Cycle 2 : MT = {Les_Saintes, Hôtel Paradisio}

Règle déclenchée R9.

MT = {Les_Saintes, Hôtel Paradisio, Caraïbes}

Cycle 3 : MT = {Les_Saintes, Hôtel Paradisio, Caraïbes}

Le moteur est maintenant bloqué en chaînage avant.

Il n'existe pas de prémisse qui contienne Caraïbes.

Il passe donc en chaînage arrière mais il n'existe pas de règle activable ayant une partie droite (conclusion) qui contienne Caraïbes.

Le moteur passe à nouveau en chaînage avant.

Les règles activables sont R1, R2, R3, R4, R6, R7, R8.

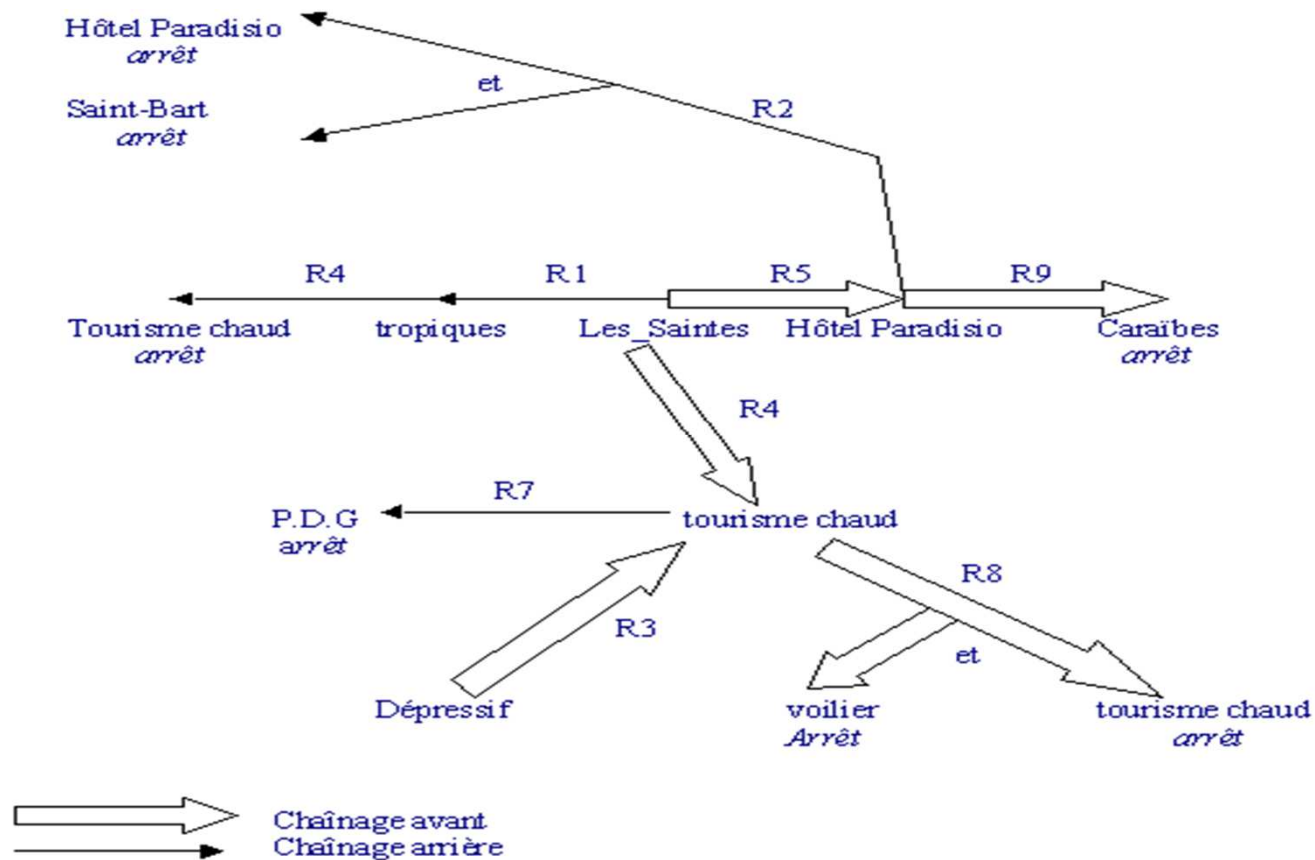
La règle R6 est sélectionnée et activée.

Elle permet de déduire tourisme chaud.

etc...

VI. Modes d'invocation des règles

■ Graphe :



VI. Modes d'invocation des règles

■ Exemple 2 : planification

■ Fonctionnement

- Les règles sont invocables par examen de leur membre gauche.
 - Chaque élément du membre gauche est un filtre, réduit à un symbole simple.
- Le premier filtre se réfère à une partie spéciale de la BF ne contenant que des faits à établir :
 - **appelée PILEDESPROBLEMES (PPB).**
 - Plus précisément, le premier filtre se réfère au premier élément de la liste matérialisant PPB.
- Les autres filtres, éventuels, du membre gauche d'une règle se référeront à la partie de la BF qui ne contient, par convention, que des faits établis.

VI. Modes d'invocation des règles

■ Considérons la BC suivante :

□ Base des règles

R1: SI A1 ET A2	ALORS P1
R2: SI A3	ALORS P1
R3 : SI P1 ET A4 ET P3	ALORS P2
R4 : SI A5 ET P4	ALORS P3 ET F1
R5 : SI P5	ALORS P3 ET F2
R6 : SI A4 ET P6	ALORS P3
R7 :SI A2	ALORS P4 ET F2 ET F3

□ Tableau des actes terminaux

- Actions à exécuter quand il y a dépilement

Symbole	Faits ajoutés	Faits enlevés
A1	F1, F2	Néant
A2	Néant	F1
A3	F1	Néant
A4	Néant	Néant
A5	F2	F4

□ Interprétation des règles :

- R1 : pour résoudre P1, il suffit d'exécuter l'acte A1 puis l'acte A2
- R4 : pour résoudre P3 sachant que F1 est établi, il suffit d'exécuter l'acte A5 puis de résoudre le problème P4.

□ BF (Fait établis) = {F3, F4}

□ PILEDESPROBLEMES (faits à établir) = Liste (P2)

VI. Modes d'invocation des règles

- **Partie déclencheur : membre gauche de la règle.**
- **Les règles sont invoquées en chaînage mixte, en ce sens que les filtres déclencheurs peuvent concerner simultanément les faits établis et les faits à établir.**
- **En l'absence de filtres relatifs à la BF, le MI fonctionne en chaînage arrière.**
- **Si le filtre correspondant à PPB est omis, le MI fonctionne en chaînage-avant.**
- **Le membre gauche de la règle est le corps de la règle :**
 - Il définit des actions à réaliser sur PPB et ultérieurement sur la BF.
 - Par exemple si le 1er élément de PPB est P1, la règle R1 peut être déclenchée : A1 et A2 sont placées respectivement en 1^{ère} et 2^{ème} position dans la PPB tandis que P1 en est retiré.

