

UNIVERSITÉ LUMIÈRE LYON 2
UNIVERSITÉ CLAUDE BERNARD LYON 1
ÉCOLE NORMALE SUPÉRIEURE DE LYON

MODÉLISATION DES PROCESSUS
COGNITIFS DANS LA PERCEPTION

MÉMOIRE DE MAÎTRISE EN SCIENCES COGNITIVES

responsable de la formation :

Professeur O. KOENIG

présenté par :

Fabrice MUHLENBACH

réalisé sous la direction du :

Professeur D.A. ZIGHED

Equipe de Recherche en
Ingénierie des Connaissances, Lyon 2

juin 1997

Remerciements

En premier lieu, je tiens à remercier le Professeur Djamel Abdelkader Zighed pour m'avoir accueilli au sein de l'*Equipe de Recherche en Ingénierie des Connaissances*. Il a su me faire confiance pour la réalisation d'un projet qui lui tenait à cœur et a mis à ma disposition les ressources matérielles et logicielles qui m'ont permis de mener à bien le travail commencé dès l'été 1996.

Je remercie aussi chaleureusement Marc Sebban avec qui j'ai collaboré durant cette année universitaire. Cette coopération fut l'occasion de nombreux moments de joie, notamment lorsqu'il soutint avec succès sa thèse au mois de décembre. Merci aussi à sa famille qui m'accueillit durant le mois d'août pour que nous puissions avancer nos travaux au frais, à la campagne, alors que l'Université désertée cuisait sous le soleil lyonnais.

Au cours de mes journées de programmation passées au laboratoire ERIC, je me suis retrouvé bien souvent face à des problèmes informatiques. Les conseils de Ricco Rakotomalala me furent très précieux pour sortir des multiples sessions de « débogage » où je m'étais empêtré. Qu'il soit ici remercié pour avoir toujours été prompt à répondre à mes questions par ses avis éclairés.

Un grand merci encore à tous les autres membres du laboratoire ERIC, qu'ils soient professeurs, maîtres de conférence, docteurs, doctorants ou simplement des stagiaires présents durant les vacances.

**Ce rapport de recherche sous forme postscript ainsi
que les fichiers d'installation du logiciel SelfMind sont
disponibles sur le serveur d'ERIC à l'adresse suivante :
<http://eric.univ-lyon2.fr/softs.html>**

Sommaire

INTRODUCTION	4
PREMIÈRE PARTIE : BASES THÉORIQUES DU MOTEUR D'INFÉRENCE "SELMIND"	7
REPRÉSENTATION DES CONNAISSANCES, LOGIQUES ET SYSTÈME EXPERT	8
LOGIQUE MULTIVALUÉE, INCERTITUDE ET IMPRÉCISION	10
SELMIND : UN OUTIL DE GESTION DES CONNAISSANCES IMPARFAITES	15
SECONDE PARTIE : APPLICATIONS DU MOTEUR D'INFÉRENCE	
SELMIND DANS LE DOMAINE DE LA PERCEPTION	28
APPLICATION 1 : GÉNÉRATION DE PROPRIÉTÉS OLFACTIVES ASSOCIÉES À UNE ODEUR	29
APPLICATION 2 : RECONNAISSANCE DE CHIFFRES MANUSCRITS	34
<i>Méthode</i>	34
<i>Résultats</i>	36
<i>Discussion</i>	37
TROISIÈME PARTIE : PROPOSITIONS MÉTHODOLOGIQUES POUR	
UNE MODÉLISATION DES ASPECTS COGNITIFS DANS LA PERCEPTION	41
DES EXPÉRIENCES DE PSYCHOLOGIE COGNITIVE COMME SOURCE D'INFORMATION	
POUR LA CONSTITUTION DE BASES DE CONNAISSANCES	42
EXTRACTION AUTOMATIQUE DE CONNAISSANCES À PARTIR DE DONNÉES	44
SELMIND : PARTIE SYMBOLIQUE DU MODÈLE HYBRIDE DE PERCEPTION INTELLIGENTE "SIMPL"	45
CONCLUSION	47
ANNEXE 1 : LISTE DES RÈGLES GÉNÉRÉES PAR SIPINA, CART ET ID3 SUR LES ODEURS	50
ANNEXE 2 : LISTE DES RÈGLES GÉNÉRÉES PAR SIPINA SUR LES CHIFFRES MANUSCRITS.	
PAR F. MUHLENBACH & M. SEBBAN	55
RÈGLES ENTRE DESCRIPTEURS SÉMANTIQUES	55
RÈGLES ENTRE DESCRIPTEURS PERCEPTIFS	57
RÈGLES ALLANT DE DESCRIPTEURS PERCEPTIFS VERS DES DESCRIPTEURS SÉMANTIQUES	59
RÈGLES ABOUTISSANT AUX LABELS DES CHIFFRES MANUSCRITS	62
RÈGLES SUPPLÉMENTAIRES	64
RÉFÉRENCES	65

Introduction

Les sciences cognitives forment un champ pluridisciplinaire où les développements récents de l'informatique et de l'intelligence artificielle apportent leurs lumières sur des problèmes classiques de psychologie et de philosophie de l'esprit. Les anciennes questions concernant le langage, le raisonnement ou notre manière de voir le monde, voient une nouvelle forme d'investigation à travers les sciences de la modélisation.

Parmi les différentes voies de modélisation existantes, nous avons adopté l'approche de l'intelligence artificielle sous sa forme dite « classique » (Pylyshyn, 1989). Pour Haugeland (1985), l'intelligence artificielle se fonde sur le fait que l'intelligence — *artificielle* ou *naturelle* — ne dépend que de la manière dont un système est organisé en tant que manipulateur de symboles. Il en résulte que les indications de niveau inférieur, comme la constitution matérielle des symboles ou leurs formes précises, n'ont rien à voir avec l'éventuelle intelligence du système. Les symboles ne doivent être conformes qu'à des spécifications abstraites du niveau supérieur. De ce fait, les indications sur la nature électronique ou physiologique de la structure sous-jacente du système n'ont pas de rôle intéressant à ce niveau de description. Or comme l'ordinateur est un objet manipulateur de symboles, il est l'outil de simulation adéquat pour l'étude de la cognition.

Pour dire qu'un programme informatique fait preuve « d'intelligence », comme le rappelle Johnson-Laird (1983), il n'existe pas de réelle méthode. Même le test de Turing de 1950 que cite cet auteur peut aisément être remporté avec succès par un programme astucieux. Le test de Turing — connu sous le nom de « jeu de l'imitation » — consiste à dire qu'un programme est « intelligent » quand une personne utilisant un ordinateur ne peut déclarer si les réponses lui sont données par l'ordinateur ou si elle est en communication avec une autre personnes reliée à un terminal. Johnson-Laird parle notamment du programme ELIZA réalisé par Weizenbaum en 1976 qui engageait un discours thérapeutique et qui a réussi à tromper des sujets humains. Des « patients » ayant participé à cette expérience prétendaient même n'avoir jamais été aussi bien compris ! Cependant ce programme n'était pas « intelligent » dans le sens où nous l'entendons et il ne révélait rien de l'architecture cognitive, tout juste montrait-il que l'être humain pouvait être abusé par certains modes d'interaction verbale. Le programme ELIZA fonctionnait en effet suivant le principe de la dissimulation, non sur la simulation. C'est ainsi que notre approche consistera à prendre les chemins de l'intelligence artificielle pour construire un modèle cherchant à *simuler* certains traitements réalisés dans une activité de perception, en nous basant principalement sur l'hypothèse forte d'une identité entre la pensée et le calcul.

En outre, le travail réalisé durant cette année s'inscrit dans le cadre de la théorie *fonctionnaliste*. Selon cette théorie, la pensée peut être étudiée indépendamment du cerveau. Pylyshyn (1984) considère d'ailleurs qu'une description *cognitive* est une description qui ne pourrait être exprimée par aucune autre description : que celle-ci soit neurophysiologique, comportementale ou phénoménologique. Il soutient que la cognition est une forme de computation et prend la métaphore de l'ordinateur sous son aspect dur, à savoir que les contenus sémantiques des états mentaux sont encodés de la même manière que sont encodées les représentations d'un ordinateur.

Il ne s'agit là pas d'un parti pris de notre part, car nous reconnaissons l'intérêt des travaux réalisés en neurophysiologie, cependant nous nous sommes focalisés sur les aspects sémantiques des traitements réalisés dans la perception, aussi n'avons-nous pas cherché à tenir compte des aspects biologiques du système nerveux. Au contraire d'une approche purement cognitive (e.g. Fodor & Pylyshyn, 1988), nous sommes simplement de l'avis qu'une modélisation la plus plausible de la cognition humaine n'est possible qu'en tenant compte à la fois des traitements de bas niveau — dont la simulation peut se faire par des approches associées au fonctionnement neuronal telles que le connexionnisme — et des traitements plus élaborés faisant intervenir des représentations mentales porteuses de sens.

Selon Fodor (1983), l'esprit humain est constitué de systèmes « modulaires » — des dispositifs cognitifs spécialisés dans le traitement d'un type limité d'information venant de l'environnement en élaborant des représentations à partir des sensations brutes — et de systèmes « centraux » — c'est-à-dire la pensée proprement dite. L'avenir de la recherche en sciences cognitives qu'il dresse en conclusion de son ouvrage est assez sombre. En effet, même s'il considère que les travaux sur la psychologie des processus périphériques, et en particulier sur les travaux au sujet du langage et de la perception, ont été dans l'ensemble couronnés de succès, il trouve que les tentatives de développement de modèles généraux de résolution de problèmes par l'approche de l'intelligence artificielle classique n'ont globalement eu que peu de retombées importantes. Fodor soutient que si la construction d'une machine intelligente n'a été possible, cela est dû au peu de connaissances que nous avons sur les traitements d'informations dans les systèmes centraux, même si on sait déjà beaucoup de choses sur les transformations qui font que les informations sensorielles prennent une forme appropriée pour les traitements ultérieurs. Notre recherche consistait justement à essayer de voir si une approche employant un système expert avec certaines propriétés particulières avait quelques chances de nous en apprendre davantage sur ces étapes sémantiques.

Pylyshyn (1984) — répondant à une critique de Dreyfus (1979) qui déclarait que l'ordinateur ne peut reproduire la logique du sens commun — indique qu'une description est associée à l'interprétation. Cette interprétation s'inscrit dans le cadre de la logique du sens commun que doit introduire, si nécessaire, le concepteur du système.

Selon cet auteur, pour qu'une modélisation utilisant l'approche des systèmes experts soit adéquate, il faut donc enseigner à l'ordinateur le résultat de cette expérience vécue, dûment traduit dans un langage. Notre travail s'est intéressé à ce sujet et nous avons cherché à reproduire la logique du sens commun en employant des approches combinant la psychologie expérimentale et l'ingénierie des connaissances.

Nous avons choisi de présenter le travail réalisé cette année en trois parties.

La première partie de ce rapport concerne les aspects théoriques sur lesquels nous nous sommes basés pour réaliser notre moteur d'inférence. Il y sera question des différentes logiques non classiques — floue et multivaluée — que nous présenterons brièvement avant de décrire le mode de représentation des connaissances que nous avons choisi d'adopter. Nous développerons certains algorithmes de calcul programmés dans SelfMind et nous exposerons le fonctionnement du logiciel sur un exemple pédagogique.

Puis nous avons cherché à employer notre moteur d'inférence en tant que modèle des processus cognitifs dans la perception. La seconde partie traitera ainsi des applications perceptives sur lesquelles nous avons travaillé : *l'olfaction* (générer des propriétés odorantes associées à une odeur) et la *vision* (dénommer un chiffre manuscrit à partir de ses propriétés physiques visuelles).

La troisième partie aura pour but de proposer une manière de concevoir des bases de connaissances afin de modéliser différents aspects de la connaissance humaine. Pour cela, nous présenterons les liens qu'entretiennent les systèmes experts développés avec différents axes de recherche d'ingénierie des connaissances. Tout d'abord, nous présenterons une expérience réalisée pour recueillir des informations sur certains aspects de la cognition. Ensuite, nous traiterons de l'apprentissage symbolique et des outils d'extraction de règles à partir de données comme source de connaissances imparfaites utilisables par notre système. Nous parlerons enfin d'un modèle hybride — car comportant des éléments numériques et symboliques — dans lequel notre moteur d'inférence s'est intégré et où il assure la gestion des traitements de haut niveau.

Première partie :

**Bases théoriques du moteur
d'inférence "SelfMind"**

Représentation des connaissances, logiques et système expert

Dans la problématique de la perception, Bonnet, Botte, et Molnar (1986) voient en l'intelligence artificielle une approche des niveaux plus complexes que ceux abordés par la psychophysique sensorielle. Ces auteurs citent notamment les travaux de Marr de 1982 qui énonçait 3 différents niveaux de traitements dans la perception, visuelle en l'occurrence. Alors que le premier niveau de traitement — appelé « calculatoire » — n'est pas réductible à un ensemble de règles logiques mais définit un ensemble de règles spécialisées, les niveaux plus élevés font intervenir des représentations que l'on peut chercher à modéliser par les techniques de l'intelligence artificielle. Le second niveau consiste ainsi en un choix de représentations qui rend explicite le type d'informations traitées par les processus et qui conduit à une description des informations. Ces représentations seront formalisées dans le troisième niveau en un algorithme qui sera concrètement réalisé par le système considéré, que celui-ci soit un organisme ou un ordinateur. C'est pourquoi, en nous situant dans cette approche, nous avons pensé que l'emploi d'un moteur d'inférence serait un bon moyen d'arriver à modéliser les aspects cognitifs des processus perceptifs.

Par construction, un moteur d'inférence est un logiciel qui a la particularité de manipuler de façon logique les connaissances dont il dispose. Ces connaissances, classiquement, sont issues de l'interview d'experts d'un domaine donné et sont organisées sous la forme de règles de production pour constituer le système expert. À partir d'une base de faits décrivant une situation donnée, le moteur d'inférence emploie la base de règles pour pouvoir apporter de l'information suivant un enchaînement logique.

Pylyshyn (1993) présente ainsi un système expert capable d'effectuer des diagnostics médicaux — comme le système Mycin qui diagnostique des maladies infectieuses — en décrivant son raisonnement et en recommandant certains traitements sur la base de la connaissance des symptômes et des actions de certains médicaments.

Mais comme le rappelle cet auteur, un système expert fait souvent figure d'*idiot savant*. Sachs (1995) décrit ces personnes atteintes de troubles autistiques qui sont inadaptées dans la vie de tous les jours mais qui font preuve par ailleurs d'extraordinaires capacités dans des domaines restreints (la musique, le dessin ou les calculs mentaux complexes). Cependant nous ne cherchons pas à modéliser des hauts niveaux d'expertise, mais souhaitons voir ce que peut apporter un moteur d'inférence en manipulation des symboles issus de modalités sensorielles différentes.

Les moteurs d'inférence des systèmes experts fonctionnent suivant des principes logiques. Mais, en tant qu'êtres humains, procédons-nous de même ? Johnson-Laird (1983) rappelle en effet que nous n'avons pas de « tables de vérité » dans la tête, et qu'il nous arrive de procéder à de faux raisonnements logiques.

Cependant, il existe d'autres logiques que la logique booléenne classique (en « tout ou rien »). Par exemple, la théorie des sous-ensembles flous (Bouchon-Meunier, 1995) développée par Zadeh en 1965 joue sur des degrés d'appartenance des individus à une classe donnée avec des valeurs comprises entre 0 et 1. Au moyen de fonctions d'appartenance, il est ainsi possible d'employer des connaissances plus proche du langage naturel. Par exemple, des notions métriques de proximité d'un lieu par des termes comme « proche », « moyennement distant » ou « éloigné ». Pourtant, si la formalisation de ces fonctions d'appartenance reste assez délicate, la logique floue propose quand même un cadre où incertitude et imprécision peuvent coexister. En combinant plusieurs propositions floues, on peut aboutir à une règle floue, ce qui permettra de procéder à des raisonnements approximatifs. Nous pensons que ce mode de raisonnement rend mieux compte des aspects naturels des opérations qui s'effectuent sur les représentations mentales qu'un raisonnement booléen classique.

Logique multivaluée, incertitude et imprécision

Pour rendre compte des concepts flous, nous avons employé la logique multivaluée et la théorie des probabilités symboliques.

Tout comme Akdag et Khoukhi (1994), nous distinguerons principalement deux sortes d'imperfection des connaissances : l'incertitude et l'imprécision. Citant Zadeh en 1987, Khoukhi (1996) reprend l'aspect complémentaire de ces deux notions : « à mesure que la complexité d'un système s'accroît, notre aptitude à formuler des affirmations précises mais signifiantes sur son comportement diminue jusqu'à un seuil au-delà duquel la précision et le sens deviennent mutuellement exclusifs ».

L'imprécision des informations est associée à la valeur qualitative des propositions des mots de la langue. Nous nous exprimons en effet en décrivant notre monde avec un certain flou, car nous partageons avec nos semblables des idées qui n'ont pas besoin d'être très précises. Des éléments sont considérés imprécis quand la vérité de connaissance qui s'applique à eux est partielle. Tel était le cas de notre exemple des distances avec les termes « proche », « moyennement distant » ou « éloigné » qui sont imprécis par rapport à une proposition quantitative comme « situé à 150 mètres ».

Quant à l'incertitude des informations, elle est liée à la croyance que les propositions dont il est fait référence existent réellement. Cette notion se représente très bien par des valeurs probabilistes. Dans le langage ordinaire, l'incertitude se rencontre par des expressions telles que « il est probable que... », « il est possible que... » ou « il y a des chances que... ».

Pour pouvoir gérer les notions d'imprécision et d'incertitude, nous avons employé l'approche de la logique multivaluée, ou *multivalente*. La logique multivaluée fournit en effet un cadre dans lequel la nuance est possible (Akdag & Khoukhi, 1994), notion fondamentalement humaine. Le principe de la multivalence réside en l'introduction de plusieurs valeurs, des degrés qui vont moduler les informations afin de rendre l'imperfection naturelle de la langue. La logique des propositions multivalente traitera non plus de propositions vraies ou fausses dans l'absolu, mais vraies à un certain degré.

L'imperfection de la langue peut se représenter sous forme de propositions multivalentes par simple transformation de la proposition « X est αA » est vraie en « X est A » est α -vraie.

Par exemple, « Il est *plutôt* grand » se transforme en « Il est grand est *plutôt vrai* ».

Cette généralisation de la logique classique est applicable à des énoncés vrais à un certain degré. Pour manipuler ces degrés de vérité, nous avons repris les tables de vérité élaborées par Khoukhi (1996) qui permettent d'effectuer les calculs sur les propositions. Nous ferons ainsi appel à un treillis $L_M = \{\tau_0, \dots, \tau_i, \dots, \tau_{M-1}\}$ avec $L_7 = \{\text{indéfini, pas vraiment précis, assez précis, moyennement précis, plutôt précis, quasiment précis, tout à fait précis}\}$. Ces degrés seront propagés lors des enchaînements de déduction par *modus ponens généralisé*. Nous ne gérerons que des règles simples (règles à une seule prémisse) ou conjonctives (règles où les propositions en prémisse sont reliées par l'opérateur logique « et »), toutes règles disjonctives (règles où les propositions en prémisse sont reliées par l'opérateur logique « ou ») pouvant être transformées en conjonctions de règles.

Par règle forte, on entend toute règle à une seule prémisse, connue avec un degré d'imprécision appartenant à notre treillis. Les modus ponens de ces règles sont présentés dans le tableau suivant :

Règles fortes	Degré d'imprécision de déduction
$\alpha A \rightarrow \beta B$ αA	βB
$\alpha A \rightarrow B$ $\alpha' A$	βB $t_\beta = V(\tau_\alpha, \tau_{\alpha'})$
$\alpha A \rightarrow \beta B$ $\alpha' A$	λB $t_\lambda = \max(t_0, t_{\mu + \beta - M + 1})$ $t_\beta = V(\tau_\alpha, \tau_{\alpha'})$

Dans la colonne de gauche du tableau ci-dessus, il y a le types de règles et de faits observés, et les conclusions obtenues par *modus ponens généralisé* se trouvent dans la colonne de droite.

Avec une règle ayant une prémisse imprécise αA et une conclusion imprécise βB , si on est en présence d'un fait imprécis de type $\alpha' A$, alors on déduira un fait B avec un degré d'imprécision λ qui sera déterminé par l'application du modus ponens généralisé tenant compte de la relation de voisinage entre α et α' .