VI. Modes d'invocation des règles

■ **Planification ou génération de plans**

□ **A1, A2, A5... peuvent être vus comme des problèmes immédiatement résolubles :**

- d'où l'appellation de problèmes primitifs.
- Ils représentent aussi des actions élémentaires qui pourraient prendre place dans un plan d'actions :
 - d'où l'appellation actes terminaux.

- Lorsqu'un tel acte est insérer dans un plan en cours d'élaboration, il faut représenter les effets de cet acte : ajout et retrait de faits.

□ **Ajouter un fait est interprété comme : établir le fait.**

□ **Retirer un fait est interprété comme :**

- le fait n'est pas ou n'est plus établi.

VI. Modes d'invocation des règles

- **Un symbole est reconnu représenter un acte terminal s'il est présent dans la table des actes terminaux.**
 - Par exemple, on aurait pu coder la règle R4 sous la forme :
 - SI Ajouter F2 ET Retirer F4 ET P4 ALORS P3 ET F1
- **Lorsque après déclenchement d'une règle :**
 - un acte terminal se trouve en 1ère position (dite sommet) de PPB, il est aussitôt retiré (dépiler) et rangé en fin de la liste du plan d'action.
 - A ce moment là, les MAJ décrites dans la tables des actes terminaux sont exécutées.
 - Par exemple lorsque A5 est dépilé, on ajoute F2 à la BF et on retire F4.

VI. Modes d'invocation des règles

■ **Non-monotonie**

□ **Le MI tente d'engendrer des plans d'actions :**

- Il peut être amené à retirer du plan en cours d'élaboration un acte terminal précédemment introduit.
- Dans ce cas le mécanisme de retour-arrière du MI provoque la remise en cause des ajouts et retraits de faits éventuellement associés à l'acte terminal.
- Même si les règles ne commandent que des ajouts, la remise en cause de ces règles se traduit aussi par un retrait des faits établis.

VI. Modes d'invocation des règles

■ **Déclenchements en profondeur d'abord**

- **Le MI déclenche la 1ère règle de la BR dont le déclencheur est compatible avec PPB et BF.**
- **La règle déclenchée ensuite sera la 1ère de la BR compatible avec la situation créée par le déclenchement de la précédente :**
 - Il s'agit bien d'un développement des déclenchements en profondeur d'abord.
- **Si la PPB est vide :**
 - Le MI s'arrête : le problème initial a été résolu
 - L'état courant du Plan est fourni.

□ **Si PPB n'est pas vide mais qu'aucune règle n'est déclenchable**

- le MI tente systématiquement d'exécuter un retour-arrière :
 - c'est-à-dire d'appliquer une nouvelle règle au dernier problème non primitif qui a été retiré de PPB
 - et ainsi de suite.
- **Remarque :**
 - Ici il y a restauration du contexte de la PPB, de la BF et du Plan au moment d'un retour arrière.

VI. Modes d'invocation des règles

■ Exemple :

□ **BF (Fait établis) = {F3, F4}**

□ **PPB = Liste (P2)**

R1: SI A1 ET A2	ALORS P1
R2: SI A3	ALORS P1
R3 : SI P1 ET A4 ET P3	ALORS P2
R4 : SI A5 ET P4	ALORS P3 ET F1
R5 : SI P5	ALORS P3 ET F2
R6 : SI A4 ET P6	ALORS P3
R7 : SI A2	ALORS P4 ET F2 ET F3

□ **Tableau des actes terminaux**

- Actions à exécuter quand il y a dépilement

Symbole	Faits ajoutés	Faits enlevés
A1	F1, F2	Néant
A2	Néant	F1
A3	F1	Néant
A4	Néant	Néant
A5	F2	F4

VI. Modes d'invocation des règles

<p>Cycle 1 : EC = {R3}, RE = {R3} Après déclenchement de R3 PPB = (P1, A4, P3) BF = {F3, F4} Plan = ()</p> <p>Cycle 2 : EC = {R1, R2}, RE = {R1} Après déclenchement de R1 PPB = (A1, A2, A4, P3) BF = {F3, F4} Plan = (), plan vide Dépilement de A1, A2, A4 ; effet associé sur BF PPB = (P3) BF = {F2, F3, F4} Plan = (A1, A2, A4)</p> <p>Cycle 3 : EC = {R5, R6}, RE = {R5} Après déclenchement de R5 PPB = (P5) BF = {F2, F3, F4} Plan = (A1, A2, A4)</p>	<p>Cycle 4 : Aucune règle n'est déclenchée dans ce contexte. Le MI revient sur la résolution de P3 d'où PPB = (P3) BF = {F2, F3, F4} Plan = (A1, A2, A4)</p> <p>Cycle 5 : RE = {R6} Après déclenchement de R6 PPB = (A4, P6) BF = {F2, F3, F4} Plan = (A1, A2, A4) Dépilement de A4 ; effet associé sur la BF PPB = (P6) BF = {F2, F3, F4} Plan = (A1, A2, A4, A4)</p> <p>Cycle 6 : Aucune règle n'est déclenchée dans ce contexte. Le MI revient sur la résolution de P3 d'où PPB = (P3) BF = {F2, F3, F4} Plan = (A1, A2, A4)</p>
--	---

VI. Modes d'invocation des règles

<p>Cycle 7</p> <p>Aucune règle n'est déclenchée dans ce contexte.</p> <p>Le MI revient sur la résolution de P1 d'où</p> <p>PPB = (P1, A4, P3)</p> <p>BF = {F3, F4}</p> <p>Plan = ()</p> <p>Cycle 8 : RE = {R2}</p> <p>Après déclenchement de R2</p> <p>PPB = (A3, A4, P3)</p> <p>BF = {F3, F4}</p> <p>Plan = (), plan vide</p> <p>Dépilement de A3, A4 ; effet associé sur BF</p> <p>PPB = (P3)</p> <p>BF = {F1, F3, F4}</p> <p>Plan = (A3, A4)</p> <p>Cycle 9 : EC={R4, R6} RE = {R4}</p> <p>Après déclenchement de R4</p> <p>PPB = (A5, P4)</p> <p>BF = {F1, F3, F4}</p> <p>Plan = (A3, A4)</p>	<p>Dépilement de A5; effet associé sur BF</p> <p>PPB = (P4)</p> <p>BF = {F1, F2, F3}</p> <p>Plan = (A3, A4, A5)</p> <p>Cycle 10 : EC={R7} RE = {R7}</p> <p>Après déclenchement de R7</p> <p>PPB = (A2)</p> <p>BF = {F1, F2, F3}</p> <p>Plan = (A3, A4, A5)</p> <p>Dépilement de A2 ; effet associé sur BF</p> <p>PPB = (), PPB vide d'où arrêt</p> <p>BF = {F2, F3}</p> <p>Plan = (A3, A4, A5, A2)</p> <p>Pour résoudre le problème P2, il faut exécuter dans l'ordre les action qui sont dans le plan.</p>
---	--

- **Chaînage arrière : résolution d'un P_i**
- **Chaînage avant : Exécution d'une action A_i**

VI. Modes d'invocation des règles

■ Exercice 1

- **BF (Fait établis) = {F2, F4}**
- **PPB = Liste (P3)**

■ Exercice 2

- **BF (Fait établis) = {F1, F3}**
- **PPB = Liste (P3)**

R1: SI A1 ET A2	ALORS P1
R2: SI A3	ALORS P1
R3 : SI P1 ET A4 ET P3	ALORS P2
R4 : SI A5 ET P4	ALORS P3 ET F1
R5 : SI P5	ALORS P3 ET F2
R6 : SI A4 ET P6	ALORS P3
R7 :SI A2	ALORS P4 ET F2 ET F3

Symbole	Faits ajoutés	Faits enlevés
A1	F1, F2	Néant
A2	Néant	F1
A3	F1	Néant
A4	Néant	Néant
A5	F2	F4

VI. Modes d'invocation des règles

■ Correction Exercice 1

<p>Cycle 1 : $EC = \{R5, R6\}$, $RE = \{R5\}$ Après déclenchement de R5 $PPB = (A5, P4)$ $BF = \{F2, F4\}$ $Plan = ()$</p> <p>Cycle 2 : Aucune règle n'est déclenchée dans ce contexte. Le MI revient sur la résolution de P3 d'où $PPB = (P3)$ $BF = \{F2, F4\}$ $Plan = ()$</p> <p>Cycle 3 : $RE = \{R6\}$ Après déclenchement de R6 $PPB = (A4, P6)$ $BF = \{F2, F4\}$ $Plan = ()$ Dépilement de A4 ; effet associé sur BF $PPB = (P6)$ $BF = \{F2, F4\}$ $Plan = (A4)$</p>	<p>Cycle 4 : Aucune règle n'est déclenchée dans ce contexte. Blocage donc arrêt. P3 ne peut donc être résolu</p>
---	--

VI. Modes d'invocation des règles

■ Correction Exercice 2

<p>Cycle 1 : $EC = \{R4, R6\}$, $RE = \{R4\}$ Après déclenchement de R4 $PPB = (P5)$ $BF = \{F1, F3\}$ $Plan = ()$</p> <p>Cycle 2 : $EC = \{R7\}$, $RE = \{R7\}$ Après déclenchement de R7 $PPB = (A2)$ $BF = \{F1, F2, F3\}$ $Plan = ()$ Dépilement de A2; effet associé sur BF $PPB = ()$ arrêt $BF = \{F2, F3\}$ $Plan = (A5, A2)$</p>	<p>Pour résoudre le problème P3, il faut exécuter dans l'ordre les action qui sont dans le plan.</p>
--	--

VI. Modes d'invocation des règles

■ Algorithme

```
FONCTION ResoudreUnProbleme(leProbleme) : TEXTE
DEBUT
  PPB( liste ayant pour seul élément leProbleme)
  Plan ← liste vide
  UnContexte ← liste ayant pour élément PPB, BF et Plan
  SI(Resoudre(unContexte, BR) = 'échec') ALORS
    ResoudreUnProbleme ← 'échec'
  SINON
    DEBUT
      Editer(Plan)
      ResoudreUnProbleme ← 'succès'
    FIN
  FIN
```

```
FONCTION Resoudre(leContexte, IndicRegle, unContexte) : TEXTE
DEBUT
  SI(PPB = Vide) ALORS
    Resoudre ← 'succès'
  IndicRegle ← executerUnCycle(tête de PPB, IndicRegle)
  SI(IndicRegle = 'échec') ALORS
    Resoudre ← 'échec'
  unContexte ← liste ayant pour élément PPB, Plan tels qu'ils sont au retour de
  executerUnCycle
  SI(Resoudre(unContexte, BR) = 'succès') ALORS
    Resoudre ← 'succès'
  SI(IndicRegle = 'primitif') ALORS
    Resoudre ← 'échec'
  PPB ← 1er élément de leContexte
  BF ← 2ième élément de leContexte
  Plan ← 3ième élément de leContexte
  IndicRegle ← IndicRegle diminué de son 1er élément
  Resoudre ← Resoudre (leContexte, IndicRegle, unContexte)
FIN
```

VI. Modes d'invocation des règles

```
FONCTION executerUnCycle(leprobleme, lesRegles, unRegle) : TEXTE
DEBUT
  SI(leProbleme est présent dans la table des actes terminaux) ALORS
    DEBUT
      Supprimer la tête de PPB : c'est leProbleme
      Placer leProbleme en tête de Plan
      Exécuter les ajouts et retraits sur BF tels que décrit pour leProbleme dans la table des actes terminaux
      executerUnCycle ← 'primitif'
    FIN
  SI(lesRegle = vide) ALORS
    executerUnCycle ← 'échec'
  uneRegle ← 'la 1ère règle de lesRegle'
  SI(uneRegle est déclenchable) : ALORS (* déclencheur compatible avec leProbleme et BF)
    DEBUT
      Supprimer la tête de PPB
      Empiler les éléments du corps de uneRegle dans PPB (* le 1er en tête et en suivant *)
      executerUnCycle ← lesRegle
    FIN
  executerUnCycle ← executerUnCycle(leProbleme, lesRegles diminuées de uneregle, uneRegleà)
FIN
```

VII. Générateur de systèmes experts

□ 7.1. Présentation

- **Outils généraux dont le but est d'éviter le maximum de programmations.**
- **Accent mis particulièrement sur :**
 - **Résolution des problèmes de conception et de diagnostic ;**
 - **Deux modes de chaînage possible.**
 - **Interfaces utilisateurs, convivialité**
 - **Intégration de plusieurs modes de RC (objet, règle).**
 - **Aide au programmeur pendant le développement.**

VII. Générateur de systèmes experts

□ 7.2. Etude du générateur de SE GExpert

■ Développé à l'IUT de Bayonne (France)

- Langage Pascal

■ Amélioré par les ING/INFO2 depuis 2005

- Langage Delphi

- Interface graphique

■ Domaines d'application très variés :

- Médecine ;

- Diagnostic de panne ;

- Aide à la décision dans l'entreprise.