Les relations de voisinage entre les degrés d'imprécision sont données par le tableau suivant :

V	τ_0	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6
τ_0	τ_6	τ_5	τ_4	τ_3	τ_2	τ_1	τ_0
τ_1	τ_5	τ_6	τ_5	τ_4	τ_3	τ_2	τ_1
τ_2	τ_4	τ_5	τ_6	τ_5	τ_4	τ_3	τ_2
τ_3	τ_3	τ_4	τ_5	τ_6	τ_5	τ_4	τ_3
τ_4	τ_2	τ_3	τ_4	τ_5	τ_6	τ_5	τ_4
τ_5	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6	τ_5
τ_6	τ_0	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6

L'éloignement ou la proximité de α et α' va influencer la décision sur la conclusion. Le degré de voisinage sera le plus grand (τ_6) pour des degrés identiques et diminuera graduellement en fonction du tableau.

La conclusion B inférée sera affectée d'un degré d'imprécision λ tel que :
 $t_\lambda = \max(t_0, t_{\mu+\beta-M+1})$ avec $t_\beta = V(\tau_\alpha, \tau_{\alpha'})$.

Cependant, avec notre moteur d'inférence, nous voulons pouvoir gérer des règles plus complexes que les règles simples fortes. Notre base de règles sera aussi constituée de règles fortes conjonctives, des règles conjonctives où les prémisses sont affectées d'un degré de précision. Le modus ponens de ces règles est présenté dans le tableau suivant :

Règles fortes conjonctives	Degré d'imprécision de déduction
$\alpha_1 A_1 \wedge \dots \wedge \alpha_n A_n \rightarrow \beta B$ $\alpha'_1 A_1, \dots, \alpha'_k A_k$	λB $t_\lambda = \max(t_0, t_{\mu+\beta-M+1})$ où $t_\mu = t_r$ avec $r = \left[\frac{\sum_{i=1}^k j}{n} \right]$ où $t_j = V(\tau_{\alpha_j}, \tau_{\alpha'_j})$

Ces différents modes de calcul des degrés d'imprécision seront implémentés dans notre moteur d'inférence. Nous présentons maintenant le mode de calcul des degrés d'incertitude.

L'incertitude que l'on rencontre dans le langage naturel se manifeste par des modificateurs linguistiques dont les tentatives de représentation furent souvent probabilistes. Les *modus ponens généralisés* que nous emploierons sont issus de la théorie des probabilités symboliques. Nous utiliserons le treillis suivant : $L_7 = \{\text{impossible, pas vraiment sûr, assez sûr, moyennement sûr, plutôt sûr, quasiment sûr, sûr}\}$.

Nous représentons le mode de calcul des degrés d'incertitude dans le tableau suivant :

Règles fortes conjonctives	Degré d'imprécision de déduction
$P(\alpha_1) A_1 \wedge \dots \wedge P(\alpha_n) A_n \rightarrow P(\beta) B$ $P(\alpha'_1) A_1, \dots, P(\alpha'_n) A_n$	$P(\lambda) B$ $\omega_1 = \max(t_0, t_{\alpha_2 + \omega_2 - M + 1})$ \dots $\omega_{n-2} = \max(t_0, t_{\alpha_{n-2} + \omega_{n-2} - M + 1})$ $\tau_{k_1} = \max(t_0, t_{\alpha_1 + \omega_1 - M + 1})$ $\tau_{k_2} = (((\tau_{\alpha'_1} \otimes \tau_{\alpha'_2}) \otimes \tau_{\alpha'_3}) \otimes \tau_{\alpha'_n})$ Si $\tau_{k_1} \geq \tau_{k_2}$ Alors $\tau_\lambda = \tau_\beta$ Sinon $t_\omega = \tau_{k_1} \nabla \tau_{k_2}$ $t_\lambda = \max(t_0, t_{\beta + \omega - M + 1})$

Ce calcul des degrés d'incertitude met en jeu deux notions, le voisinage symbolique, représenté par l'opérateur ∇ , ainsi que la multiplication symbolique représentée par l'opérateur \otimes .

Le degré d'incertitude de voisinage probabiliste est calculé suivant le tableau ci-dessous :

∇	τ_0	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6
τ_0	τ_0	τ_0	τ_0	τ_0	τ_0	τ_0	τ_0
τ_1	τ_0	τ_0	τ_0	τ_0	τ_0	τ_0	τ_0
τ_2	τ_0	τ_0	τ_1	τ_1	τ_1	τ_1	τ_1
τ_3	τ_0	τ_0	τ_1	τ_2	τ_2	τ_2	τ_2
τ_4	τ_0	τ_0	τ_1	τ_2	τ_3	τ_3	τ_3
τ_5	τ_0	τ_0	τ_1	τ_2	τ_3	τ_4	τ_4
τ_6	τ_0	τ_0	τ_1	τ_2	τ_3	τ_4	τ_5

Le calcul de la multiplication symbolique est donné par le tableau suivant :

\otimes	τ_0	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6
τ_0	τ_0	τ_0	τ_0	τ_0	τ_0	τ_0	τ_0
τ_1	τ_0	τ_1	τ_1	τ_1	τ_1	τ_1	τ_1
τ_2	τ_0	τ_1	τ_1	τ_1	τ_1	τ_1	τ_2
τ_3	τ_0	τ_1	τ_1	τ_2	τ_2	τ_2	τ_3
τ_4	τ_0	τ_1	τ_1	τ_2	τ_3	τ_3	τ_4
τ_5	τ_0	τ_1	τ_1	τ_2	τ_3	τ_4	τ_5
τ_6	τ_0	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6

Pour la gestion de l'incertitude, le *modus ponens généralisé* s'applique en trois étapes :

1. Calcul du degré d'incertitude sur l'ensemble des prémisses de la règle.
2. Calcul du degré d'incertitude des observations.
3. Comparaison des deux degrés et application du modus ponens.

Ainsi, à chaque proposition, nous associerons un degré d'incertitude et un degré d'imprécision, que cette proposition soit :

- un fait observé (constitutif de la base de faits),
- un fait en prémisses de règles (dans la base de règles),
- une conclusion de règle (dans la base de règles),
- ou un fait produit par application d'une règle (fait déduit de la base de connaissances où les degrés d'incertitude et d'imprécision sont calculés).

En outre, chaque règle stockée dans la base de règles sera associée à un degré de croyance. Dans notre moteur d'inférence, nous appelons cet indice *degré de priorité* en raison de l'ordre pris par les règles dans la base suivant ce paramètre.

SelfMind : un outil de gestion des connaissances imparfaites

Nous avons appelé notre logiciel SelfMind, abréviation de *Système Expert en Logiques Floue et Multivaluée pour l'Imitation Naturelle des Décisions*. Ce logiciel, en libre diffusion sur Internet, a été développé au sein du laboratoire ERIC de Lyon. Nous l'avons programmé en *Borland Delphi* pour Windows. SelfMind fonctionne sur des ordinateurs PC, dans les environnements de *Microsoft Windows 3.1* et *95*, il emploie des objets visuels classiques de ces interfaces graphiques et offre une aide aussi bien par index que contextuelle.

Nous allons maintenant présenter notre moteur d'inférence, voir les fonctionnalités qui sont proposées et en quoi ses particularités font de notre outil un modèle intéressant pour l'étude des processus cognitifs de la perception.

À l'exécution du logiciel SelfMind, la page d'accueil suivante apparaît :



Cette page fait apparaître les noms de l'auteur, de son collaborateur et du directeur, ainsi que l'endroit où ce logiciel a été élaboré. SelfMind est associé à l'image d'une tête présentée sous la forme d'une coupe sagittale obtenue par imagerie médicale, avec des engrenages symbolisant les mécanismes de la pensée. Cette image est aussi l'icône du logiciel.

Dans SelfMind, nous avons choisi de représenter trois ensembles de données :

- une **base de règles** composant l'expertise du système ;
- une **base de faits** correspondant à l'état d'un ou de plusieurs individus observés (ensemble de faits particuliers) ;
- un **dictionnaire des faits** regroupant l'ensemble des briques constitutives des propositions exploitées par la base de règles et par la base de faits.

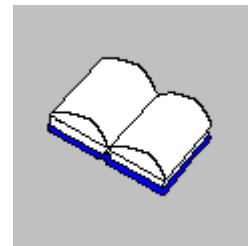
Pour plus de clarté dans l'utilisation du logiciel, nous avons associé à chacun de ces ensembles de données une image symbolique :



base de règles



base de faits



dictionnaire des faits

Pour se servir de SelfMind, il faut disposer d'un ensemble de faits. Au démarrage, l'utilisateur peut donc ouvrir un projet déjà créé ou concevoir un nouveau projet. La conception d'un nouveau projet peut se réaliser de deux manières :

- l'utilisateur peut importer un fichier de règles issues d'une autre application, comme un logiciel d'extraction automatique de connaissances à partir de données, et SelfMind constituera son dictionnaire de faits en retrouvant les propriétés présentes dans le fichier de règles issu de l'apprentissage symbolique ;
- l'utilisateur peut créer son propre dictionnaire de faits en choisissant le nom des attributs et le format des *antécédents* comme nous le présentons ci-dessous.

Phase de création du dictionnaire par ajout de nouveaux faits :

Close

Attribute:
[Enter a new attribute or select an attribute in the list]

ATTRIBUTE_1
COLOR
PROPOSITION_1
PROPOSITION_2
PROPOSITION_3

Operator: Select ->

Value(s):

=
>
<
>=
<=
<>
in
not in

Remove data

Add antecedent ->

Data Dictionary: 8 data

ATTRIBUTE_1 > 1000
ATTRIBUTE_1 < 5000
ATTRIBUTE_1 > 1235000
ATTRIBUTE_1 < 503,3
COLOR = BLUE
PROPOSITION_1 = TRUE
PROPOSITION_2 = TRUE
PROPOSITION_3 = TRUE

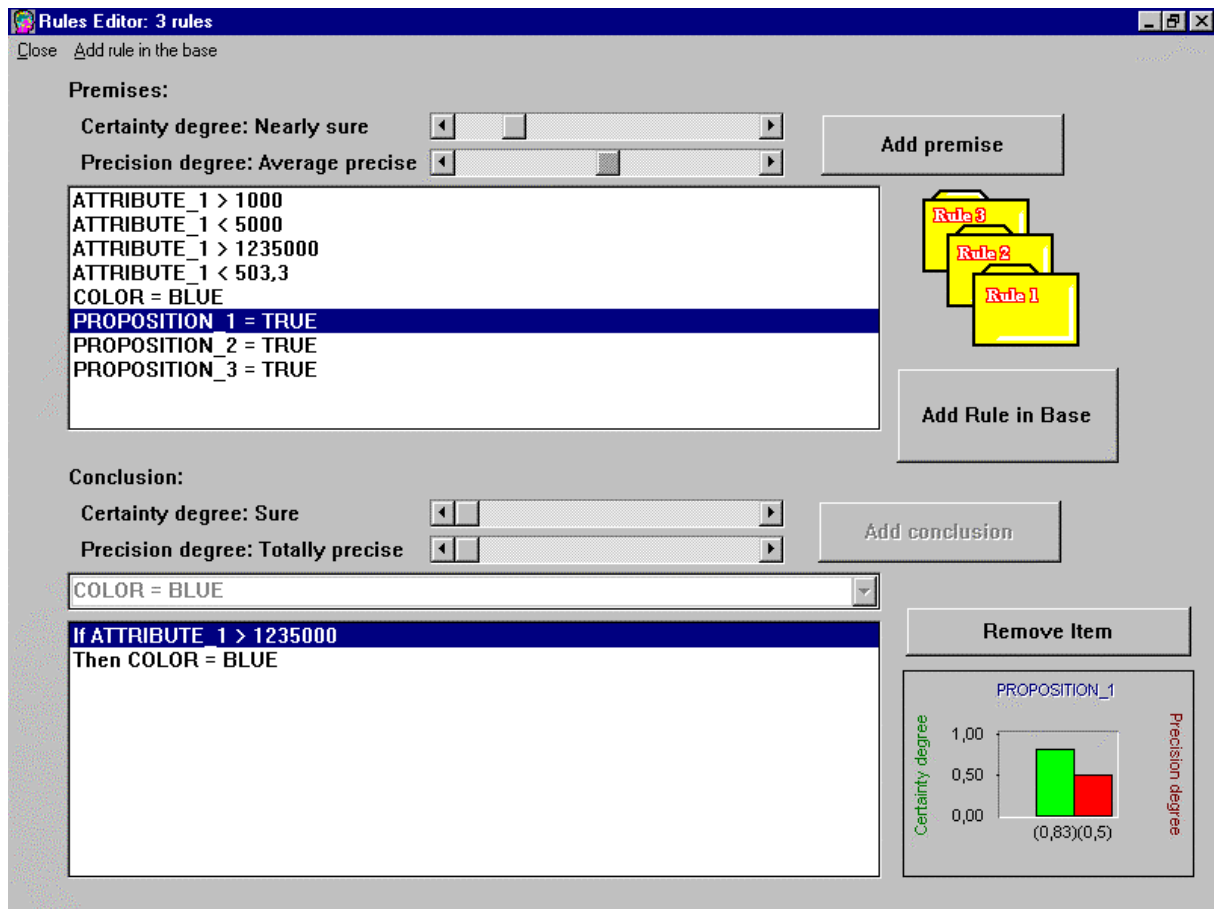
Antecedent:
[Attribute] [Operator] [Value(s)]

Le dictionnaire est composé d'un ensemble de faits, appelés *antécédents*.

Un antécédent est composé :

- d'un attribut (nom de la variable étudiée)
- d'un opérateur :
 - ◇ = suivi par une valeur numérique, une valeur booléenne (« vrai » ou « faux ») ou par une propriété
 - ◇ <>, >, <, >=, <= suivi par une valeur numérique
 - ◇ **In**, **Not_in** suivi d'un ensemble de propriétés
- d'une valeur ou d'un ensemble de valeurs qui peuvent être :
 - ◇ des valeurs entières ou réelles, positives ou négatives
 - ◇ des valeurs booléennes (**True** ou **False**)
 - ◇ des propriétés (un nom écrit sous forme de chaîne de caractères)

À partir de ces faits, il est possible de constituer la base de règles :



Une règle est de la forme :

If *Prémisse 1* **And** *Prémisse 2* **And** ... **And** *Prémisse n*
Then *Conclusion* [Degré de priorité]

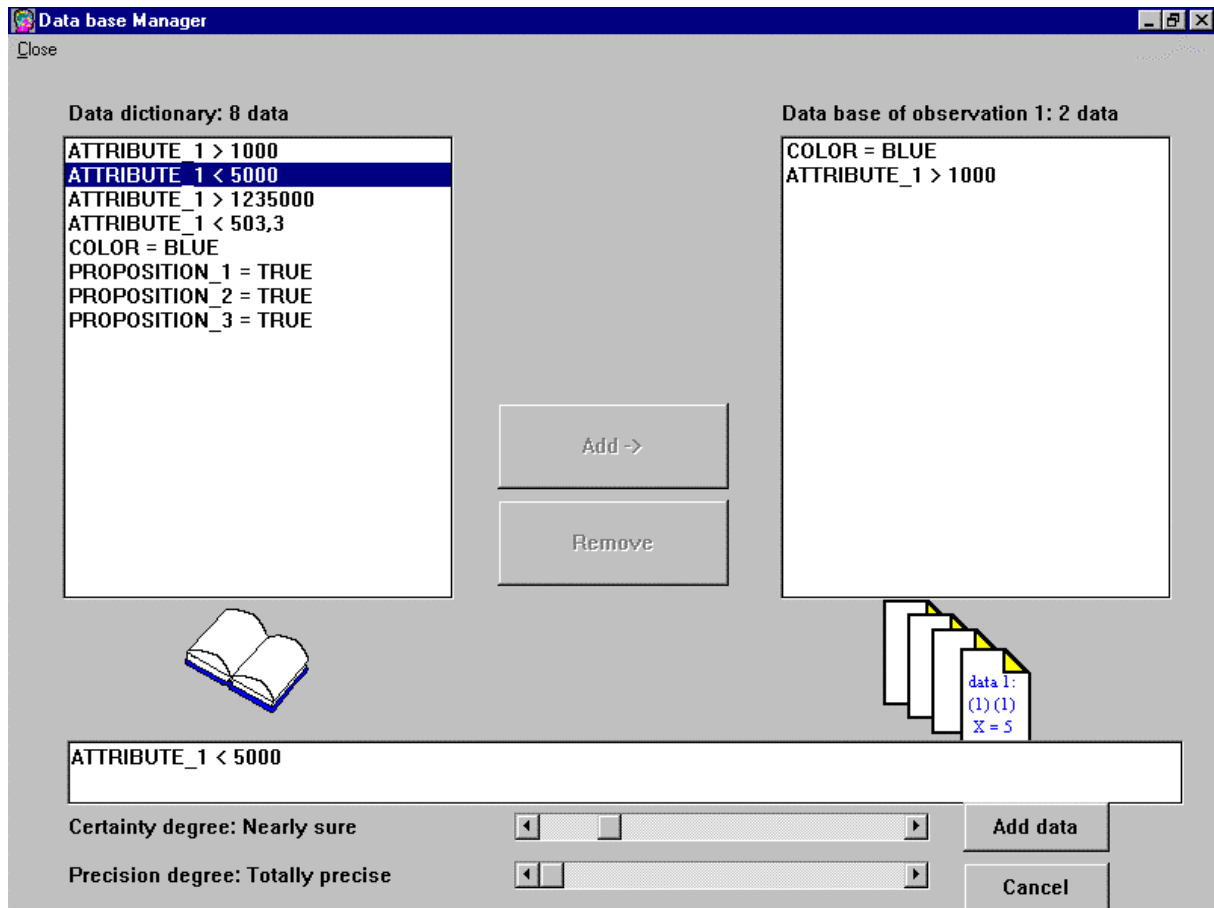
Les prémisses et la conclusion de la règle sont des *propositions*. Selon la terminologie que nous avons adoptée, une proposition est un antécédent issu du dictionnaire de faits auquel on a ajouté un degré d'incertitude et un degré d'imprécision.

Les règles se trouvent sous forme conjonctive (prémisses liées par l'opérateur logique **Et**). Le degré de priorité est une valeur comprise entre 0 et 1 qui indique le degré de croyance que l'on attribue à la règle. Nous préciserons le sens de ce degré de priorité lorsque nous aborderons la gestion des paramètres de l'inférence.

La base de règles constitue la connaissance générale sur un domaine d'étude. La base de faits, quant à elle, regroupe les faits observés en pratique sur un ou plusieurs individus.

Dans la problématique de la perception, ces faits observés consistent en un ensemble de propriétés physiques issus des stimuli sensoriels. Ces données peuvent être recueillies par des outils d'acquisition jouant le rôle de récepteurs sensoriels (comme une caméra, un microphone, ou des capteurs chimiques) et apportées sous forme de fichier de propositions.

Il est toutefois possible de décrire les observations avec le seul logiciel SelfMind. Dans ce cas, les propositions décrivant les observations sont obtenues par ajout de degrés d'incertitude et d'imprécision à certains antécédents issus du dictionnaire de faits. L'écran suivant présente une telle étape :



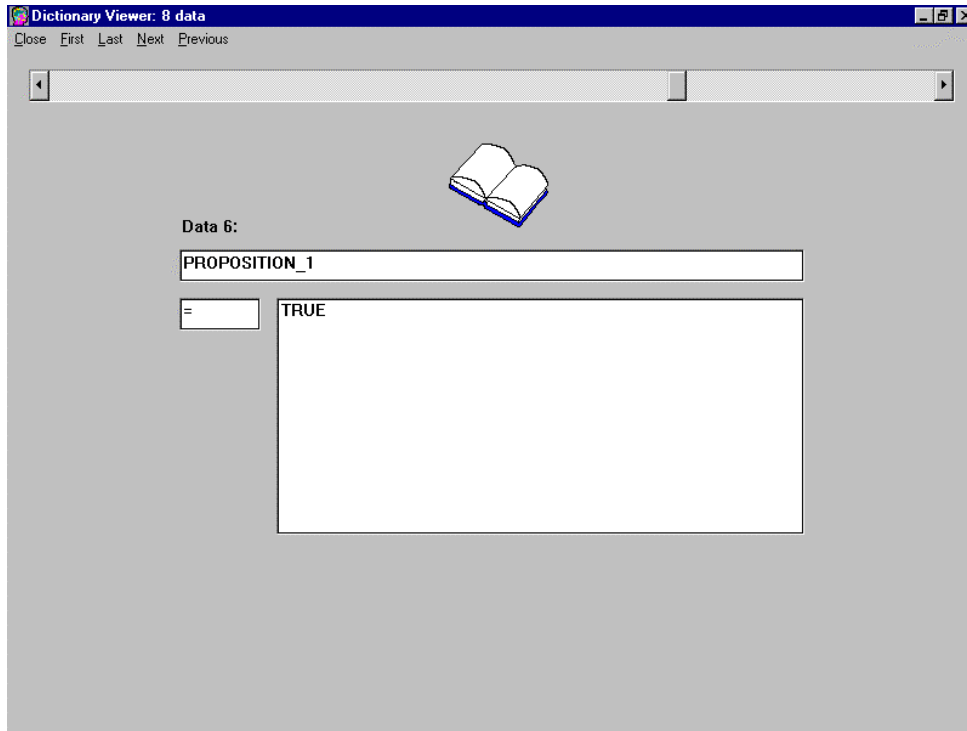
Nous disposons à présent d'un dictionnaire des faits, d'une base de règles et d'une base de faits. Des options nous permettent de visualiser chacun de ces éléments.