INP-HB/K. M. BROU

■ Installation

- Copier le dossier GExpert/Expert sur le disque C

- Lancer PROJET.EXE

VII. Générateur de systèmes experts

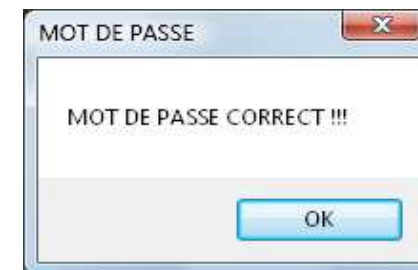
□ 7.3. Charger ou créer une BC



INP-HB/K. M. BROU

IA : Système expert

- **GExpert nécessite un mot de passe : base**



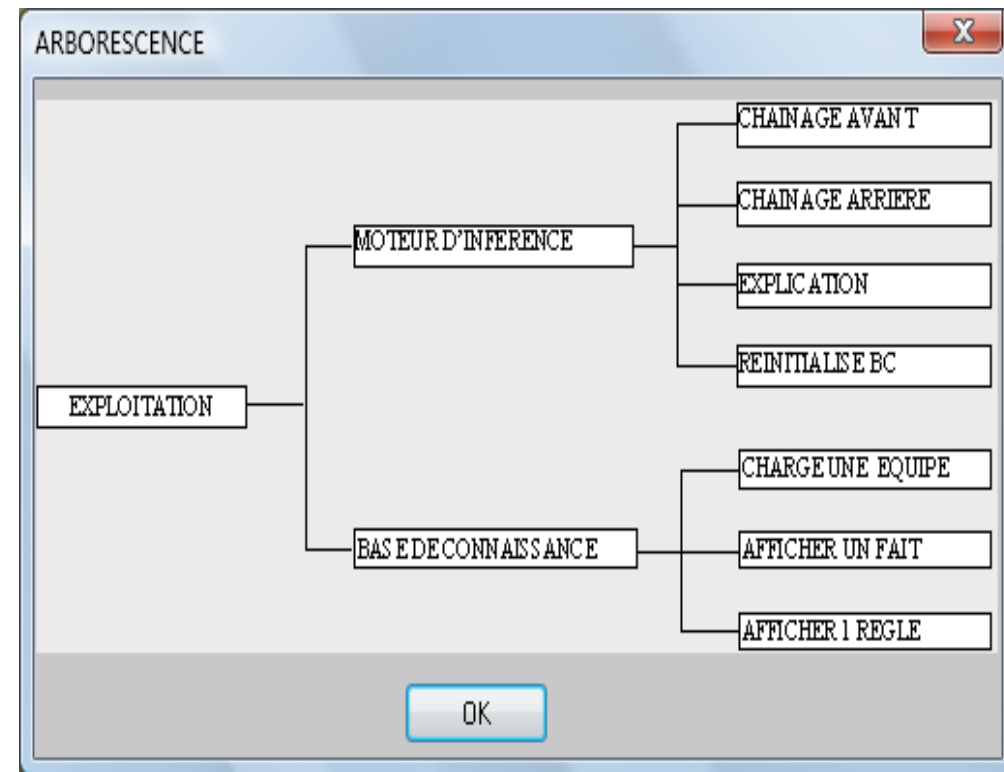
VII. Générateur de systèmes experts

■ Les menus :

- **SORTIR** : permet de quitter le logiciel ;
- **EXPERT** : donne la possibilité de réaliser les différents chainages ;
- **EXPLOITATION** : permet de gérer la BC ;
- **GUIDE** : permet lister les faits et règles d'une BC.

■ Structuration de GExpert

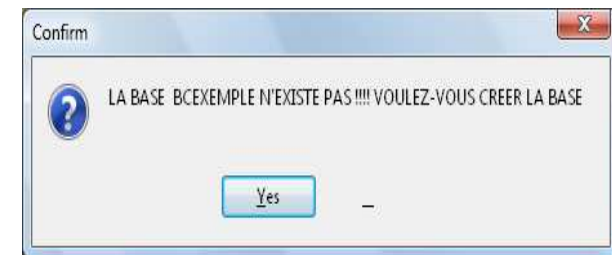
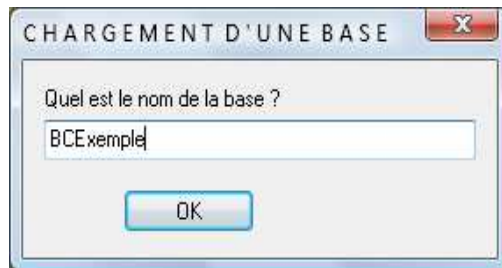
- Menu Guide/Arborescence



VII. Générateur de systèmes experts

■ Chargement d'une BC

- Soit par le menu Expert
 - **Expert/Charger une BC**
- Soit par le menu Exploitation
 - **Exploitation/Base de connaissance/Charger une BC**
- L'utilisateur tape le nom de la BC : BCExemple

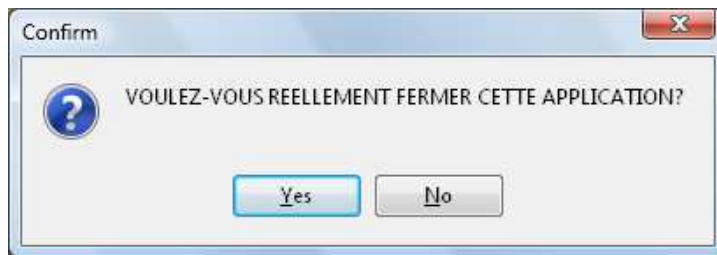


- **Si la BC existe elle sera chargée à l'appui sur le bouton Ok.**
- **Si la BC n'existe pas elle sera créée à l'appui sur le bouton Yes**

VII. Générateur de systèmes experts

■ Sauvegarde d'une BC

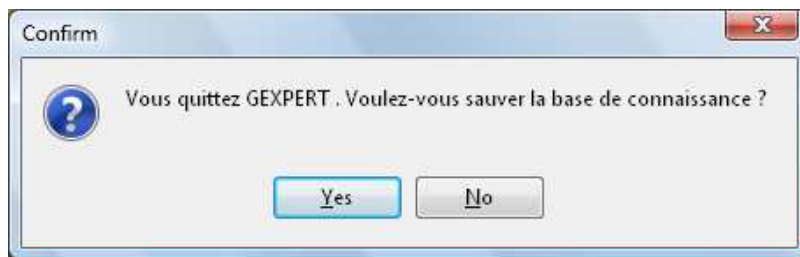
□ Menu Sortir



□ Cliquer sur Yes

□ Ou Menu Expert/Sauver la base de connaissance

□ Cliquer sur Yes



VII. Générateur de systèmes experts

- **Listage des BC**
 - **Menu Guide/Listage/Listage des bases de connaissances**

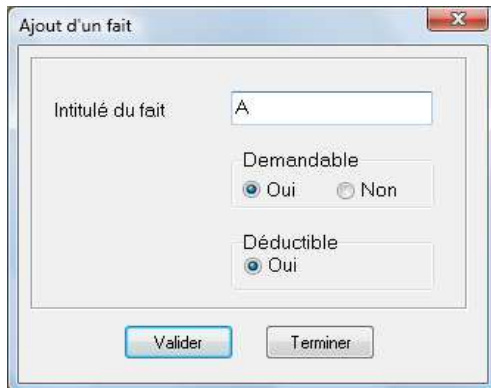


VII. Générateur de systèmes experts

□ 7.4. Gérer les faits

■ Saisie d'un fait

- Menu Expert/Gérer la base des faits/Ajouter un fait



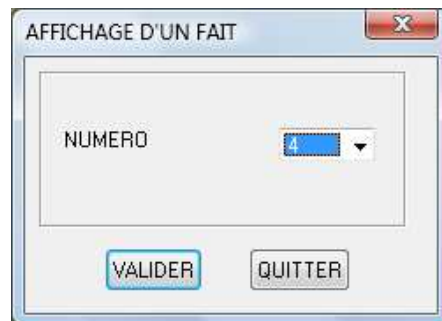
Fait	Demandable	Déductible
A	oui	oui
B	oui	non
C	oui	non
D	oui	oui
E	oui	oui
F	oui	oui
G	oui	non
K	oui	oui
S	non	oui
X	oui	non
Z	non	oui

- **Demandable** : le fait est une prémisse : fait B
- **Déductible** : le fait est une conclusion : fait Z
- **On peut avoir un fait qui soit :**
 - Demandable uniquement : fait B
 - Déductible uniquement : fait Z
 - Demandable et déductible : fait A
- **Valider** : ajoute le fait
- **Terminer** : termine la saisie des faits
- **Remarque** : À tout moment, on peut fermer une fenêtre en appuyant sur la touche "echap" ou en fermant la fenêtre