Chaque ensemble ne présente qu'une donnée à la fois, mais des barres de défilement horizontales et des options de menus permettent une visualisation aisée des informations.

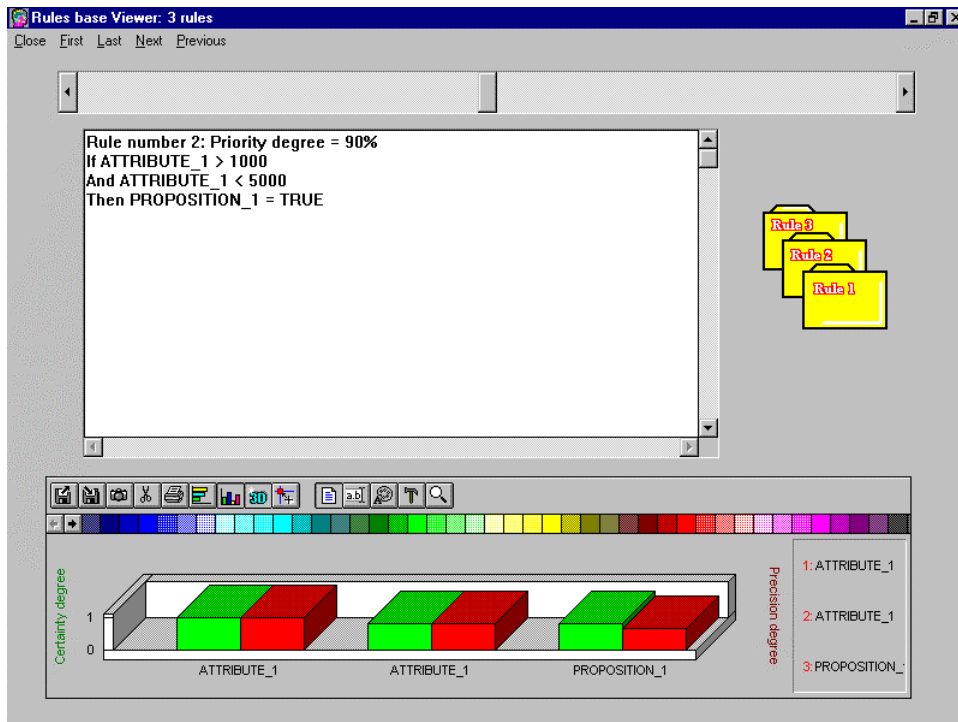
Les données se trouvent ordonnées dans chacun des ensembles selon :

- le degré de priorité pour les règles ;
- l'ordre alphanumérique des attributs d'antécédents pour les faits du dictionnaire ;
- l'ordre alphanumérique des attributs de propositions pour les faits de la base.

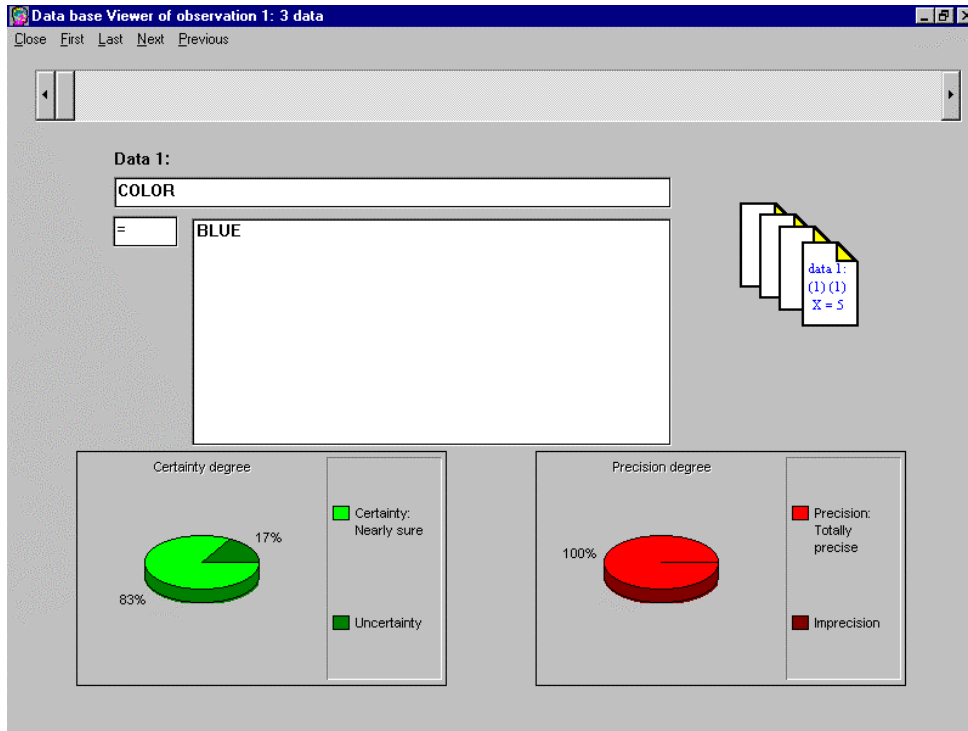
Dictionnaire des faits :



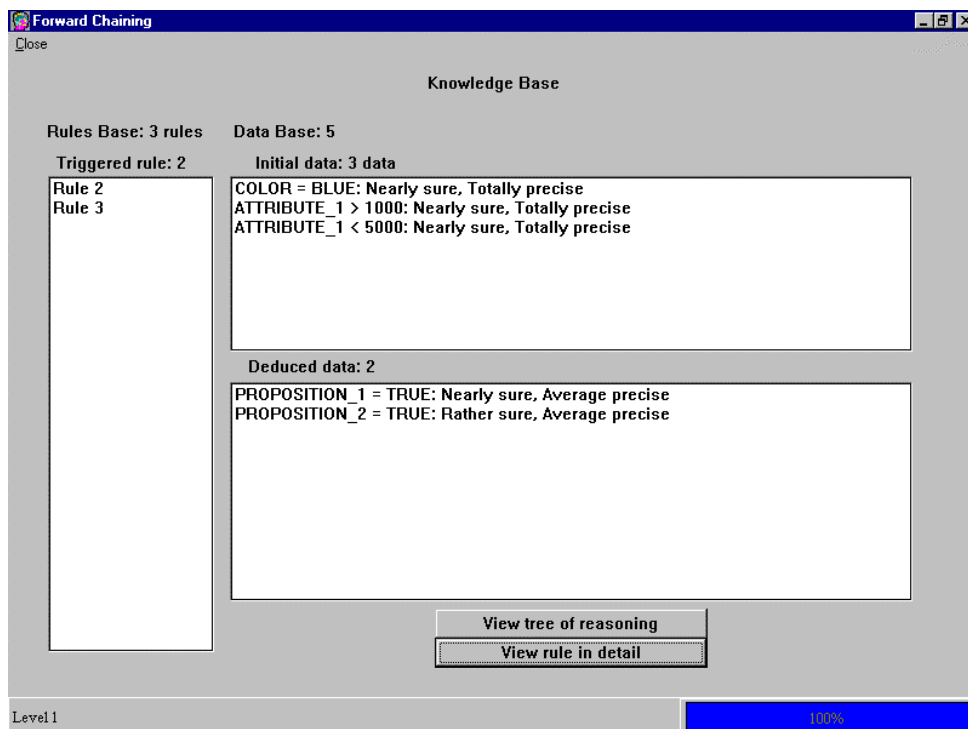
Base de règles :



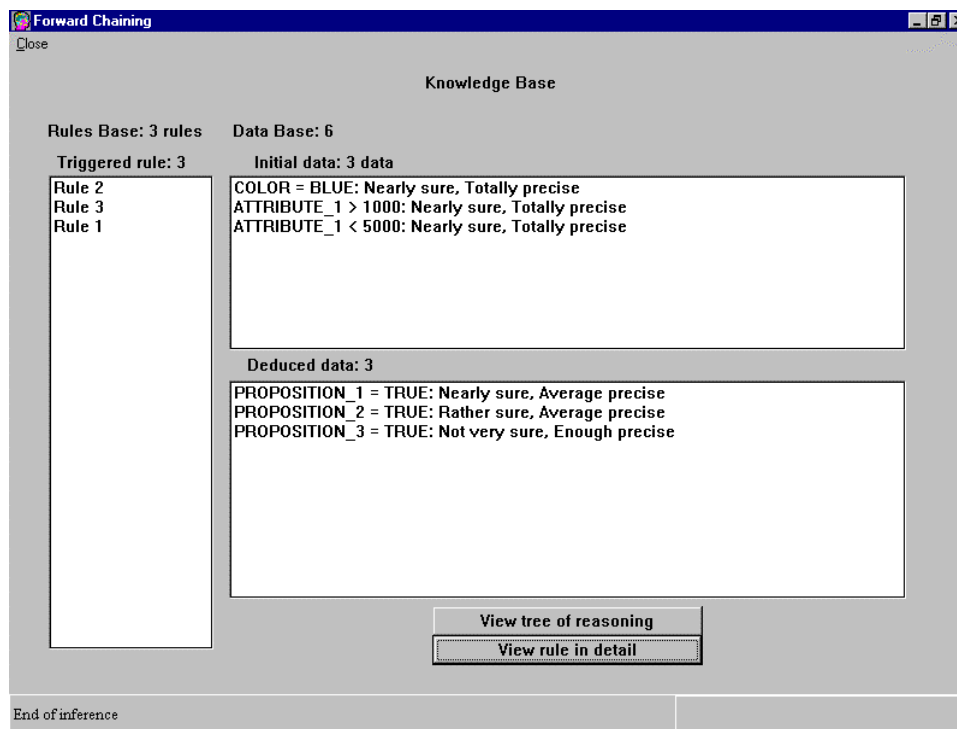
Base de faits :



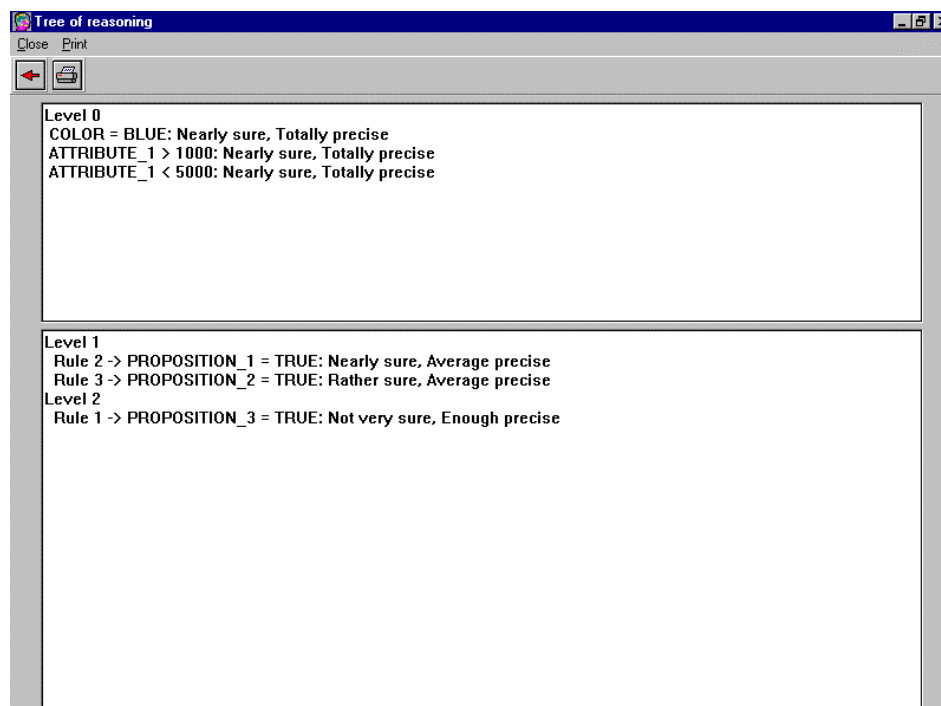
Lorsque l'on dispose d'une base de connaissances — c'est-à-dire une base de faits observés sur un individu particulier avec une base de règles représentant la connaissance générale sur le domaine étudié —, nous pouvons enfin procéder à l'inférence.



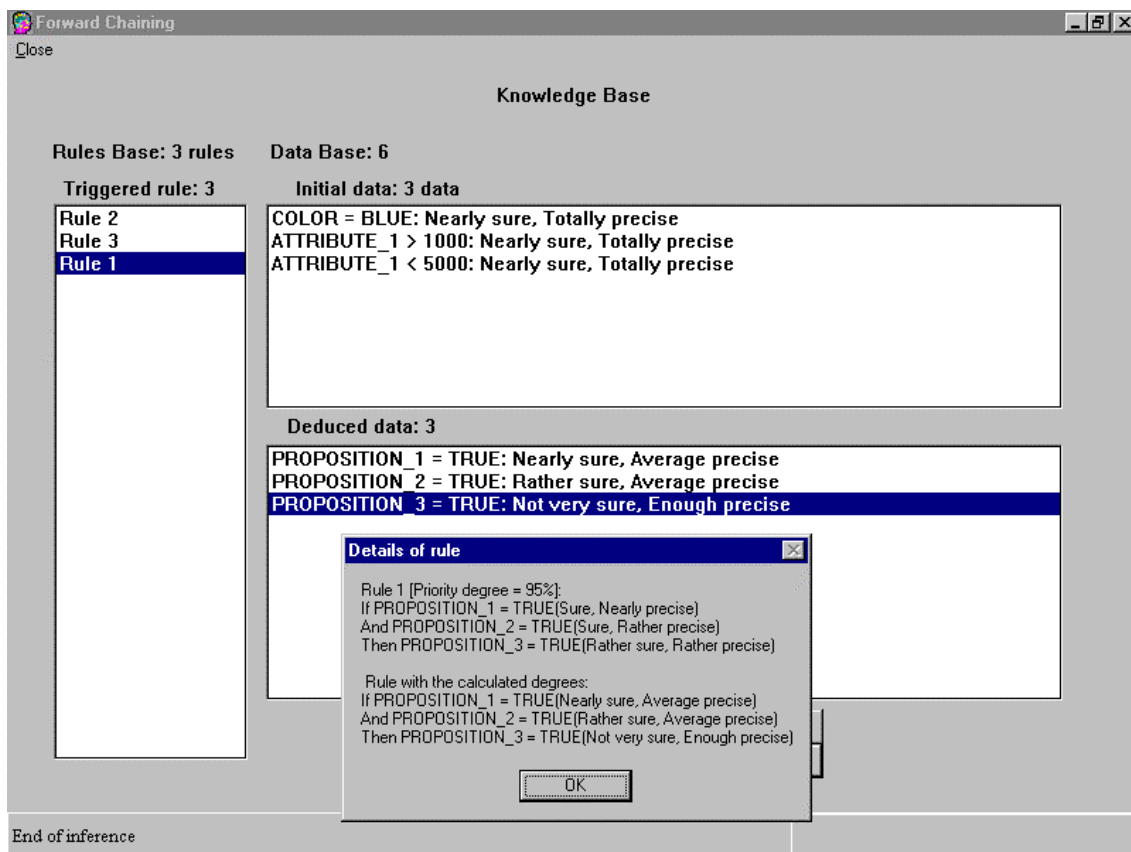
SelfMind effectue du chaînage avant (*forward chaining*) : il essaie de générer de nouveaux faits en déclenchant des règles à partir des données observées. Ces faits déduits serviront à leur tour pour déclencher d'autres règles, et ainsi jusqu'à ce que toutes règles de la base aient été déclenchées ou que plus aucun fait n'ait été déduit après un nouveau parcours.



Les faits sont par conséquent déduits suivants des niveaux correspondant au passage en revue de la base de règles. Il est d'ailleurs possible de voir cet arbre de raisonnement :



D'autre part, il est possible de voir les détails de la règle qui a permis d'aboutir au déclenchement de tel ou tel fait. En sélectionnant un fait déduit ou une règle utilisée, on peut faire apparaître une boîte de messages dans laquelle sont décrites la règle telle qu'elle se trouve dans la base et la règle telle qu'elle a été utilisée lors de l'inférence.



Avec le logiciel SelfMind, nous proposons aussi la possibilité d'effectuer du chaînage arrière (*backward chaining*). Ce mode d'inférence permet, à partir de la sélection d'une conclusion, de retrouver les faits qu'il est nécessaire d'avoir en données observées pour arriver à conclure ce fait. Pour cela, SelfMind se charge de retrouver les règles qui permettent d'aboutir à cette conclusion et demande à l'utilisateur si les prémisses de ces règles sont validées ou non, amenant éventuellement à une recherche en profondeur des règles permettant de déduire ces faits de prémisses, puis les prémisses des règles aboutissant à ces prémisses, et ainsi de suite. Ce mode interactif est surtout utile lorsque l'on cherche à optimiser la base de connaissances en étudiant les liens s'effectuant entre les faits par l'ensemble des règles.

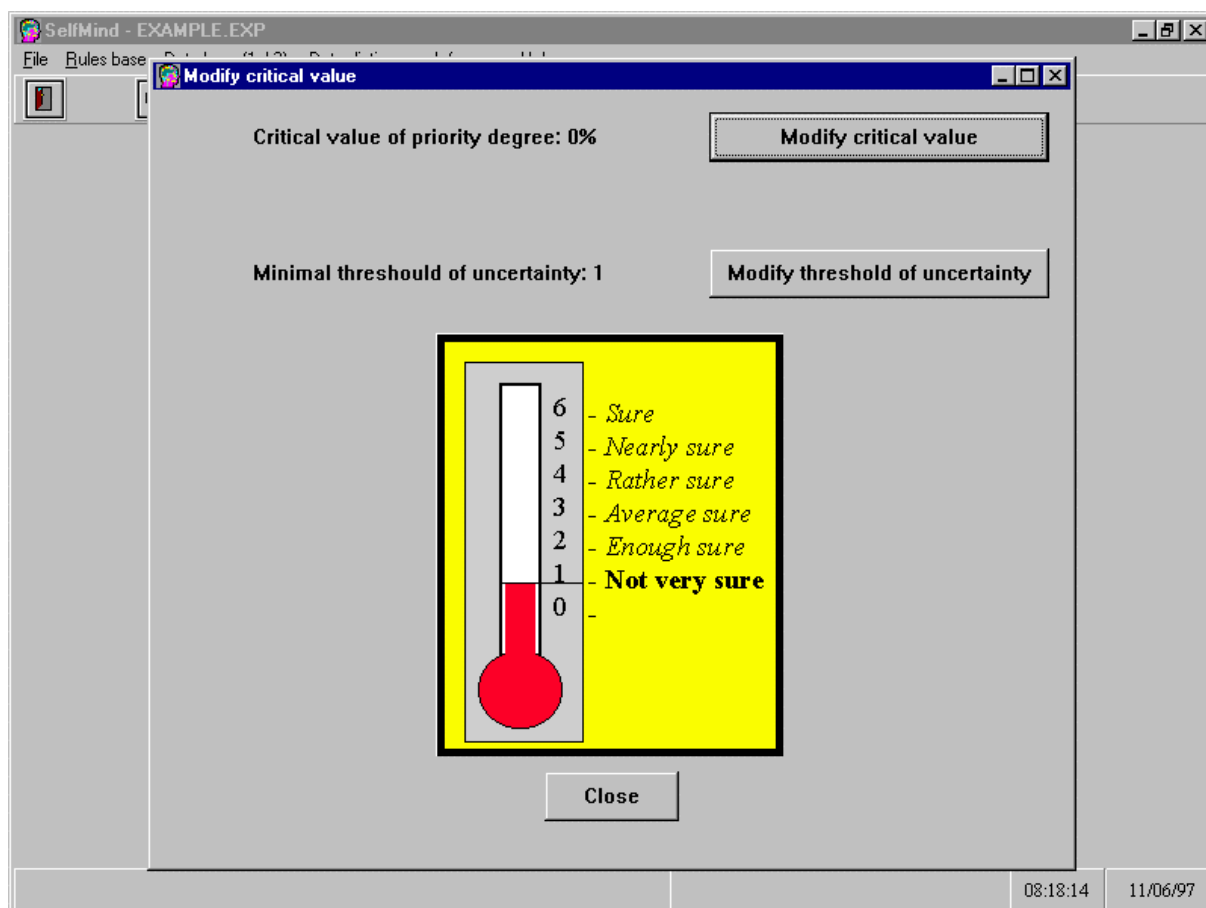
Mais il faut rappeler que notre apport principal par rapport aux systèmes experts classiques est l'emploi de degrés d'incertitude et d'imprécision dont le mode de calcul, inspiré des travaux de Khoukhi (1996), a été présenté en début de cette partie.

En raison de l'imperfection des données et de l'imperfection des règles, les informations activées produiront des inférences de degrés d'incertitude et d'imprécision moins forts que les degrés des propositions initialement observées. Ces informations inférées engendreront à leur tour d'autres informations, et ainsi de suite, jusqu'à ce que le degré d'incertitude des informations calculées ne soit plus suffisant pour qu'elles soient déduites.

Il est en effet possible de modifier des paramètres qui limiteront la recherche des faits à déduire et le nombre de faits déduits selon :

- le degré de croyance minimal d'une règle pour qu'elle soit déclenchée (degré de priorité de la règle),
- le degré de certitude calculé le plus bas pour qu'un fait soit déduit.

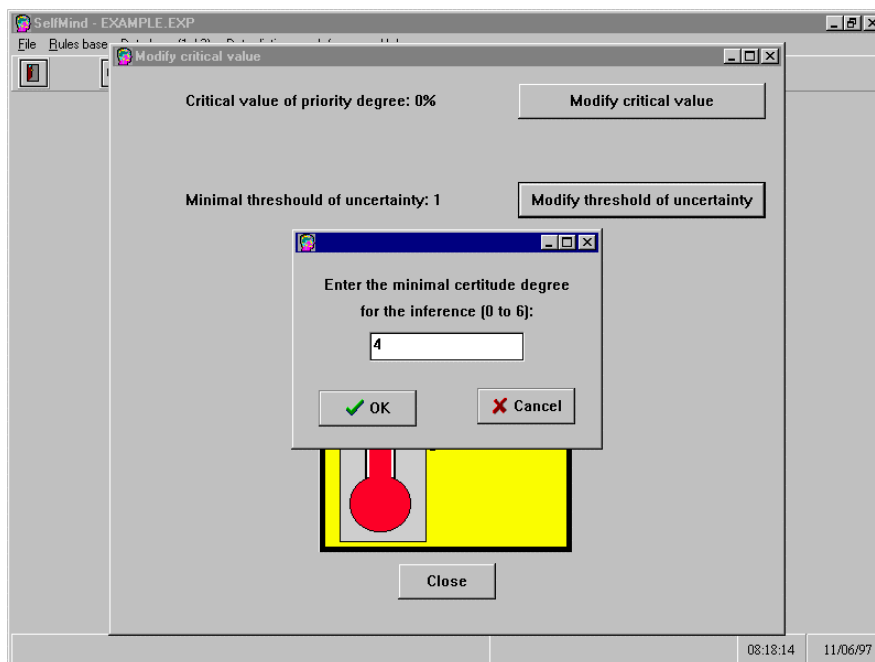
Ces paramètres sont gérés sur l'écran de SelfMind suivant :



Par défaut, le seuil minimal du degré d'incertitude calculé pour qu'un fait soit déclenché est de 1, correspondant à « Pas vraiment sûr ». Il n'y aura ainsi pas de faits déduits avec un degré de 0, correspondant à « Impossible ».

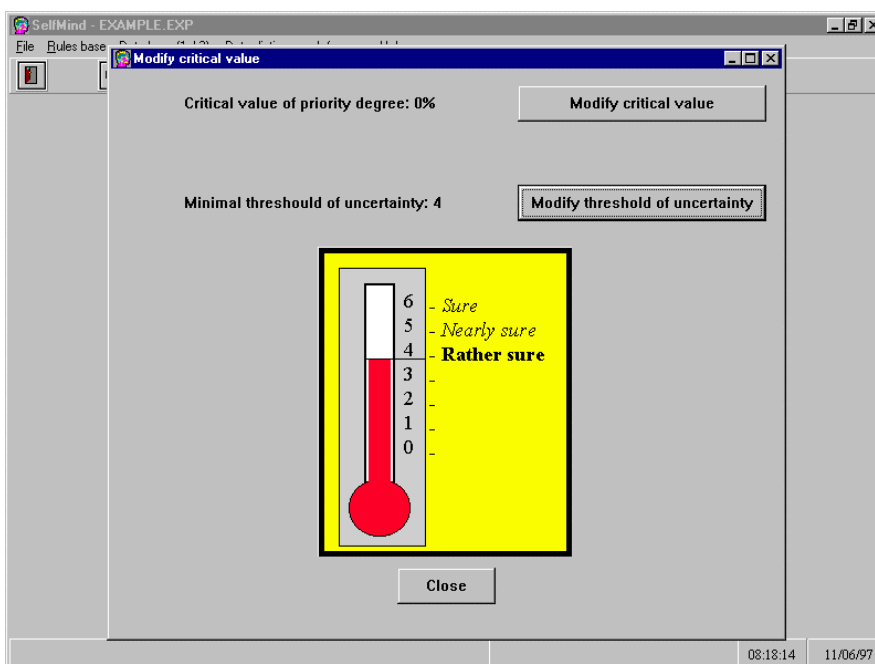
Nous souhaitons faire remarquer que dans notre système, une proposition « $X = \text{Vrai}$ » avec un degré « Impossible » est différente de « $X = \text{Faux}$ » avec un degré « Sûr ».

La valeur du seuil minimal d'incertitude peut être modifiée en entrant une valeur différente allant de 0 (« Impossible ») à 1 (« Sûr »).

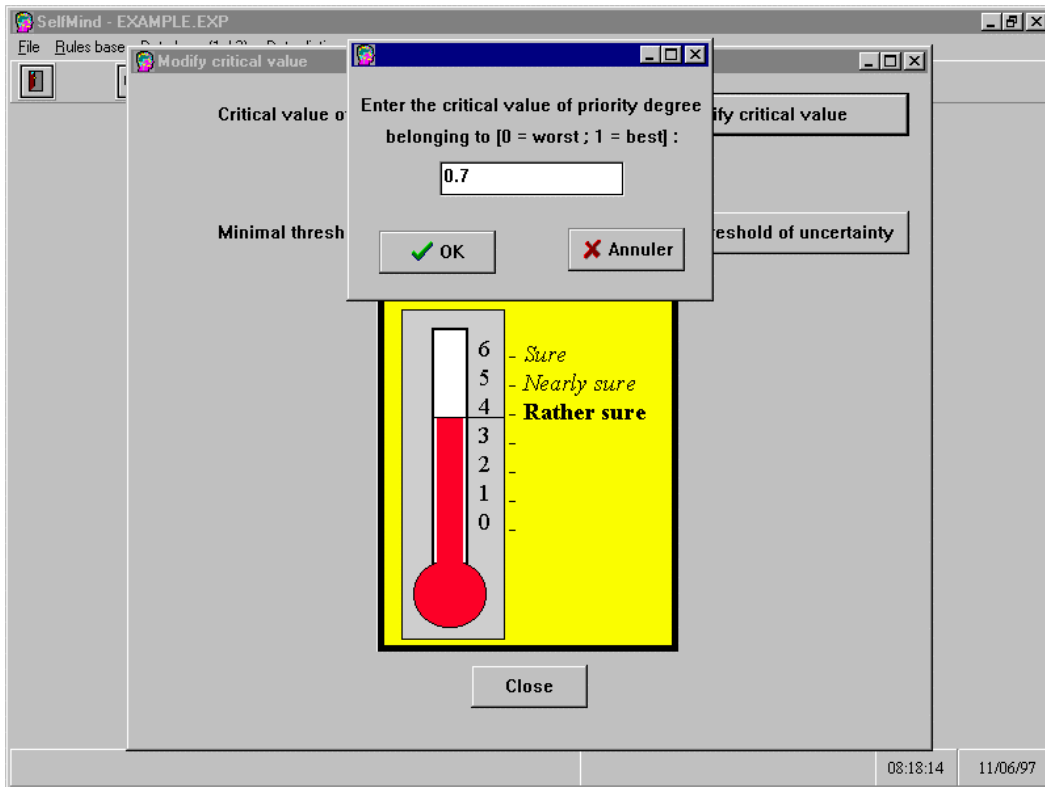


Après avoir entré une valeur de 4, correspondant à « assez sûr », l'écran affiche un thermomètre qui fait monter le seuil minimal d'incertitude.

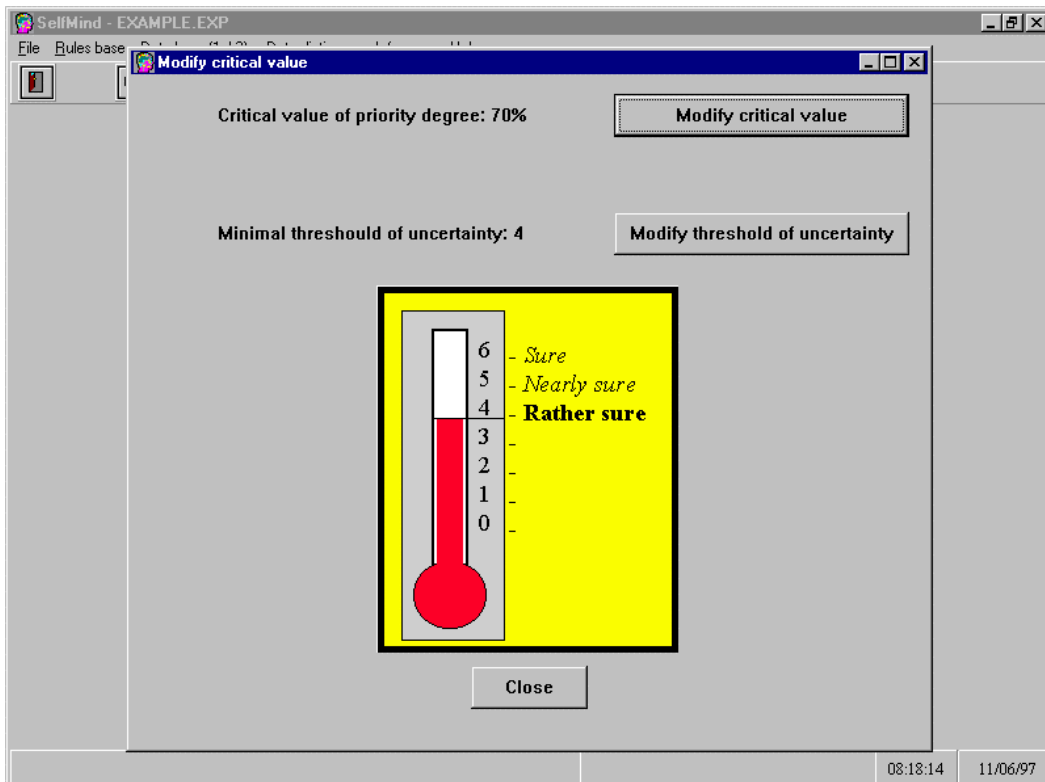
De la sorte, seuls les faits ayant un degré d'incertitude calculé « assez sûr », « quasiment sûr » ou « (totalement) sûr » seront déduits lors du chaînage avant.



Il est aussi possible de modifier la valeur critique du degré de priorité :



De cette façon, nous ne tiendrons compte que des règles ayant un degré de priorité élevé, ce degré correspondant à la croyance que nous avons en la règle.



L'intérêt de pouvoir limiter la recherche de ces deux manières apparaît dans notre façon de modéliser les processus perceptifs humains. En effet, avec une base de règles exhaustive, il serait théoriquement possible de lier toute information X à une information Y en empruntant un chemin composé d'un certain nombre de règles. Toute proposition de notre base de règles pourrait finir par être reliée à toutes les autres propositions, à la manière d'un réseau sémantique totalement interconnecté. Mais, avec la gestion de degrés d'incertitude dans notre système, nous évitons de déduire des informations trop lointaines.

De plus, suivant le contexte, on pourra ne retenir que les règles aux degrés de priorité supérieurs à une valeur minimale pour effectuer nos inférences. Ce procédé permet d'être soit très sélectif dans l'usage des règles employées soit d'exploiter plus intensément la base de règles.

Ces deux critères d'arrêt sont importants dans une modélisation des processus cognitifs. En effet, un mot — dans le sens du langage mental fodorien — peut amener à la signification d'un autre mot, et cela jusqu'à une certaine distance car, grâce à ces paramètres, notre système permet d'éviter de produire des « épilepsies » sémantiques.

Ajoutons encore après cette présentation des bases théoriques de notre moteur d'inférence et de son utilisation générale que l'acronyme SelfMind peut se traduire par les termes « auto-esprit » ou « esprit de soi ». Ce nom reflète bien l'idée que nous avons du fonctionnement du logiciel, car nous souhaitons qu'à partir d'informations symboliques qui lui sont données, SelfMind génère de lui-même des informations associées à des propriétés perceptives grâce à ses connaissances, tout comme le ferait de manière naturelle un être humain.

Seconde partie :

**Applications du moteur d'inférence
SelfMind dans le domaine de la perception**

Application 1 :

Génération de propriétés olfactives associées à une odeur

L'olfaction est une modalité sensorielle privilégiée pour l'étude des traitements sémantiques impliqués dans la perception. En effet, les odeurs sont associées au goût — une odeur peut être sucrée, salée, aigre, amère ou acide — et aux sensations tactiles — une odeur peut, par exemple, être piquante, chaude ou fraîche —. C'est cet ensemble d'informations sensorielles multimodales qui amène à la perception d'une odeur.

Classiquement, on distingue différents niveaux de profondeur de traitements au sein de la perception. À partir des informations sensorielles de base, il est possible d'effectuer la *reconnaissance* du stimulus — l'odeur dans notre cas —. À cette étape, il est possible d'indiquer si l'odeur présentée est familière ou non, si elle produit une sensation de déjà vu (ou de *déjà senti*). Des traitements plus élaborés amènent à l'étape d'*identification*. À ce niveau, il devient possible de donner du sens à une odeur et d'effectuer des associations sémantiques, c'est-à-dire notamment de citer des propriétés olfactives associées à l'odeur. La dernière étape de traitement perceptif consiste en la *dénomination* de l'odeur, c'est-à-dire l'accès au lexique mental pour pouvoir donner un nom à l'odeur.

Cependant cette dernière étape est très peu performante dans le domaine olfactif chez des personnes de culture occidentale. Dans nos sociétés, il n'existe pas de termes de base d'odeurs dans la langue, vraisemblablement associé au fait que l'olfaction naturelle n'y joue pas — ou plus — de rôle significatif. Nous essayons en effet de masquer nos odeurs corporelles par des déodorants et des parfums, nous ajoutons des arômes à nos aliments et, globalement, nous cherchons à ignorer la sphère olfactive de notre environnement. Un nom d'odeur devient par conséquent délicat à donner. On peut d'ailleurs se demander quel serait ce nom d'odeur : faudrait-il considérer le nom chimique de la molécule odorante ou celui de l'objet naturel d'où émanent de nombreuses molécules chimiques ?

L'objectif principal est donc bien de pouvoir générer des propriétés olfactives associées à une odeur, de procéder par associations d'idées, ce qui se prête bien à une modélisation par un moteur d'inférence. Effectivement, même si ces associations d'idées sur les odeurs ne sont pas très « logiques », les règles du moteur d'inférence peuvent s'adapter en modèle de proximités sémantiques, avec dans SelfMind l'avantage de tenir compte de la distance entre ces propriétés au moyen de degrés d'incertitude.

La connaissance sur ces associations est basée sur une expérience décrite par Sebban et Zighed (1996) qui a été réalisée sur près de 150 sujets humains pour étudier l'effet de la mémorisation selon les niveaux de traitements opérés sur les odeurs.

La liste des propriétés sémantiques utilisées pour décrire les odeurs est la suivante :

Acide	Herbacée
Amère	Médicinale
Boisée	Salée
Corporelle	Sucrée
Epicée	Entêtante
Florale	Fraîche
Fruitée	Graisseuse
Fumée	Irritante

Ces descripteurs étaient utilisés pour caractériser une trentaine d'odeurs (allant du lilas à l'éther, en passant par la menthe, la lavande, le saumon fumé et une odeur de transpiration). Afin d'avoir une base de connaissances plus fournie que ces 16 propriétés sémantiques, Sebban (1996) a utilisé les données des profils sémantiques d'odeurs de Dravnieks, ce qui a permis d'ajouter 26 propriétés supplémentaires. Ces propriétés sont les suivantes :

Fruité (agrumes)	Alcoolisé
Fruits pourris	Viande (odeur de chair)
Parfumé	Anesthésique
Terreuse	Rance
Odorant (selon la fragrance)	Produits de nettoyage
Chimique	Douce
Aromatisée	Dissolvant
Grasse	Métallique
Assaisonnement (de plat)	Légumes verts frais
Chaude	Putride
Lourde	Piquante (pointue)
Légère	Mentholée
Aigre (vinaigre)	Ecoeurante

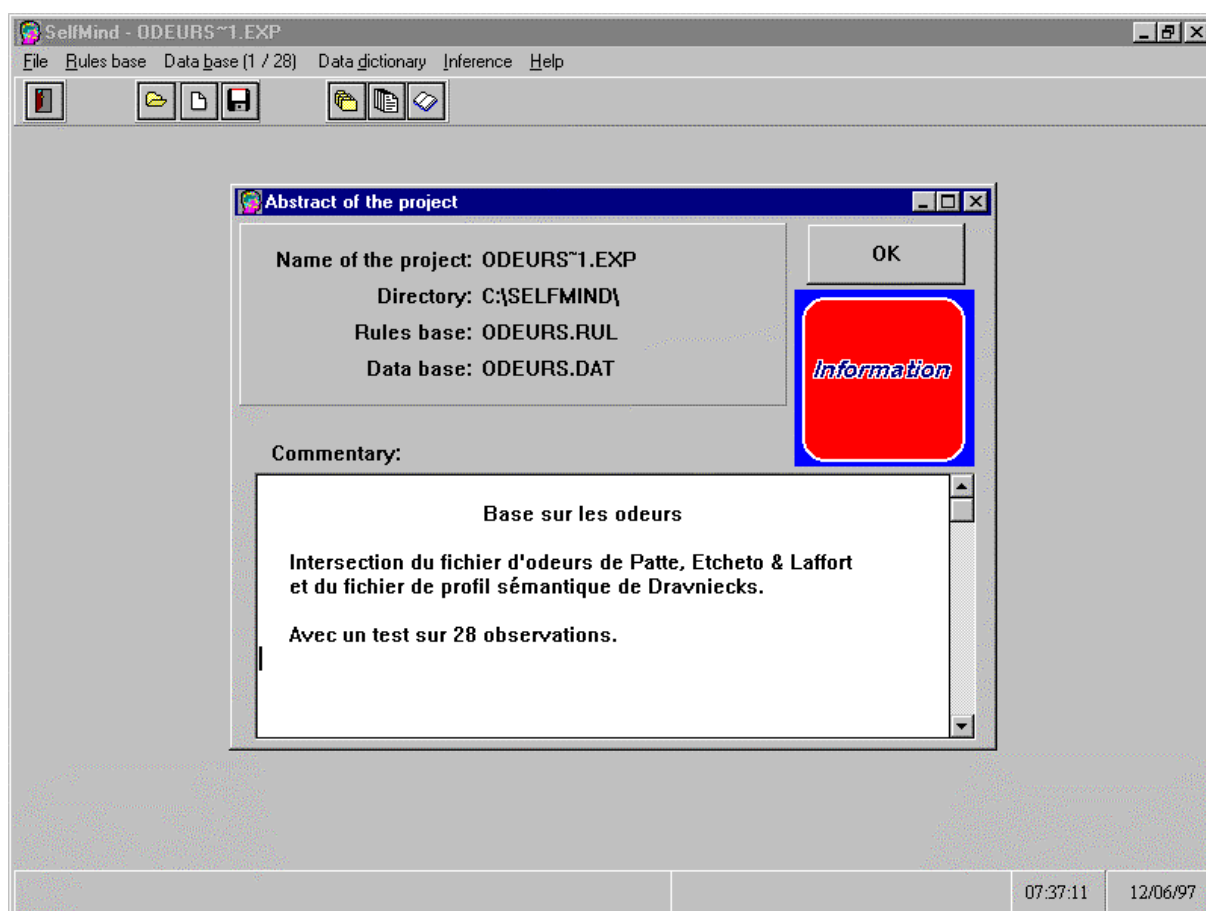
Sur les 146 descripteurs d'origine de Dravnieks, ces 26 propriétés ont la particularité d'être communes avec celles étudiées par l'équipe de Patte (Sebban, 1996). Pour l'élaboration d'un modèle cognitif pour l'identification des odeurs, Zighed et Sebban (1996) ont en effet besoin d'odeurs dont les paramètres physico-chimiques sont connus. Ces 5 paramètres physico-chimiques — la dispersion, la basicité, l'orientation, l'acidité et la polarisabilité des

molécules odorantes — sont à la base d'un traitement de bas niveau qui permet d'identifier une molécule à laquelle sont associées des propriétés sémantiques définies par des experts comme Arctander (cité par Sebban, 1996).