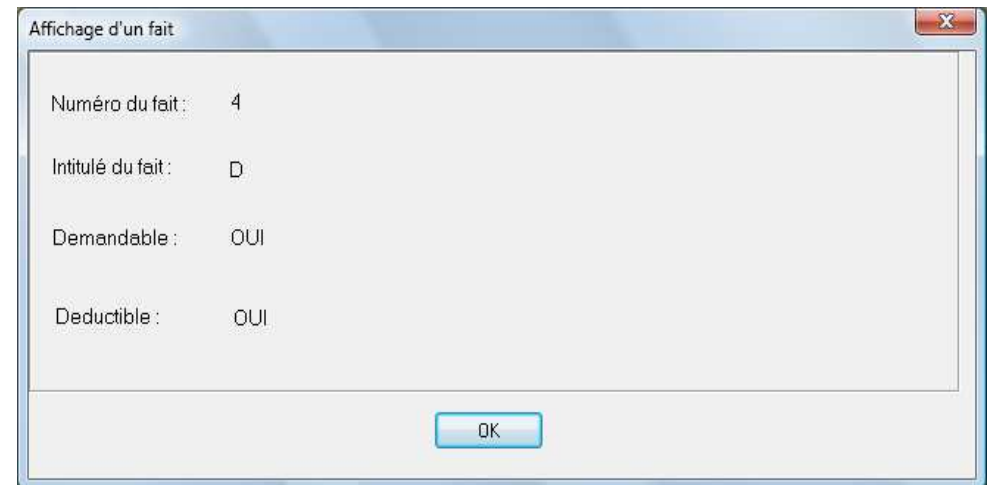
VII. Générateur de systèmes experts

■ Afficher un fait

- Menu Exploitation/Base de connaissance/Afficher un fait

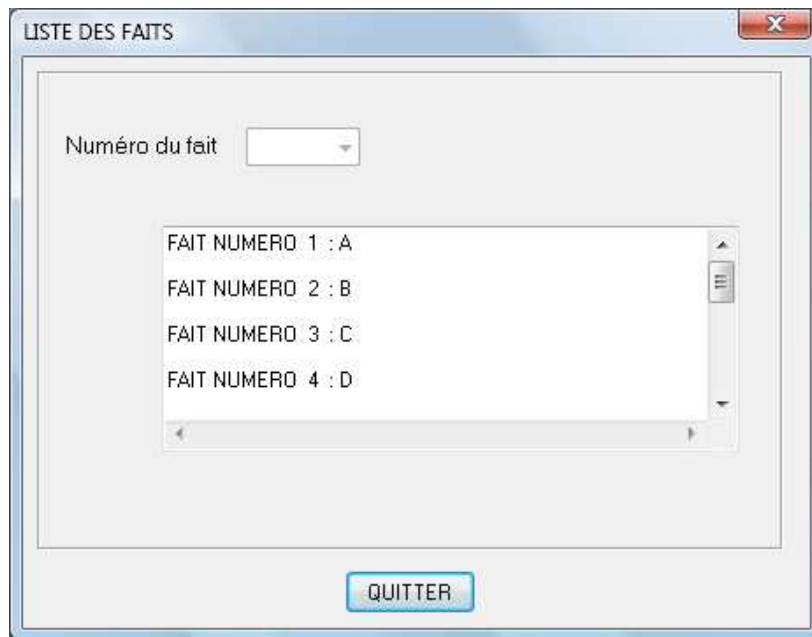


- Sélectionner le numéro du fait
- Cliquer sur le bouton Valider



VII. Générateur de systèmes experts

- **Lister tous les faits**
 - **Menu Guide/Listage/Listages des faits/Liste toute la base**

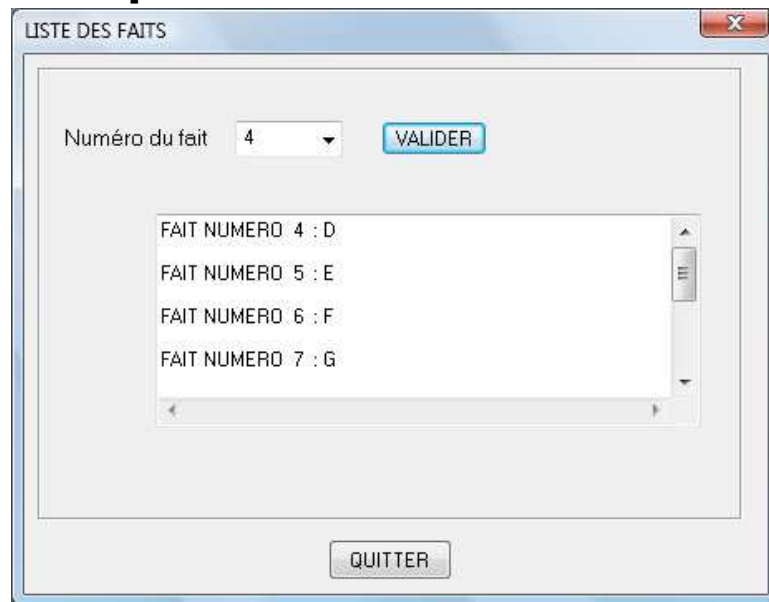


VII. Générateur de systèmes experts

■ Lister partiellement les faits

- Menu Guide/Listage/Listages des faits/Lister partiellement
- Sélectionner le numéro du fait
- Cliquer sur le bouton Valider

- Le fait sélectionné devient le premier fait de la zone d'affichage



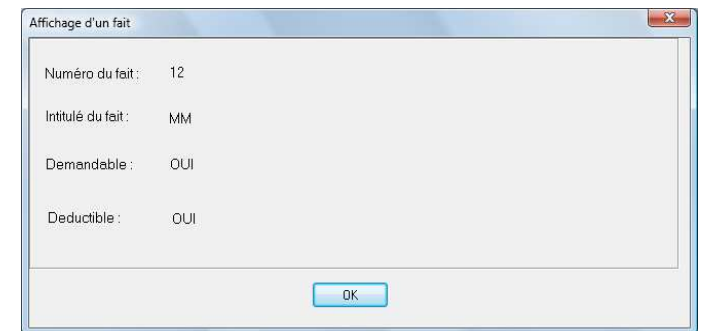
VII. Générateur de systèmes experts

■ Supprimer un fait

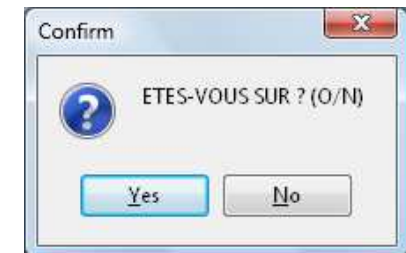
- Menu Expert/Gérer la base des faits/Supprimer un fait
- Sélectionner le numéro du fait à supprimer
- Cliquer sur le bouton Valider