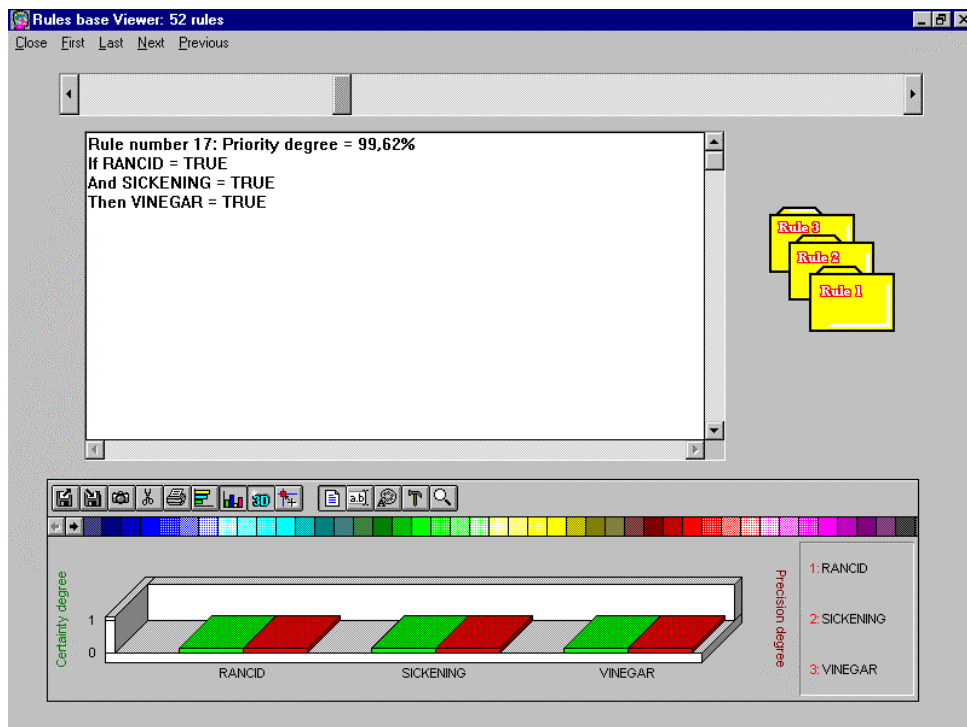
La base de connaissances a consisté à reprendre les associations réalisées par les sujets dans l'expérience de Dravnieks (cité par Sebban, 1996) en formant des règles de production utilisables par SelfMind. Dans cette expérience, les sujets avaient à décrire leurs sensations par des propriétés jugées pertinentes mais aussi en codant la précision de ces sensations par une valeur d'adéquation entre le descripteur et l'odeur compris entre 0 et 5. Ces valeurs peuvent aisément être transformées par moyennage pour représenter le *degré de précision* exploité par notre moteur d'inférence. En outre, en considérant le pourcentage de sujets ayant choisi d'associer un descripteur à une odeur, il est possible d'extraire le degré de certitude de cette règle d'association. Il a ainsi été possible de concevoir des règles d'associations de propriétés odorantes (dont un échantillon est présenté en annexe 1) au moyen de plusieurs techniques d'extraction de connaissances à partir de données dont nous parlerons dans la troisième partie.

Nous présentons maintenant l'utilisation de cette base de connaissance dans SelfMind.

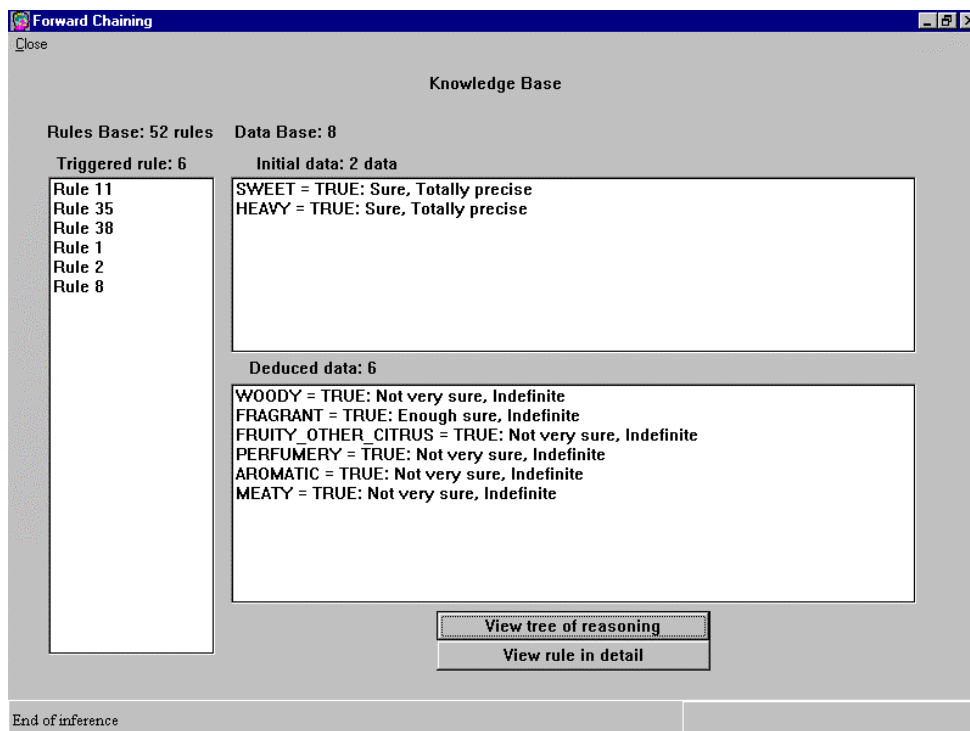
Le projet sur les odeurs s'ouvre avec les informations suivantes :



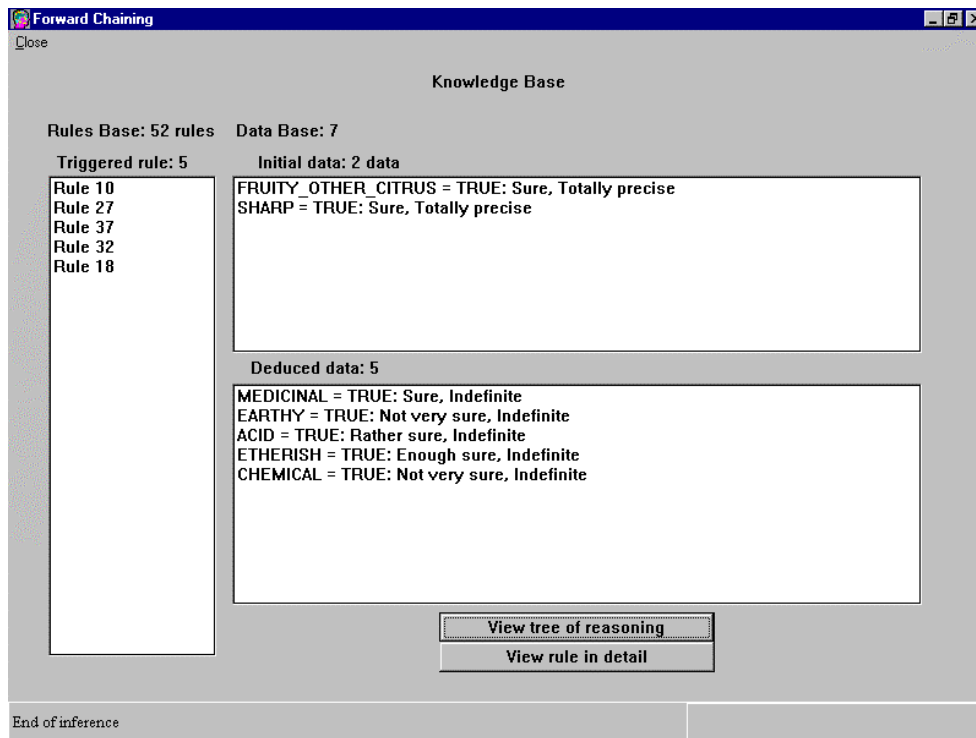
Comme le montre l'écran suivant, les règles présentes dans la base ont des faibles degrés de précision et de certitude :



Les propriétés inférées par chaînage avant auront par conséquent des degrés de certitude et de précision faibles :



Autre exemple de génération des propriétés olfactives associées à une odeur :



Nous découvrons qu'à partir d'odeurs sucrées et lourdes, des associations sont possibles vers des odeurs boisées (comme une forêt de sapins), des odeurs de parfums, et des odeurs de nourriture (fruits et viande).

Avec des odeurs de fruits (autres que le citron) et piquantes, nous aboutissons à des odeurs du monde médical (odeurs médicinales, acides, chimiques, odeurs d'éther), ainsi qu'à des odeurs de terre.

Les degrés de certitude et de précision nous renseignent cependant sur le poids de telles associations. L'olfaction est une modalité sensorielle où les informations demeurent assez peu précises, faute d'un bon mode de description. Cet aspect est bien rendu par une modélisation avec le logiciel SelfMind, car les associations apportent de l'information suivant différentes voies d'inférence. D'autre part, il est possible de visualiser ces chemins par affichage de l'arbre de raisonnement dont la profondeur des niveaux rend compte de l'entendue des traitements opérés au sein de la mémoire associative.

Application 2 :

Reconnaissance de chiffres manuscrits

Forts des résultats encourageants obtenus dans le domaine de l'olfaction, nous avons cherché à utiliser notre moteur d'inférence dans un système expert en reconnaissance de formes ayant trait à la modalité visuelle. Par essence, l'olfaction met en jeu la synesthésie (des informations gustatives et tactiles sont mêlées aux informations olfactives). Les sensations visuelles, quant à elles, ne font intervenir que des informations issues des photorécepteurs de la rétine. Elles n'impliquent donc pas l'association spontanée de sensations appartenant à des domaines différents. Nous avons cherché à voir si de telles associations étaient toutefois possibles en nous intéressant aux processus de reconnaissance des chiffres manuscrits.

Pour cela, nous avons effectué une expérimentation qui nous a permis de recueillir des informations sur les impressions à la fois perceptives et sémantiques associées à la présentation visuelle de chiffres manuscrits.

METHODE

Matériel

L'expérience réalisée se déroulait en deux phases :

- décisions sur les associations entre chiffres et propriétés perceptives ;
- décisions sur les associations entre chiffres et propriétés sémantiques.

Pour la passation de l'expérience, nous avons développé un logiciel qui a permis l'affichage de nos stimuli et l'enregistrement de la nature des réponses ainsi que des temps de réaction mis pour la décision.

Les stimuli consistaient en 10 chiffres écrits sous forme d'une police manuscrite, présentés pendant un temps suffisant pour être lus (500 ms). Puis apparaissait une propriété que nous avons qualifiée de « perceptive » ou de « sémantique » suivant la phase expérimentale dans laquelle se trouvaient les sujets.

Epreuve de décision sur les propriétés perceptives

Les 11 propriétés « perceptives » que nous avons utilisées sont les suivantes :

Fermé	Anguleux	Courbé	Arrondi
Continu	Symétrique	Pointu	Spiralé
Incliné	Long	Large	

Ces propriétés sont qualifiées de perceptives car elles ne dépendent que d'indications géométriques sur les chiffres manuscrits. Des algorithmes de traitement d'image permettent en effet d'indiquer si une figure est totalement fermée (comme un 0 ou un 8), si elle présente des angles, et si ceux-ci sont aigus (ou « pointus »), ou s'il y a des symétries (0, 3 ou 8).

Afin d'éviter des interprétations de ces termes différentes des sens que nous cherchons à leur donner, nous avons précisé le sens de chacune des propriétés perceptives lors de la consigne.

Epreuve de décision sur les propriétés sémantiques.

Les 12 propriétés « sémantiques » utilisées sont données par le tableau suivant :

Féminin	Masculin	Fort	Faible
Porte-bonheur	Sacré	Sensuel	Harmonieux
Doux	Stable	Sifflant	Complexe

Ces propriétés sont dites « sémantiques » car, contrairement aux précédentes, elles font intervenir des informations issues d'autres modalités sensorielles ou de connaissances complexes. Ainsi, nous avons cherché à voir si un chiffre manuscrit présenté visuellement pouvait avoir un lien avec l'audition à travers son expression langagière, en demandant d'indiquer si la première consonne du chiffre lu était « sifflante » ou « grésillante » comme *zéro, cinq, six, sept*. Nous avons aussi proposé des informations d'origine tactile : un chiffre doux. Les représentations kinesthésiques ont également été étudiées avec une décision sur la complexité d'écriture d'un chiffre par rapport aux autres. Nous avons cherché à mettre en jeu des connaissances mathématiques de base sur les chiffres, en demandant si un chiffre était fort (9, 8, 7...) et faible (0, 1, 2...). Un jugement esthétique sur les chiffres a été proposé : un chiffre harmonieux. Nous avons fait appel à l'imagerie mentale en demandant d'indiquer si un chiffre est stable ou non, c'est-à-dire que si ce chiffre, matérialisé, tient sur place ou se renverse. Nous avons encore voulu étudier les connotations culturelles ou religieuses particulières que pouvaient avoir certains chiffres : un chiffre sacré. Nous avons demandé si des chiffres étaient associés à une valeur de superstition : un chiffre porte-bonheur. Enfin, nous avons proposé des valeurs que nous voulions très subjectives en demandant aux sujets si des chiffres pouvaient être associés à des notions de sensualité, de féminité ou de masculinité.

Sujets

15 personnes volontaires, âgées de 22 à 40 ans, participèrent à cette expérience. Parmi celles-ci se trouvaient 7 femmes et 8 hommes. Chaque sujet a passé à la fois la phase de décision perceptive et celle de décision sémantique.

Procédure

Après la présentation de la consigne, chaque sujet était placé devant l'ordinateur et devait observer attentivement le centre de l'écran. Un point de fixation apparaissait pour préciser au sujet de se concentrer. Puis un chiffre était présenté sous forme d'une police manuscrite. Au bout de 500 ms, ce chiffre disparaissait pour laisser place à une propriété, perceptive ou sémantique suivant la phase expérimentale en cours. Le logiciel d'expérimentation que nous avons programmé se chargeait de présenter aléatoirement un chiffre et une propriété, de telle sorte que chaque chiffre soit associé à une propriété différente au cours de la phase expérimentale. Nous avons donc enregistré 10×11 réponses (nature des réponses et temps de réaction) en phase de décision perceptive et 10×12 réponses en phase de décision sémantique.

La consigne prescrivait aux sujets de répondre au clavier par « oui » ou « non » selon que la propriété correspondait ou non au chiffre présenté juste avant. Les sujets étaient invités à répondre le plus rapidement possible selon ce qu'ils considéraient comme étant le plus juste, aucune réponse n'étant réellement vraie ou fausse, en particulier pour les propriétés sémantiques.

RESULTATS

Les temps de réponse recueillis sont les suivants (par ordre croissant) :

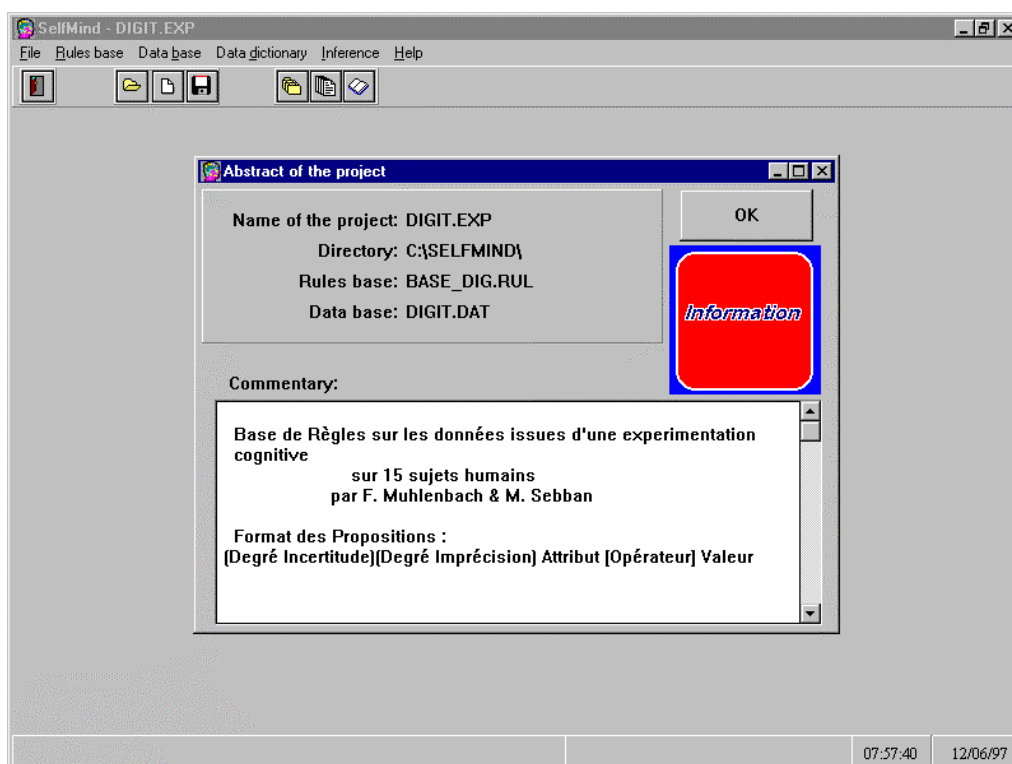
Propriétés	Temps de réponse (en ms)	Catégories des descripteurs
Porte-bonheur	1343.8	Sémantique
Fort	1352.8	Sémantique
Fermé	1355.3	Perceptif
Anguleux	1357.4	Perceptif
Arrondi	1393.7	Perceptif
Continu	1427.9	Perceptif
Courbé	1430.8	Perceptif
Symétrique	1463.4	Perceptif
Pointu	1467.5	Perceptif
Spiralé	1467.7	Perceptif
Doux	1471.2	Sémantique
Faible	1492.1	Sémantique
Féminin	1517.5	Sémantique
Incliné	1532.7	Perceptif
Sacré	1537.9	Sémantique
Sensuel	1546.9	Sémantique
Stable	1570.9	Sémantique
Sifflant	1605.7	Sémantique
Complexe	1608.5	Sémantique
Harmonieux	1617.7	Sémantique
Long	1648.8	Perceptif
Large	1683.4	Perceptif
Masculin	1750.1	Sémantique

La nature des réponses servira à construire la base de règles, et les temps de réaction seront un indice du sens de l'implication des associations faites entre les diverses propriétés. Nous présentons la base de règles issue de cette expérience en annexe 2.

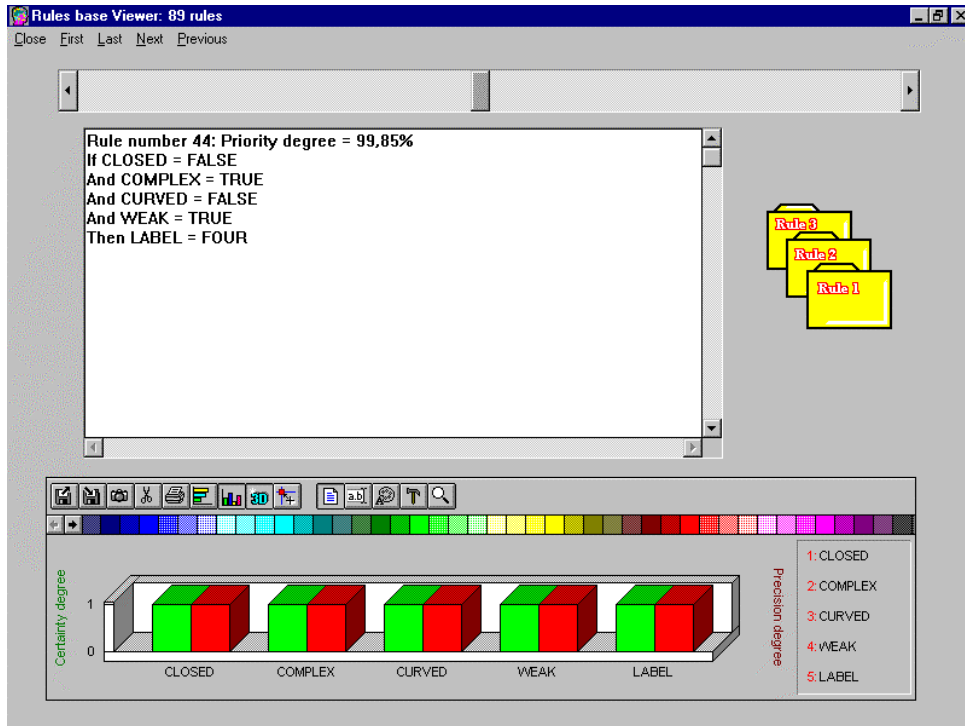
DISCUSSION

Nous remarquons que la majorité des descripteurs perceptifs apparaissent dans notre tableau avant les descripteurs sémantiques, ce qui nous conforte dans l'idée que des traitements plus élaborés mettent en jeu des temps de computation plus importants. En ce qui concerne les résultats qui dérogent à cette règle, nous aurions quelques explications possibles. La propriété sémantique « porte-bonheur » occupe le haut du tableau car elle fait appel à un élément spontané et irréfléchi — irrationnel —, à la différence de « sacré » qui met en jeu une valeur culturelle d'où un temps de réponse plus important. La propriété sémantique « fort » est aussi vite décidable, car elle fait appel à des notions basiques sur les mathématiques et la décision se fait vis-à-vis d'un seuil (fixé à 5 par exemple), mais nous ne comprenons pas très bien dans ce cas pourquoi le terme « faible » se retrouve plus bas dans la liste. D'autre part, des informations demandant des comparaisons graphiques avec une barre verticale (« incliné ») ou les autres chiffres (« long » et « large ») font de ces propriétés perceptives des termes décidés assez lentement.

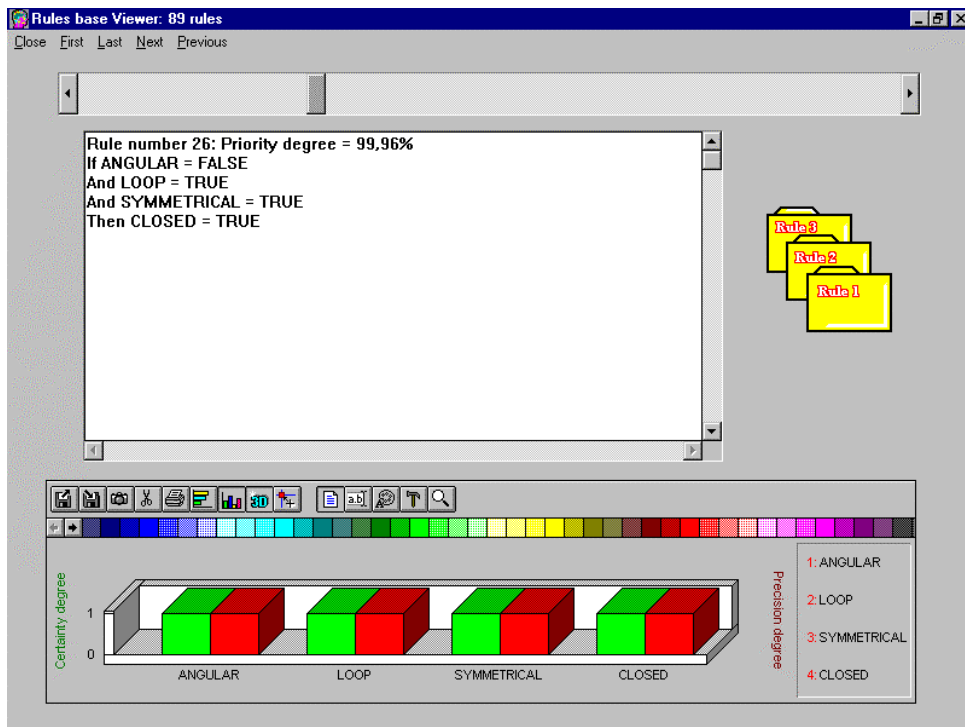
Nous présentons maintenant l'utilisation de la base de connaissances ainsi conçue dans notre logiciel. Le projet s'ouvre avec les indications suivantes :



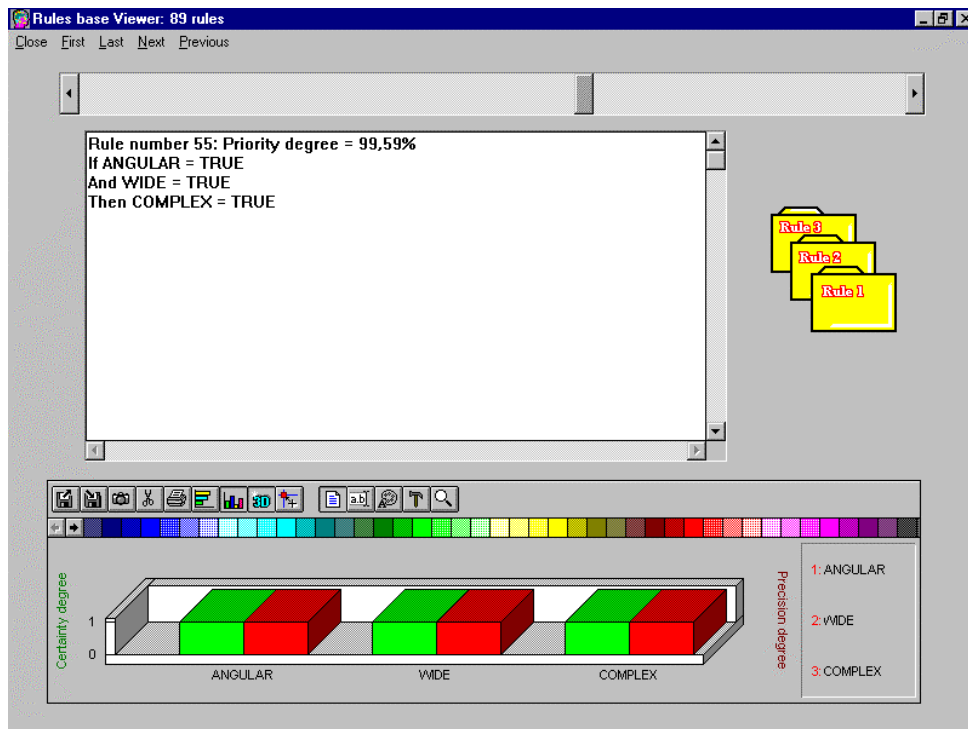
Les règles de la base aboutissent à la dénomination des chiffres :



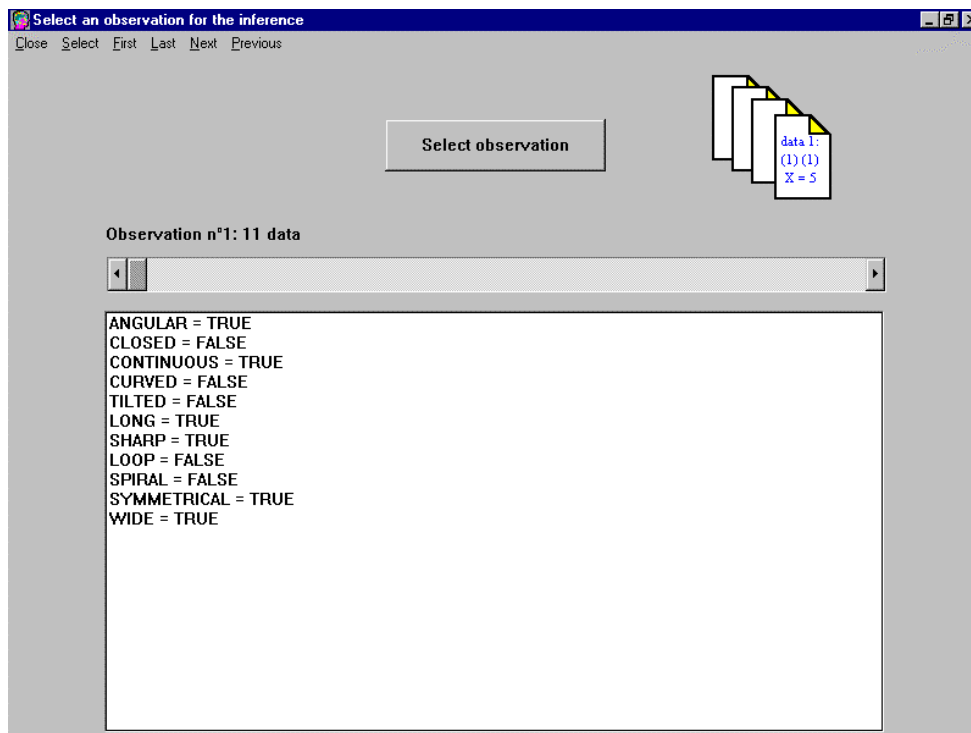
Pour cela, il y a des règles entre descripteurs perceptifs :



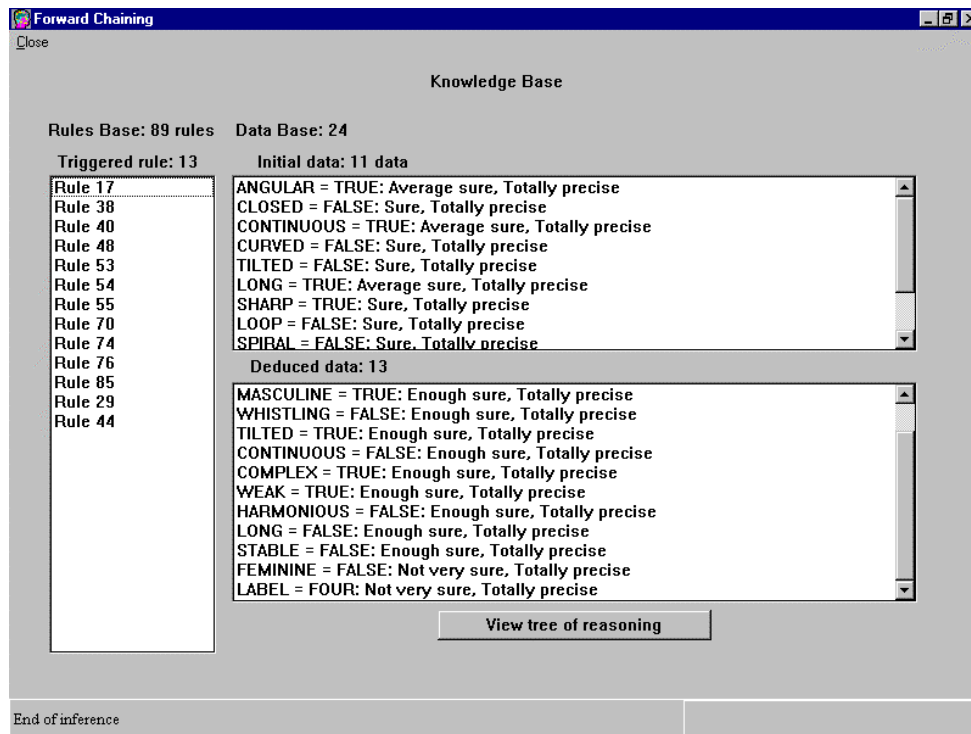
Des règles amènent à des descripteurs sémantiques à partir de descripteurs perceptifs :



Enfin, des règles entre descripteurs sémantiques existent aussi dans la base. Les propriétés observées sont d'ordre perceptif, et l'inférence consistera à attribuer si possible un nom au chiffre via la génération de propriétés sémantiques.



Exemple de dénomination d'un chiffre à partir de propriétés perceptives :



Les résultats que nous obtenons à partir de notre modèle restent quand même assez modestes. En effet, dans quelques cas, nous n'arrivons pas à aboutir à l'étape de dénomination alors que les propriétés perceptives fournies en entrées permettraient théoriquement de déduire un chiffre particulier. Nous attribuons cette faiblesse à la construction de notre base de connaissances à partir des descripteurs employés lors de notre expérience.

Notre choix de descripteurs a peut-être été maladroit car, avec deux valeurs possibles pour chaque descripteur perceptif (« vrai » ou « faux »), nous aurions pu discriminer à l'extrême 2^{11} éléments (= 2048), ce qui aurait été largement suffisant pour les 10 chiffres arabes que nous utilisons. De plus, le relais apporté par les 12 descripteurs sémantiques se combinant entre eux donnerait un nombre de possibilités encore plus conséquent.

Cependant notre modèle reste intéressant dans le sens où nous arrivons à retrouver des associations possibles entre des propriétés jugées très subjectives. Les performances moyennes du système expert en reconnaissance de formes sont liées au type d'activité que nous avons choisi de modéliser. La dénomination d'un chiffre est en effet une activité très courante ; elle résulte d'un apprentissage et consiste en une activité de lecture dont les caractères automatiques sont connus — l'effet *Stroop* en étant un exemple révélateur classique.

Troisième partie :

Propositions méthodologiques pour une modélisation des aspects cognitifs dans la perception

Des expériences de psychologie cognitive comme source d'information pour la constitution de bases de connaissances

L'expérience que nous venons de décrire dans la deuxième partie de notre rapport peut s'inscrire dans une approche méthodologique plus globale de modélisation des aspects cognitifs dans la perception. En effet, SelfMind utilisé indépendamment présente un intérêt moindre que s'il est considéré comme un simple module d'un modèle plus global.

Certes, SelfMind, comme tout moteur d'inférence, peut évidemment avoir une application pratique de système expert car il est un outil adapté aux problèmes d'aide à la décision, à condition de disposer d'une base de connaissances bien constituée sur ce domaine. Dans ce genre de situation, il est effectivement tout aussi important d'avoir le raisonnement qui a abouti à une conclusion que d'obtenir au final cette conclusion. Cette problématique demande ainsi un travail important, difficile et onéreux auprès d'experts afin d'explicitier et de formaliser leurs connaissances.

Cependant notre objectif n'est pas de modéliser des connaissances spécialisées dans un domaine d'expertise mais de concevoir un modèle rendant compte des processus cognitifs dans la perception. Nous cherchons à voir comment les informations sensorielles peuvent amener à des représentations signifiantes et comment s'effectuent les opérations sur ces représentations.

Pour une telle modélisation, il nous semble opportun de nous inscrire dans une démarche débutant par l'extraction des informations sémantiques associées à la perception. À cette fin, nous proposons une expérience se déroulant en deux étapes pour tester les sujets humains dans une situation où ils doivent effectuer des associations entre des stimuli d'une modalité sensorielle donnée et des propriétés sémantiques.

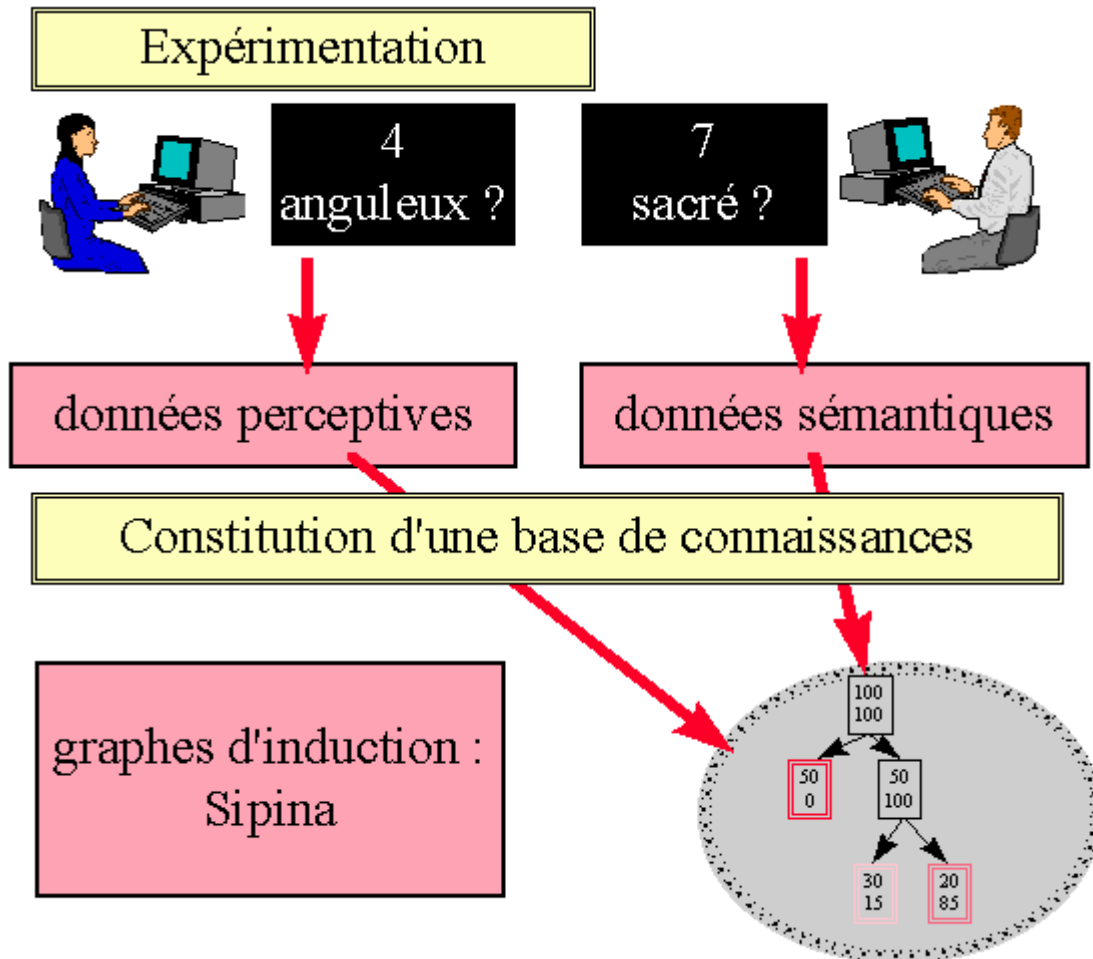
Nous proposons dans une première étape de demander aux individus testés d'indiquer librement les propriétés qui leur viennent à l'esprit lors de la présentation des stimuli. Les réponses recueillies devront constituer un échantillon suffisamment important de propriétés sémantiques. À partir de traitements statistiques, seules les réponses les plus significatives sont retenues, et ces traitements peuvent même servir dans la génération d'indices de précision des informations.

Les informations les plus pertinentes constituent un échantillon de propriétés utilisées dans la seconde étape. Dans cette nouvelle expérience, ces propriétés sélectionnées sont présentées en même temps que les stimuli étudiés. Les sujets ont alors à décider si ces

propriétés correspondent aux stimuli ou non le plus rapidement possible, car les temps de réponse sont enregistrés.

Ces temps de réponse sont utilisés pour l'agencement des propriétés, en considérant que les temps donnés sont représentatifs de la profondeur des traitements sur les représentations mentales. Comme les règles d'un système expert sont des relations d'implication entre les faits, cette notion d'ordre suivant la précocité des traitements est importante. En ce qui concerne les propriétés à relier entre elles, nous proposons d'utiliser des outils d'apprentissage symbolique — tels que les arbres ou graphes d'induction — qui cherchent à identifier une variable (dans notre cas, une propriété très élaborée) en fonction d'autres variables (des propriétés traitées plus précocement).