- Cliquer sur le bouton Ok de la fenêtre d'affichage du fait



- Cliquer sur le bouton Yes de la fenêtre de confirmation



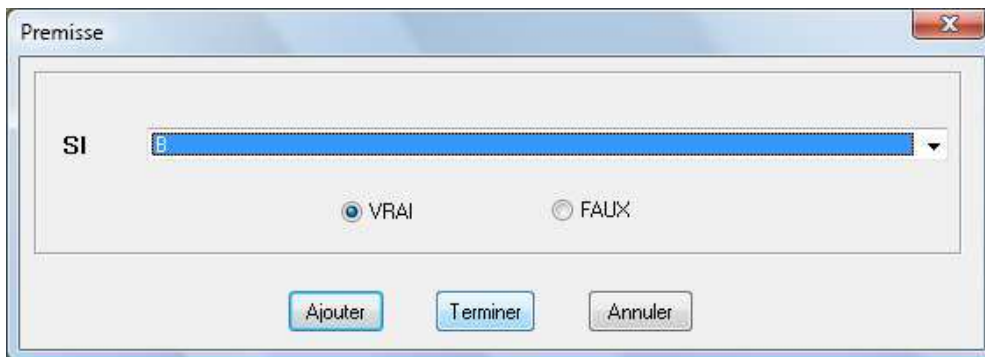
VII. Générateur de systèmes experts

□ 7.5. Gestion des règles

■ Saisie des prémisses

- Menu Expert/Gérer la base des règles/Ajouter une règle

- Cliquer sur le bouton Ajouter
- Cliquer sur le bouton Terminer pour arrêter la saisie des prémisses

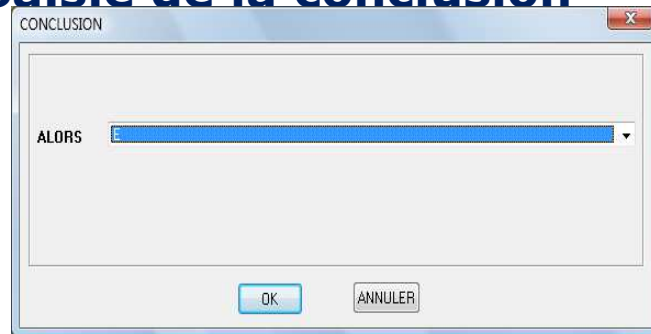


- Sélectionner les prémisses une à une

- VRAI : le fait est vérifié
- FAUX : le fait n'est vérifié

VII. Générateur de systèmes experts

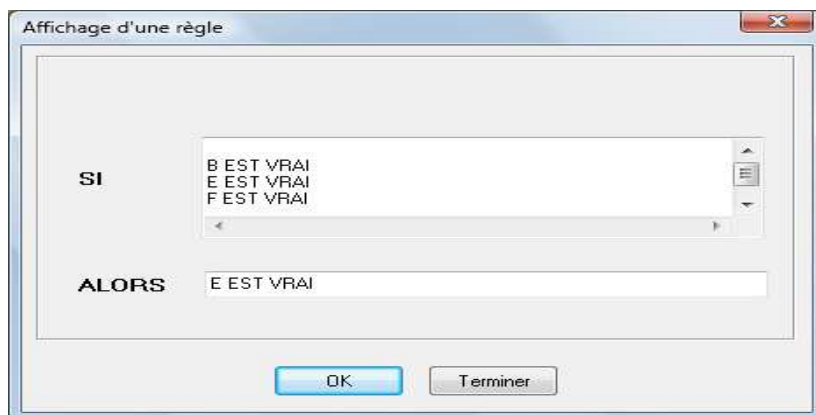
■ Saisie de la conclusion



- Sélectionner la conclusion
- Cliquer sur le bouton Ok



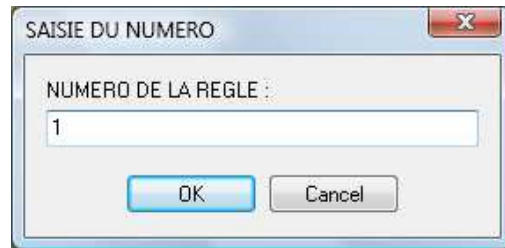
- Appuyer sur le bouton Yes pour ajouter la règle
- Saisir les règles suivantes :



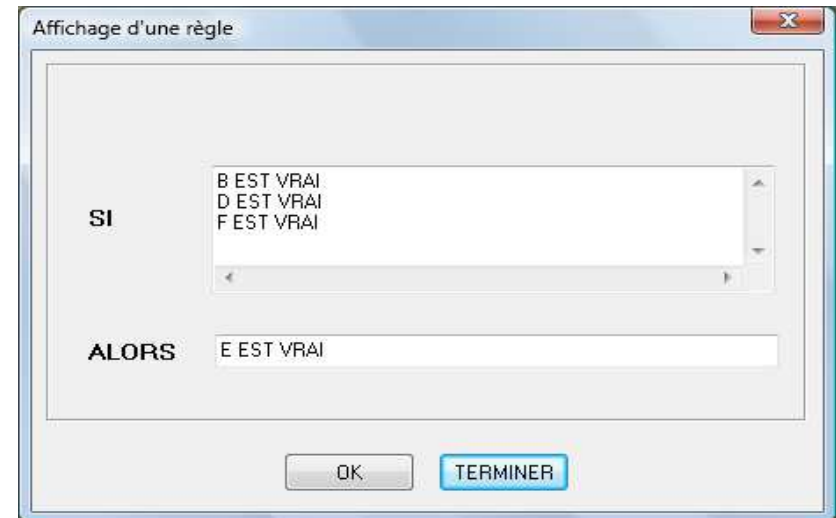
R1 : SI B ET D ET F	ALORS E
R2 : SI D ET G	ALORS A
R3 : SI C ET E	ALORS A
R4 : SI B	ALORS K
R5 : SI D	ALORS F
R6 : SI A ET K	ALORS S
R7 : SI C	ALORS D
R8 : SI K ET C	ALORS A
R9 : SI K ET C	ALORS Z
R10 : SI B ET K	ALORS D
R11 : SI X	ALORS Z

VII. Générateur de systèmes experts

- **Afficher une règle**
 - **Menu Exploitation/Base de connaissance/Afficher une règle**



- **Saisir le numéro de la règle**
- **Cliquer sur le bouton OK**



VII. Générateur de systèmes experts

- **Lister toutes les règles**
 - ▣ **Menu Guide/Listage/Listage des règles/Liste toute la base**

Affichage d'une règle

Numéro de la règle: VALIDER

SI: B EST VRAI
D EST VRAI
F EST VRAI

ALORS: E EST VRAI

SUIVANT QUITTER

- ▣ **Le bouton Suivant permet de parcourir la BR**

VII. Générateur de systèmes experts

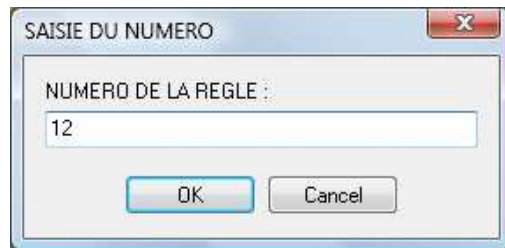
- **Lister partiellement les règles**
 - **Menu Guide/Listage/Listage des règles/Lister partiellement**
 - **Sélectionner le numéro d'une règle**
 - **Cliquer sur le bouton Valider**

- **Ne fonctionne pas**

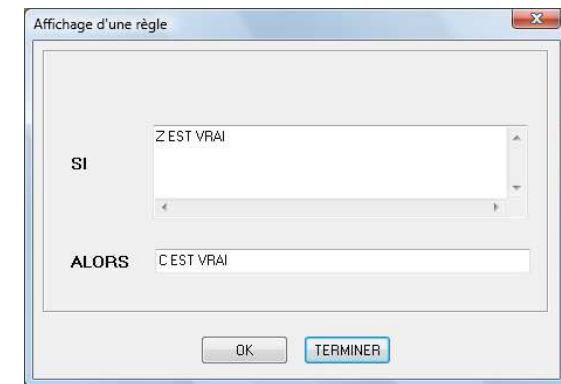
VII. Générateur de systèmes experts

■ Supprimer une règle

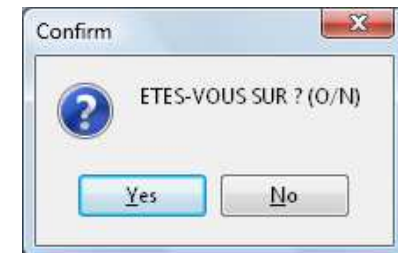
- Menu Expert/Gérer la base des règles/Supprimer une règle
- Sélectionner le numéro de la règle à supprimer
- Cliquer sur le bouton Valider



- Cliquer sur le bouton Ok de la fenêtre d'affichage de la règle



- Cliquer sur le bouton Yes de la fenêtre de confirmation



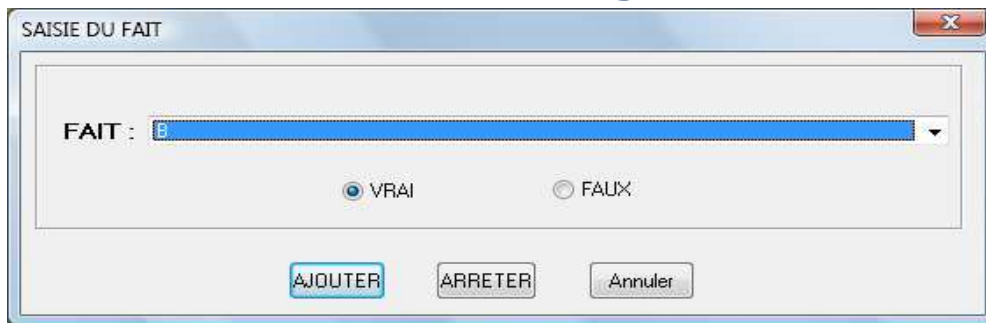
VII. Générateur de systèmes experts

□ 7.6. Chainage avant

■ Hypothèse

- La BF initiale : B, C
- Le but à prouver : S

■ Menu Exploitation/Moteur d'inférence/Chainage avant



SAISIE DU FAIT

FAIT : B

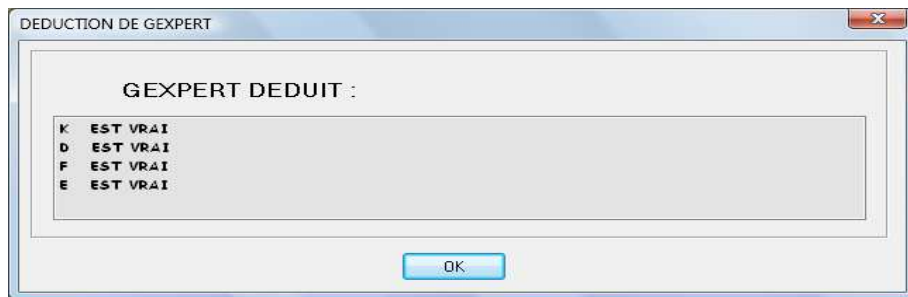
☒ VRAI ☐ FAUX

AJOUTER ARRETER Annuler

- Sélectionner les faits de la BF et cliquer sur le bouton Ajouter

VII. Générateur de systèmes experts

■ Le MI fait des déductions



■ Explication d'un déduction

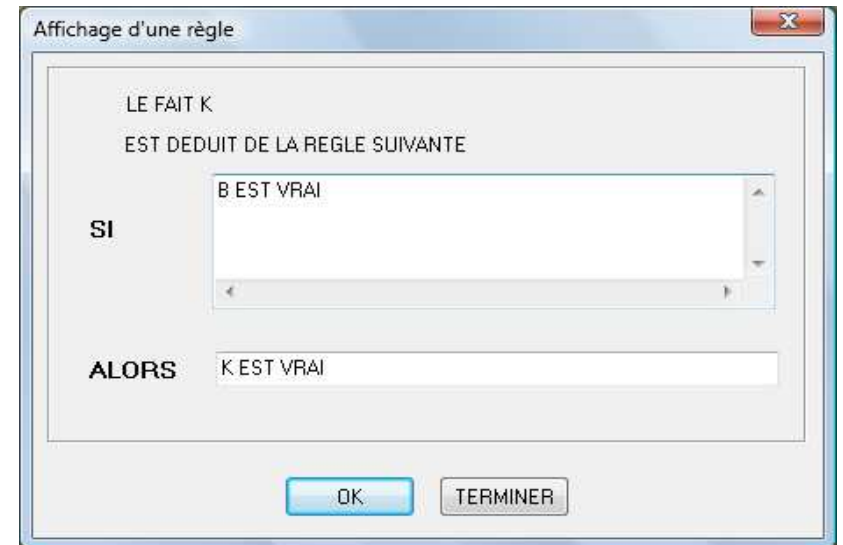
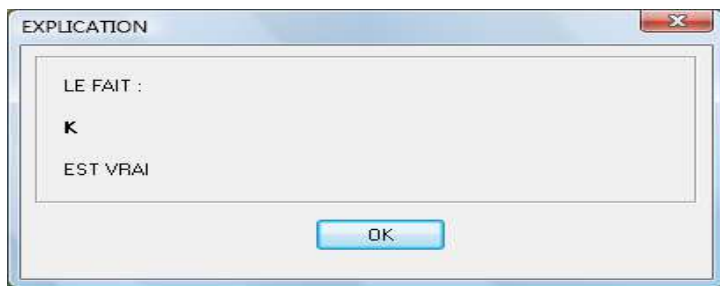
■ Menu Exploitation/Moteur d'inférence/Explication du raisonnement



■ Sélectionner le fait déduit et cliquer sur le bouton Valider

VII. Générateur de systèmes experts

■ Affichage des règles utilisées pour la déduction



VII. Générateur de systèmes experts

□ 7.6. Chainage arrière

■ Hypothèse

- La BF initiale : B, C
- Le but à prouver : S

■ Menu Exploitation/Moteur d'inférence/Chainage arrière

VALEUR DU FAIT

C

EST-IL Vrai , Faux ou Inconnu ?