Ce passage de la connaissance humaine à une base de règles utilisable dans des outils de modélisation est représenté ci-dessous :

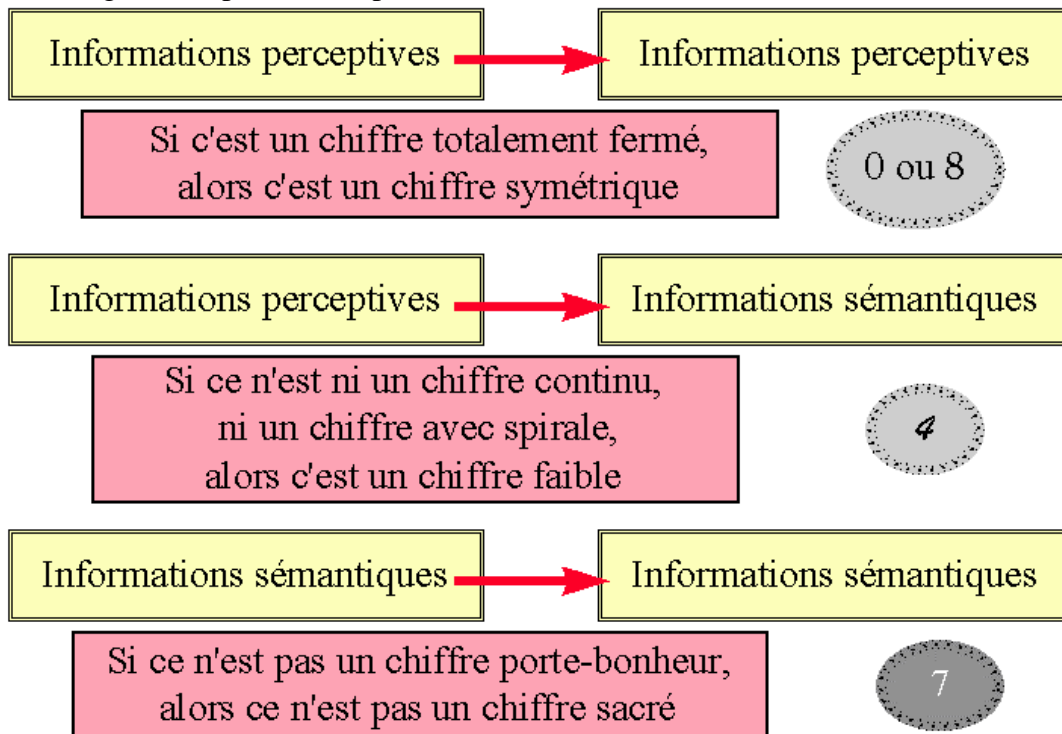


Sur ce schéma, nous reprenons l'expérience réalisée en perception visuelle sur la dénomination des caractères manuscrits. Cependant, nous pouvons imaginer réaliser des expériences dans d'autres modalités sensorielles en suivant la même démarche. Il serait même très intéressant de pouvoir assembler des règles d'origines sensorielles diverses et, à partir de cet ensemble, d'élaborer des règles de traitements descendants amenant les propriétés sémantiques amodales à retrouver des propriétés sensorielles.

Extraction automatique de connaissances à partir de données

Les arbres et graphes d'induction sont des techniques d'apprentissage supervisé. Ces méthodes d'apprentissage symbolique permettent de concevoir des modèles qui s'expriment sous forme de règles de production où la partie prémisses correspond à une conjonction de propositions et où la conclusion correspond à l'une des classes étudiées (Rabaséda, Rakotomalala, & Zighed, 1996 ; Rabaséda, 1996). Dans notre cas, ces propositions correspondent à des propriétés sensorielles ou sémantiques.

Le logiciel Sipina, développé par Zighed, Auray et Duru (1992) a ainsi été utilisé pour générer les règles d'implication exploitées dans SelfMind :



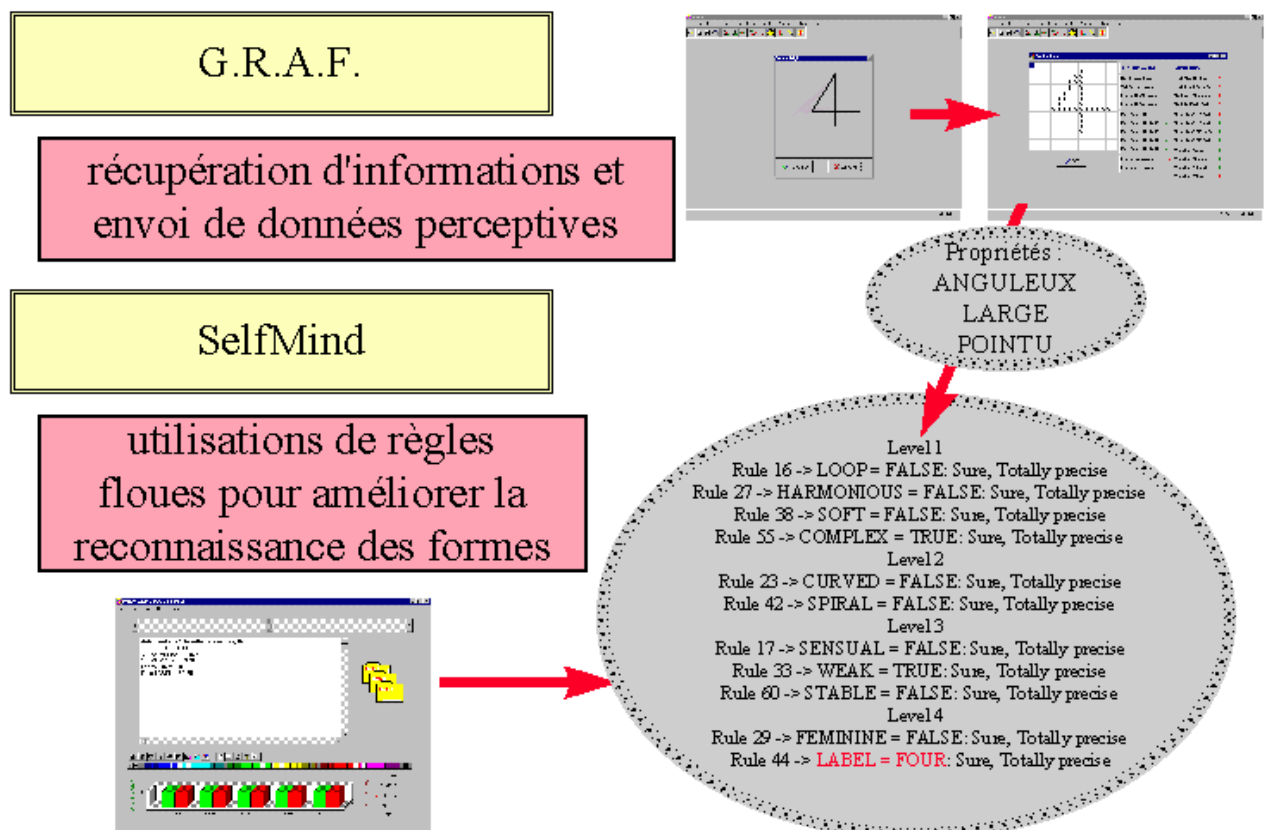
Le logiciel Sipina offre la possibilité de produire des règles au format exploitable par SelfMind. Sipina reproduit aussi l'imperfection des données qui fait la particularité du fonctionnement du logiciel SelfMind. Tout d'abord, les règles produites par Sipina sont fournies avec des degrés d'incertitude sur les conclusions qui sont fonction du pourcentage d'individus bien classés. D'autre part, chaque règle est associée à une intensité d'implication (Rakotomalala, Zighed, & Rabaséda, 1996) qui représente la force du lien unissant les prémisses à la conclusion. Cette intensité d'implication est assimilable aux degrés de priorité qui ordonnent les règles dans la base.

SelfMind : partie symbolique du modèle hybride de perception intelligente “SIMPL”

Nous avons indiqué que l’approche adoptée pour la modélisation des processus cognitifs de la perception se voulait symbolique afin de rendre compte des étapes de traitements de haut niveau. Mais il serait illusoire de considérer SelfMind comme représentatif d’un modèle de la perception complète en raison de l’inadéquation à représenter les traitements de bas niveau. C’est pourquoi nous insistons sur l’emploi d’un modèle comprenant un module de traitements numériques, dont la fonction est d’élaborer des propriétés perceptives à partir de traits des stimuli, et d’un module de traitements sémantiques qui effectue des traitements sur des représentations complexes.

Un tel modèle a été développé par Sebban dans le cadre de sa thèse de doctorat (Sebban, 1996). Notre moteur d’inférence a ainsi été mêlé à son logiciel appelé GRAF — comme *Géométrie en Reconnaissance Automatique de Formes* — pour composer la plateforme informatique SIMPL — *Système Intelligent Multimodal pour la Perception Logique* —.

Au sein de SIMPL, le fonctionnement de GRAF et SelfMind sur l’application des chiffres manuscrits est représenté par le schéma suivant :



Le module GRAF effectue les traitements perceptifs sur les stimuli sensoriels. Il contient des ensembles appelés *Mémoire Perceptive* et *Mémoire Etiquettes*, et gère l'apprentissage de nouveaux exemples grâce à des méthodes géométriques.

Lors de la présentation d'un stimulus sensoriel (comme le dessin d'un chiffre manuscrit ou les valeurs de propriétés physico-chimiques d'une molécule odorante), des traits sensoriels sont projetés dans la *Mémoire Perceptive* où des connexions s'établissent avec les expériences sensorielles déjà stockées. De là, la représentation sensorielle entrée hérite de propriétés perceptives stockées en *Mémoire Etiquettes* avec un certain degré d'incertitude.

Les propriétés déduites constituent une observation de faits. Cette base de faits est traitée par la partie symbolique constituée par le module SelfMind. De nouveaux faits sont ainsi déduits par des associations symboliques issues de la base de règles.

Le modèle SIMPL, avec ses parties numérique et symbolique, a été inspiré d'une architecture fonctionnelle développée par Kosslyn et Koenig (1992). Dans l'architecture proposée par ces auteurs, SelfMind pourrait modéliser la mémoire associative, dont une partie des informations sont communes aux différentes modalités sensorielles, ainsi que les traitements du sous-système de recherche top-down, des opérations effectués sur des représentations mentales complexes.

Conclusion

L'approche proposée dans ce rapport rend compte d'une observation à laquelle j'avais abouti au cours de mes premières années universitaires. Alors étudiant en informatique, je trouvais que les modélisations proposées par l'intelligence artificielle étaient trop éloignées des aspects humains. Cette constatation fut à l'origine de ma réorientation en psychologie afin de mieux chercher à appréhender nos capacités cognitives. Le travail réalisé cette année concilie ces deux domaines de recherche en s'efforçant de défendre l'intérêt d'une vision pluridisciplinaire : nous combinons des expériences issues de la psychologie cognitive, des connaissances extraites de ces expériences par des outils d'apprentissage symbolique et numérique, et nous participons à un modèle hybride gérant de manière souple ces informations.

Le logiciel SelfMind que nous avons programmé a démontré certaines capacités à modéliser des aspects d'une architecture cognitive, avec la possibilité de générer des propriétés de type « associations d'idées ». L'apport de notre logiciel a été de présenter un mode d'utilisation de connaissances imparfaites en se basant sur une logique non classique.

D'autre part, la méthode expérimentale proposée permet de modéliser le fonctionnement de la logique du sens commun par moyennage des subjectivités individuelles. Cette méthode nous amène à produire des règles n'ayant pas nécessairement de caractère logique, mais dont la qualité est modulée par certains paramètres.

En outre, la nécessaire intégration de SelfMind avec un système périphérique de recueil des faits a été possible en incluant notre moteur d'inférence dans le modèle hybride SIMPL qui gère de manière complémentaire les traitements de bas niveau. Une perspective s'ouvre à nous en cherchant à dépasser le stade fonctionnaliste afin de concevoir un modèle qui soit non seulement valable d'un point de vue fonctionnel mais aussi plausible selon nos connaissances neurobiologiques sur le cerveau.

Les sciences cognitives s'intéressent évidemment au versant matériel nécessaire à la cognition, à savoir le système nerveux. Dans une optique de traitement de l'information, les modélisations neurobiologiques se sont surtout évertuées à rendre compte des aspects électrophysiologiques que démontrent les cellules du système nerveux. Des propriétés membranaires des neurones sous forme de circuit électrique RC (résistance et condensateur), les modélisations sont allées jusqu'à décrire des fonctionnements de populations de neurones connectés les uns aux autres. Or les réseaux de neurones modélisés par les neurosciences sont restés prudemment associés à des fonctionnements de très bas niveau, établissant un lien entre une activité physique se déroulant dans l'environnement et une modification des propriétés

biophysiques des récepteurs sensoriels et des cellules connectées à ces derniers. La perception constitue en effet l'un des domaines d'étude privilégiés pour l'investigation de la cognition. La formalisation des réseaux de neurones a ensuite connu un développement à part car ils présentent des propriétés qui font d'eux de bons candidats comme outils de classification. Les réseaux de neurones formels ont alors été employés dans des applications industrielles, mais l'origine biologique de ces objets mathématiques n'avait plus qu'un sens historique. Les réseaux de neurones formels sont ainsi apparus en sciences cognitives comme des outils puissants par leurs possibilités de simulation et leurs liens avec le vivant, même s'ils ne sont que des représentations schématiques des neurones biologiques.

Nous espérons par conséquent qu'une modélisation du fonctionnement du système nerveux sera faite d'une manière qui rende mieux compte des propriétés biologiques que ne le font actuellement les réseaux de neurones formels. À ce niveau d'abstraction du neuromimétisme, le logiciel SelfMind présente des caractéristiques comparables à celles d'approches connexionnistes classiques.

Tout d'abord, les capacités d'apprentissage d'un réseau de neurones formels proviennent de l'arrangement du poids des connexions entre les unités lors de la phase d'entraînement du réseau. Pour SelfMind, les connaissances proviennent de l'élaboration de liens (les règles de production) qui sont construites par apprentissage symbolique. L'apprentissage dont bénéficie SelfMind provient d'un autre logiciel tel que Sipina mais n'est pas très éloigné du fonctionnement psychologique puisqu'il consiste à retrouver des régularités à partir d'exemples observés. L'apprentissage symbolique est même assez proche du fonctionnement neuronal par certains aspects car les modes de calcul employés consistent à passer de liaisons fonctionnelles à des relations implicatives, analogues en cela à la règle de Hebb et en sa manifestation neurophysiologique au travers de la potentialisation à long terme.

En ce qui concerne l'activation d'une unité, celle-ci se produit dans un réseau de neurones formels lorsque l'ensemble des activités excitatrices (ou inhibitrices) pondérées des entrées arrivant à cette unité dépasse un certain seuil. Dans SelfMind, une unité est une propriété porteuse de sens, c'est-à-dire un « fait » selon notre terminologie. Un fait est déduit seulement si le degré de certitude calculé est supérieur à un seuil minimal fixé dans le logiciel en tant que paramètre global. De plus, l'arrangement des règles au sein de la base selon le degré de priorité est garant d'une déduction des propriétés aux plus hauts degrés de certitude avant celles de degrés de certitude moindre, si la base de règles est convenablement conçue.

Une critique est habituellement faite aux architectures cognitives classiques dont SelfMind est issu : le mode de fonctionnement séquentiel des traitements, en désaccord avec le système nerveux qui procède de manière massivement parallèle. Nous répondons à cette dernière que dans notre logiciel, la déduction des faits s'effectue en parallèle au sein d'un niveau de raisonnement. Le parcours des règles de la base se produit effectivement en « largueur d'abord » — assimilable en cela aux calculs s'effectuant pour une couche donnée

d'un réseau de neurones formels — et les faits déduits servent d'entrées (ou de prémisses) à l'origine du déclenchement des règles à un niveau supérieur — comme des calculs sur une couche de neurones formels plus profonde —.

Enfin, avec une architecture telle que SelfMind, nous pouvons traiter de problèmes plus généraux que ceux auxquels s'attaquent les réseaux de neurones, limités par construction dans la simulation d'une activité particulière avec un nombre déterminé d'exemples apprenables. Dans notre logiciel, il apparaît tout à fait possible de gérer aussi bien des bases de règles globales que des bases dont les règles sont spécialisées dans un type de traitement particulier. Cette spécialisation des traitements associée à des unités signifiantes n'est pas dénuée de crédibilité neurobiologique : la littérature neuropsychologie (e.g. Damasio, 1990) fait bien état de déficits spécifiques à des catégories de traitements sémantiques.

Cette conclusion nous amène à avoir une vision critique de l'emploi de modèles qu'effectuent certaines disciplines en allant rechercher des informations auprès d'autres domaines de recherche. Bien souvent, hélas, ces dernières procèdent à des adaptations et des simplifications abusives de ce qui ne demeure que des propositions de modèles. Il faut plutôt chercher à encourager les collaborations transdisciplinaires en sciences cognitives, en prenant garde que la vision des choses présentées ne soit partielle. Les problèmes fondamentaux que rencontre chacun de ces domaines sont essentiels car les questions et les différences d'interprétations constituent les éléments de base de toute recherche scientifique.

Annexe 1 :

Liste des règles générées par SIPINA, CART et ID3 sur les Odeurs

Règle 1 :

IF (0.5)(0.3) Fragrant = True
Then (0.1)(0.1) Fruity_Citrus=True [0.9597];

Règle 2 :

IF (0.3)(0.3) Sweet = True
Then (0.1)(0.1) Fruity_Other_Citrus = True [0.9474];

Règle 3 :

IF (0.3)(0.3) Fragrant = True
Then (0.1)(0.1) Perfumery = True [0.9999];

Règle 4 :

IF (0.3)(0.3) Floral = True
Then (0.3)(0.3) Perfumery = True [0.9810];

Règle 5 :

IF (0.3)(0.3) Sweet = True
Then (0.3)(0.3) Fragrant = True [0.9579];

Règle 6 :

IF (0.1)(0.3) Fragrant = True
Then (0.1)(0.3) Aromatic = True [0.9999];

Règle 7 :

IF (0.1)(0.1) Cool = True
And (0.1)(0.1) Sweet = True
Then (0.1)(0.1) Fragrant = True [0.9999];

Règle 8 :

IF (0.1)(0.1) Rancid = True
And (0.1)(0.1) Warm = True
Then (0.1)(0.1) Spicy = True [0.9413];

Règle 9 :

IF (0.3)(0.3) Rancid = True
Then (0.1)(0.1) Spicy = True [0.9801];

Règle 10:

IF (0.7)(0.7) Sweet = True
Then (0.1)(0.1) Woody = True [0.9998];

Règle 11 :

IF (0.5)(0.5) Fragrant = True
And (0.5)(0.5) Sweet = True
Then (0.1)(0.1) Woody = True [0.9944];

Règle 12 :

IF (0.1)(0.1) Cool = True
And (0.3)(0.3) Warm = True
Then (0.1)(0.1) Minty = True [0.9040];

Règle 13 :

IF (1)(0) Cool = True
Then (1)(0) Minty = True [0.9999];

Règle 14 :

IF (0.3)(0.3) Aromatic = True
And (0.1)(0.1) Seasoning = True
And (0.3)(0.3) Cool = True
Then (0.1)(0.1) Minty = True [0.9990];

Règle 15 :

IF (0.1)(0.1) Cool = True
Then (0.1)(0.1) Sweet = True [0.9999];

Règle 16 :

IF (0.3)(0.3) Alcoholic = True
Then (0.7)(0.7) Anaesthetic = True [0.8577];

Règle 17 :

IF (0.1)(0.1) Chemical = True
And (0.1)(0.1) Medicinal = True
Then (0.1)(0.1) Anaesthetic = True [0.9999];

Règle 18 :

IF (0.3)(0.7) Medicinal = True
Then (0.3)(0.3) Anaesthetic = True [0.9682];

Règle 19 :

IF (0.1)(0.1) Cleaning_Fluid = True
And (0.1)(0.1) Medicinal = True
Then (0.1)(0.1) Anaesthetic = True [0.9998];

Règle 20 :

IF (0.3)(0.3) Chemical = True
Then (0.1)(0.1) Cleaning_Fluid = True [0.9413];

Règle 21 :

IF (0.1)(0.1) Cleaning_Fluid = True
Then (0.1)(0.1) Solvent = True [0.9523];

Règle 22 :

IF (0.1)(0.1) Rotten_Fruit = True
Then (0.1)(0.1) Perfumery = True [0.9067];

Règle 23 :

IF (0.3)(0.3) Earthy = True
Then (0.5)(0.5) Fresh_Green_Vegetables = True [0.8577];

Règle 24 :

IF (0.3)(0.3) Fruity_Other_Citrus = True
Then (0.1)(0.1) Earthy = True [0.9818];

Règle 25 :

IF (0.3)(0.3) Sickening = True
Then (0.1)(0.1) Rotten_Fruit = True [0.9942];

Règle 26 :

IF (0.1)(0.1) Cleaning_Fluid = True
And (0.1)(0.1) Pungent = True
Then (0.1)(0.1) Chemical = True [0.8317];