☒ VRAI ☐ FAUX ☐ INCONNU

OK Cancel

- C est vrai car il est dans la BF

SAISIE DU FAIT

FAIT : S

VALIDER ARRETER Annuler

- Sélectionner le faits à prouver et cliquer sur le bouton Valider

VALEUR DU FAIT

G

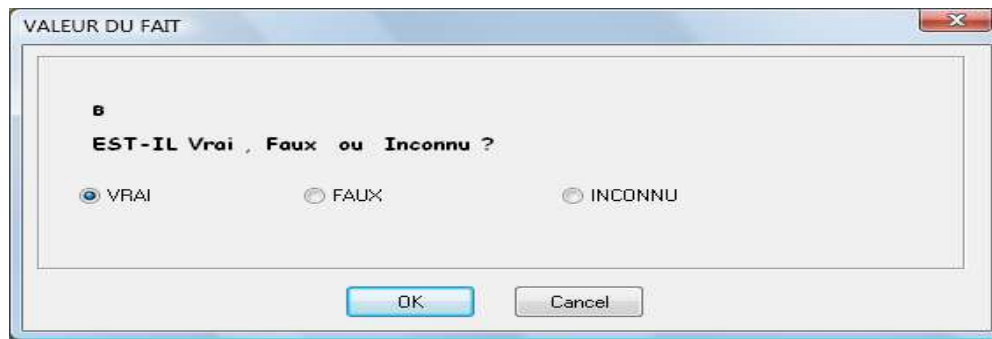
EST-IL Vrai , Faux ou Inconnu ?

☐ VRAI ☒ FAUX ☐ INCONNU

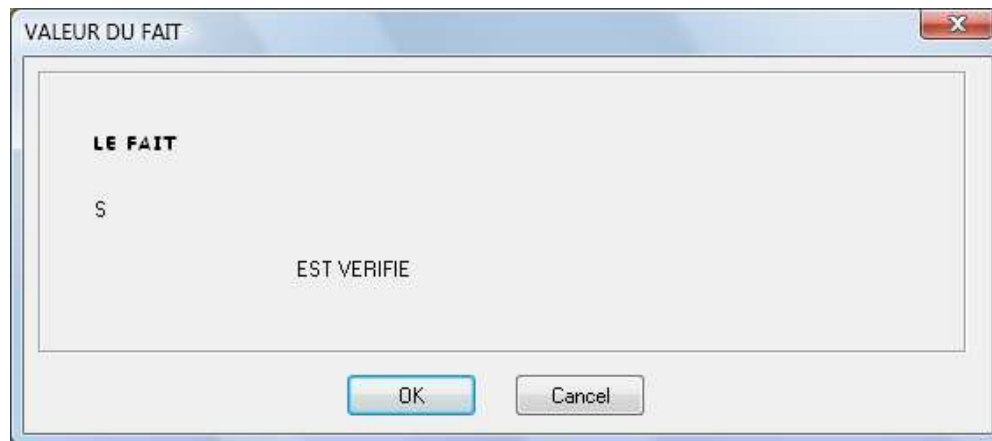
OK Cancel

- G est faux car il n'est pas dans la BF

VII. Générateur de systèmes experts

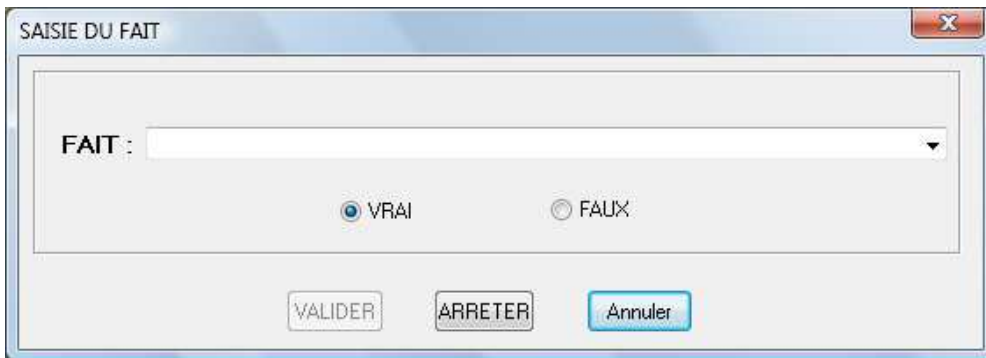


■ B est vrai car il est dans la BF



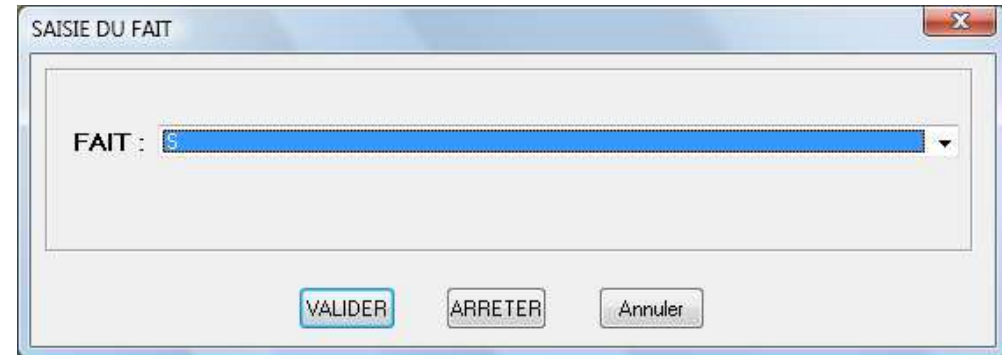
VII. Générateur de systèmes experts

- On peut faire d'autre déduction



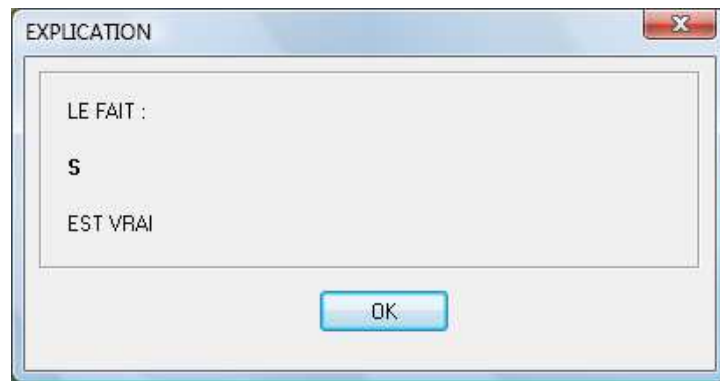
- Explication du raisonnement

- Menu Exploitation/Moteur d'inférence/Explication du raisonnement



- Sélectionner le fait et cliquer sur le bouton Valider

VII. Générateur de systèmes experts



- Un clic sur le bouton OK affiche la règle qui a permis de déduire S



- On sait que la BC contient seulement B et C, donc les faits A et K ont été déduits
- Un clic sur le bouton OK affiche la fenêtre qui permet demander l'explication d'une déduction.
- Suivre le même processus pour avoir l'explication des faits déduits A et K.

VII. Générateur de systèmes experts

- **7.7. Projet BC MedTrad**
 - **BC en médecine traditionnelle**

Bibliographie

□ Livres

- "Les Systèmes Experts, Principes et exemples", H Farreny, CEPADUES Edition
- "Exercices programmés d'intelligence artificielle", H Farreny, Edition Masson

□ Support de cours Web

- [poly-4-Systemes-Experts.pdf](#)