Règle 27 :

IF (0.1)(0.1) Alcoholic = True
And (0.1)(0.1) Fruity_Other_Citrus = True
And (0.1)(0.1) Sickening = True
Then (0.1)(0.1) Smoky = True [0.9913];

Règle 28 :

IF (0.3)(0.3) Smoky = True
Then (0.3)(0.3) Burnt = True [0.9942];

Règle 29 :

IF (0.3)(0.3) Fruity_Other_Citrus = True
And (0.1)(0.1) Sickening = True
Then (0.1)(0.1) Smoky = True [0.9092];

Règle 30 :

IF (0.1)(0.1) Anaesthetic = True
And (0.1)(0.1) Minty = True
Then (0.1)(0.1) Medicinal = True [0.9998];

Règle 31 :

IF (0.1)(0.1) Anaesthetic = True
Then (0.1)(0.1) Chemical = True [0.9956];

Règle 32 :

IF (0.7)(0.1) Anaesthetic = True
Then (0.3)(0.1) Sharp = True [0.999];

Règle 33 :

IF (0.1)(0.1) Chemical = True
And (0.1)(0.1) Sickening = True
Then (0.1)(0.1) Sharp = True [0.9999];

Règle 34 :

IF (0.1)(0.1) Rancid = True
And (0.1)(0.1) Sickening = True
Then (0.1)(0.1) Vinegar = True [0.9962];

Règle 35 :

IF (0.5)(0.5) Rancid = True
Then (0.3)(0.3) Vinegar = True [0.9787];

Règle 36 :

IF (0.1)(0.1) Fragnant = True
Then (0.1)(0.1) Meaty = True [0.9999];

Règle 37 :

IF (0.3)(0.3) Fatty = True
Then (0.1)(0.1) Meaty = True [0.9879];

Règle 38 :

IF (0.3)(0.3) Rancid = True
Then (0.3)(0.3) Sweaty = True [0.9999];

Règle 39 :

IF (0.7)(0.7) Sickening = True
Then (0.5)(0.5) Sweaty = True [0.8146];

Règle 40 :

IF (0.1)(0.1) Sickening = True
Then (0.1)(0.1) Rancid = True [0.9939];

Règle 41 :

IF (0.3)(0.1) Sickening = True
Then (0.1)(0.1) Putrid = True [0.9888];

Règle 42 :

IF (0.1)(0.1) Heavy = True
And (0.1)(0.1) Putrid = True
Then (0.1)(0.1) Sickening = True [0.9994];

Règle 43 :

IF (0.1)(0.1) Fruity_Citrus = True
And (0.1)(0.1) Minty = True
Then (0.1)(0.1) Cool = True [0.9659];

Règle 44 :

IF (0.5)(0.5) Cool = True
Then (0.5)(0.5) Minty = True [0.9888];

Règle 45 :

IF (0.3)(0.1) Spicy = True
Then (0.1)(0.1) Warm = True [0.9289];

Règle 46 :

IF (0.3)(0.1) Aromatic = True
Then (0.1)(0.1) Warm = True [0.8767];

Règle 47 :

IF (0.3)(0.3) Alcoholic = True
Then (0.3)(0.3) Rotten_Fruit = True [0.9767];

Règle 48 :

IF (0.1)(0.1) Fruity_Other_Citrus = True
And (0.7)(0.0) Sharp = True
Then (0.7)(0.0) Acid = True [0.9512];

Règle 49 :

IF (0.1)(0.1) Cool = True
And (0.1)(0.5) Herbaceous = True
Then (0.1)(0.5) Woody = True [0.9053];

Règle 50 :

IF (0.5)(0.5) Fatty = True
Then (0.5)(0.5) Corporal = True [0.9752];

Règle 51 :

IF (0.5)(0.5) Cool = True
Then (0.5)(0.5) Heady = True [0.9022];

Règle 52 :

IF (0.1)(0.1) Salty = True
Then (0.1)(0.1) Smoky = True [0.9222];

Règle 53 :

IF (1.0)(1.0) Fruity_Other_Citrus = True
Then (1.0)(0.0) Chemical = True [0.9999];

Annexe 2 :

Liste des règles générées par SIPINA sur les chiffres manuscrits.

Par F. Muhlenbach & M. Sebban

Règles entre descripteurs sémantiques

If (1)(1) Harmonious = True
And (1)(1) Sensual = True
And (1)(1) Stable = False
Then (1)(1) Complex = False [0.9993];

If (1)(1) Sensual = True
Then (1)(1) Soft = True [0.9999];

If (1)(1) Sensual = False
Then (1)(1) Soft = False [0.9999];

If (1)(1) Strong = False
Then (1)(1) Weak = True [0.9998];

If (1)(1) Strong = True
Then (1)(1) Weak = False [0.9999];

If (1)(1) Sensual = True
Then (1)(1) Feminine = True [0.9995];

If (1)(1) Sensual = False
Then (1)(1) Feminine = False [0.9995];

If (1)(1) Lucky_Charm = False
And (1)(1) Masculine = False
And (1)(1) Sacred = False
And (1)(1) Sensual = False
And (1)(1) Whistling = False
Then (1)(1) Strong = False [0.9605];

If (1)(1) Complex = False
And (1)(1) Sacred = False
And (1)(1) Whistling = True
Then (1)(1) Lucky_Charm = False [0.8241];

If (1)(1) Complex = False
And (1)(1) Sacred = False
And (1)(1) Sensual = True
And (1)(1) Stable = False
And (1)(1) Whistling = False
Then (1)(1) Lucky_Charm = False [0.8689];

If (1)(1) Sacred = False
And (1)(1) Sensual = False
And (1)(1) Whistling = False
Then (1)(1) Lucky_Charm = False [0.9951];

If (1)(1) Lucky_Charm = False
Then (1)(1) Sacred = False [0.9790];

If (1)(1) Feminine = True
And (1)(1) Soft = True
Then (1)(1) Sensual = True [0.9999];

If (1)(1) Harmonious = False
And (1)(1) Soft = False
And (1)(1) Strong = False
Then (1)(1) Sensual = False [0.9999];

If (1)(1) Strong = False
And (1)(1) Weak = True
Then (1)(1) Whistling = False [0.9985];

If (1)(1) Complex = True
And (1)(1) Harmonious = False
And (1)(1) Sacred = False
And (1)(1) Soft = False
Then (1)(1) Stable = False [0.9506];

If (1)(1) Complex = False
And (1)(1) Sensual = True
And (1)(1) Soft = True
Then (1)(1) Harmonious = True [0.9999];

If (1)(1) Sensual = False
Then (1)(1) Harmonious = False [0.9999];

If (1)(1) Feminine = True
Then (1)(1) Masculine = False [0.9986];

If (1)(1) Feminine = False
And (1)(1) Sacred = True
Then (1)(1) Masculine = True [0.9951];

Règles entre descripteurs perceptifs

If (1)(1) Round = True
Then (1)(1) Angular = False [0.9999];

If (1)(1) Continuous = False
And (1)(1) Round = False
And (1)(1) Tilted = False
Then (1)(1) Angular = True [0.9882];

If (1)(1) Round = False
And (1)(1) Tilted = True
Then (1)(1) Angular = True [0.9998];

If (1)(1) Angular = False
And (1)(1) Round = True
And (1)(1) Symmetrical = True
Then (1)(1) Closed = True [0.9996];

If (1)(1) Angular = False
Then (1)(1) Continuous = True [0.9993];

If (1)(1) Angular = True
And (1)(1) Curved = False
And (1)(1) Tilted = False
Then (1)(1) Continuous = False [0.9962];

If (1)(1) Round = True
Then (1)(1) Curved = True [0.9998];

If (1)(1) Continuous = False
And (1)(1) Round = False
And (1)(1) Pointed = True
Then (1)(1) Curved = False [0.9998];

If (1)(1) Pointed = False
And (1)(1) Round = False
And (1)(1) Symmetrical = True
Then (1)(1) Curved = True [0.9407];

If (1)(1) Continuous = True
And (1)(1) Pointed = True
And (1)(1) Round = False
And (1)(1) Symmetrical = False
Then (1)(1) Curved = False [0.9712];

If (1)(1) Angular = False
And (1)(1) Wide = True
Then (1)(1) Tilted = False [0.9811];

If (1)(1) Angular = True
And (1)(1) Continuous = True
And (1)(1) Round = False
Then (1)(1) Tilted = True [0.9966];

If (1)(1) Long = True
And (1)(1) Symmetrical = False
And (1)(1) Tilted = True
Then (1)(1) Wide = False [0.9820];

If (1)(1) Spiral = False
And (1)(1) Symmetrical = True
And (1)(1) Wide = True
Then (1)(1) Long = False [0.9683];

If (1)(1) Angular = True
And (1)(1) Wide = False
Then (1)(1) Long = True [0.9568];

If (1)(1) Round = True
Then (1)(1) Pointed = False [0.9999];

If (1)(1) Continuous = False
And (1)(1) Curved = False
And (1)(1) Round = False
Then (1)(1) Pointed = True [0.9998];

If (1)(1) Angular = False
And (1)(1) Pointed = False
Then (1)(1) Round = True [0.9998];

If (1)(1) Angular = True
Then (1)(1) Round = False [0.9999];

If (1)(1) Round = False
Then (1)(1) Spiral = False [0.9986];

If (1)(1) Closed = True
Then (1)(1) Symmetrical = True [0.9990];

Règles allant de descripteurs Perceptifs vers des descripteurs Sémantiques

If (1)(1) Angular = False
Then (1)(1) Complex = False [0.9445];

If (1)(1) Angular = True
And (1)(1) Long = True
And (1)(1) Wide = False
Then (1)(1) Complex = False [0.9445];

If (1)(1) Angular = True
And (1)(1) Wide = True
Then (1)(1) Complex = True [0.9959];

If (1)(1) Angular = False
And (1)(1) Round = True
Then (1)(1) Soft = True [0.9992];

If (1)(1) Angular = True
Then (1)(1) Soft = False [0.9989];

If (1)(1) Continuous = False
And (1)(1) Spiral = False
Then (1)(1) Weak = True [0.9993];

If (1)(1) Closed = False
And (1)(1) Continuous = True
And (1)(1) Spiral = False
And (1)(1) Symmetrical = True
Then (1)(1) Weak = true [0.9771];

If (1)(1) Angular = False
And (1)(1) Spiral = True
Then (1)(1) Weak = False [0.9977];

If (1)(1) Angular = False
And (1)(1) Continuous = True
And (1)(1) Round = True
Then (1)(1) Feminine = True [0.9973];

If (1)(1) Angular = True
And (1)(1) Spiral = False
And (1)(1) Wide = False
Then (1)(1) Feminine = False [0.9995];

If (1)(1) Pointed = False
And (1)(1) Symmetrical = True
Then (1)(1) Harmonious = True [0.9959];

If (1)(1) Long = False
And (1)(1) Round = False
And (1)(1) Spiral = False
Then (1)(1) Strong = False [0.9983];

If (1)(1) Spiral = True
And (1)(1) Symmetrical = False
And (1)(1) Tilted = False
Then (1)(1) Strong = True [0.9982];

If (1)(1) Continuous = False
And (1)(1) Pointed = True
Then (1)(1) Harmonious = False [0.9996];

If (1)(1) Angular = True
And (1)(1) Continuous = True
And (1)(1) Pointed = True
Then (1)(1) Harmonious = False [0.9695];

If (1)(1) Angular = False
Then (1)(1) Masculine = False [0.9957];

If (1)(1) Angular = True
And (1)(1) Long = True
Then (1)(1) Masculine = True [0.9987];

If (1)(1) Long = False
Then (1)(1) Lucky_Charm = False [0.9112];

If (1)(1) Round = True
Then (1)(1) Sacred = False [0.9843];

If (1)(1) Long = False
And (1)(1) Round = False
Then (1)(1) Sacred = False [0.9037];

If (1)(1) Angular = False
And (1)(1) Wide = True
Then (1)(1) Sensual = True [0.9990];

If (1)(1) Angular = True
And (1)(1) Spiral = False
Then (1)(1) Sensual = False [0.9999];

If (1)(1) Angular = False
And (1)(1) Long = True
And (1)(1) Pointed = False
And (1)(1) Wide = False
Then (1)(1) Sensual = True [0.9563];

If (1)(1) Long = True
And (1)(1) Spiral = False
And (1)(1) Tilted = False
Then (1)(1) Whistling = False [0.9982];

If (1)(1) Angular = False
And (1)(1) Curved = True
And (1)(1) Round = true
And (1)(1) Symmetrical = True
And (1)(1) Wide = True
Then (1)(1) Stable = True [0.9502];

If (1)(1) Continuous = False
And (1)(1) Curved = False
Then (1)(1) Stable = False [0.9949];

If (1)(1) Closed = False
And (1)(1) Continuous = True
And (1)(1) Curved = False
And (1)(1) Round = False
Then (1)(1) Stable = False [0.9391];

Règles aboutissant aux labels des chiffres manuscrits

If (1)(1) Closed = True
And (1)(1) Strong = False
Then (1)(1) Label = Zero [0.9999];

If (1)(1) Closed = False
And (1)(1) Complex = False
And (1)(1) Curved = False
And (1)(1) Weak = True
Then (1)(1) Label = One [0.9988];

If (1)(1) Closed = False
And (1)(1) Curved = True
And (1)(1) Symmetrical = False
And (1)(1) Weak = True
And (1)(1) Whistling = False
Then (1)(1) Label = Two [0.9888];

If (1)(1) Closed = False
And (1)(1) Curved = True
And (1)(1) Symmetrical = True
And (1)(1) Weak = True
And (1)(1) Whistling = False
Then (1)(1) Label = Three [0.9969];

If (1)(1) Closed = False
And (1)(1) Round = True
And (1)(1) Weak = False
And (1)(1) Whistling = True
Then (1)(1) Label = Six [0.9754];

If (1)(1) Closed = False
And (1)(1) Long = True
And (1)(1) Round = False
And (1)(1) Weak = False
Then (1)(1) Label = Seven [0.9999];

If (1)(1) Closed = True
And (1)(1) Strong = True
Then (1)(1) Label = Eight [0.9993];

If (1)(1) Closed = False
And (1)(1) Round = True
And (1)(1) Weak = False
And (1)(1) Whistling = False
Then (1)(1) Label = Nine [0.9888];

If (1)(1) Closed = False
And (1)(1) Complex = True
And (1)(1) Curved = False
And (1)(1) Weak = True
Then (1)(1) Label = Four [0.9985];

If (1)(1) Sacred = False
And (1)(1) Sensual = False
And (1)(1) Stable = True
And (1)(1) Weak = True
And (1)(1) Whistling = False
Then (1)(1) Label = Two [0.9687];

If (1)(1) Sacred = False
And (1)(1) Sensual = False
And (1)(1) Weak = False
And (1)(1) Whistling = True
Then (1)(1) Label = Six [0.9906];

If (1)(1) Sacred = True
And (1)(1) Sensual = False
And (1)(1) Weak = False
And (1)(1) Whistling = True
Then (1)(1) Label = Seven [0.9981];

If (1)(1) Harmonious = True
And (1)(1) Masculine = False
And (1)(1) Sensual = True
And (1)(1) Weak = False
And (1)(1) Whistling = False
Then (1)(1) Label = Eight [0.9985];

If (1)(1) Complex = False
And (1)(1) Masculine = True
And (1)(1) Sensual = False
And (1)(1) Stable = False
And (1)(1) Weak = True
Then (1)(1) Label = One [0.9745];

If (1)(1) Weak = True
And (1)(1) Whistling = True
Then (1)(1) Label = Five [0.9877];

Règles supplémentaires

If (1)(1) Closed = True
Then (1)(1) Even = True [1];

If (1)(1) Even = True
Then (1)(1) Uneven = False [1];

If (1)(1) Round = True
And (1)(1) Closed = False
Then (1)(1) Divisible.by.3 = True [1];

If (1)(1) Divisible.by.3 = True
And (1)(1) Round = True
Then (1)(1) Strong = True [1];

If (1)(1) Uneven = True
Then (1)(1) Even = False [1];

If (1)(1) Closed = True
Then (1)(1) Prime = False [1];

Références

- Akdag, H, & Khoukhi, F. (1994). Une approche logico-symbolique dans les systèmes experts, *ERGO-IA '94*, Bayonne.
- Bonnet, C., Botte, M. C., & Molnar, F. (1986). De l'automatique à l'intelligence artificielle en perception. In Bonnet, Hoc, & Tiberghien, *Psychologie, Intelligence artificielle et Automatique*, Bruxelles: Pierre Mardaga Editeur.
- Bouchon-Meunier, B. (1995). *La logique floue et ses applications*. Paris: Editions Addison-Wesley France.
- Damasio, A. R. (1990). Category-related recognition defects as clue to the neural substrates of knowledge. *Trends in Neuroscience*, 13(3), 95-98.
- Dreyfus, H. L. (1979) *What Computers Can't Do: A Critique of Artificial Reason*, New York, NY: Harper & Row. *Intelligence Artificielle, Mythes et Limites*, Traduit de l'américain par R.M. Vassallo-Villaneau et D. Andler. Paris: Editions Flammarion., 1984.
- Fodor, J. A. (1983). *The Modularity of Mind, An Essay on Faculty Psychology*. Cambridge, MA: The MIT Press. *La Modularité de l'Esprit. Essai sur la Psychologie des Facultés*. Traduit de l'américain par A. Gerschenfeld. Paris: Les Editions de Minuit, 1986.
- Fodor J. A., & Pylyshyn Z. W. (1988). Connectionism and Cognitive Architecture: a Critical Analysis, *Cognition*, 28:3-72.
- Haugeland, J. (1985). *Artificial Intelligence, the very Idea*. Cambridge, MA: The MIT Press. *L'Esprit dans la Machine. Les Fondements de l'Intelligence Artificielle*. Traduit de l'américain par J. Henry. Paris: Editions Odile Jacob, 1986.
- Johnson-Laird, P. N. (1983). *Mental Models: Towards a Cognitive Science of Language, Inference, and Consciousness*, Cambridge, MA: Harvard University Press.
- Kosslyn, S. M., & Koenig, O. (1992) *Wet Mind: the New Cognitive Neuroscience*. New York, NY: The Free Press.

- Khoukhi, F. (1996). *Approche logico-symbolique dans le traitement de connaissances incertaines et imprécises dans les systèmes à base de connaissances*. Thèse de doctorat, Université de Reims Champagne-Ardenne, Reims.
- Pylyshyn, Z. W. (1984). *Computation and Cognition. Toward a Foundation for Cognitive Science*, Cambridge, MA: The MIT Press.
- Pylyshyn, Z. W. (1989). Computing in Cognitive Science. In M. Posner (Ed.), *Foundations of Cognitive Science*. Cambridge, MA.: The MIT Press.
- Pylyshyn, Z. W. (1993). Computer and the Symbolization of Knowledge. In Morelli, Anselmi, Brown, Haberlandt, & Lloyd (Eds.) *Minds, Brains and Computers: Perspectives in Cognitive Science and Artificial Intelligence*. Ablex.
- Rabaséda, S. (1996). *Contributions à l'extraction automatique de connaissances : application à l'analyse clinique de la marche*. Thèse de doctorat, Université Claude Bernard, Lyon 1, Lyon.
- Rabaséda, S., Rakotomalala, R., & Zighed, D. A. (1996) Rules extracted automatically by induction. *Proceedings of IPMU'96*. Granada, 2:551-556.
- Rakotomalala, R., Zighed, D. A., & Rabaséda, S. (1996) Validation of rules issued from induction graph. *Proceedings of IPMU'96*. Granada, 3:1259-1264.
- Sachs, O. (1995). *An Anthropologist on Mars*. Alfred A. Knopf Editor. *Un Anthropologue sur Mars*. Traduit de l'anglais par C. Cler, Paris: Editions du Seuil, 1996.
- Sebban, M. (1996). *Modèles théoriques en reconnaissance de formes et architecture hybride pour machine perceptive*. Thèse de doctorat, Université Claude Bernard, Lyon 1, Lyon.
- Sebban, M., & Zighed, D. A. (1996). Discrimination of odors by semantic attributes and priming effects. *Proceedings of IPMU'96*. Granada, 2:693-697.
- Zighed, D. A., Auray, J. P., Duru, G. (1992). *SIPINA, Méthode et Logiciel*. Lyon: Editions Alexandre Lacassagne.
- Zighed, D. A., & Sebban, M. (1996). Cognitive model for the identification of odors. *Proceedings of IPMU'96*. Granada, 2:699-703.