



جامعة أبو بكر بلقايد - تلمسان

Université Abou Bakr Belkaid de Tlemcen

Faculté de Technologie
Département de Génie Biomédical
Laboratoire de Recherche de Génie Biomédical

MEMOIRE DE PROJET DE FIN D'ETUDES

pour obtenir le Diplôme de
MASTER en Génie Biomédical
Spécialité : Informatique Biomédicale

Présenté par : STAMBOULI Sarah et MOUSSAOUI Zahra

Thème

Conception et Réalisation
du Système Expert Médical INFUR

Soutenu le : 27/05/2015 devant le Jury

M^r BEHADADA Omar

M.A.A, Université de Tlemcen

Président

M^r MOUSSAOUI Djilali

M.C.B, Université de Tlemcen

Examineur

M^r BENZAOUZ Mourtada

M.C.B, Université de Tlemcen

Encadreur.

Année universitaire 2014/2015

Dédicaces

Je dédie ce modeste travail à :

Mes très chers parents

Que ce travail soit l'expression de ma reconnaissance pour vos sacrifices consentis, votre soutien moral et matériel que vous n'avez cessé de prodiguer, qui m'ont appris à aimer le travail et le bon comportement.

Vous avez tout fait pour ma réussite. Que dieu vous garde et vous protège.

A mon cher mari qui était toujours là pour moi âme et corps, pour son soutien et ses encouragements.

Mon grand frère Smail, sans oublier sa femme et ma soeur Meriem qui m'ont beaucoup soutenue.

Ma grand-mère qui est une grande dame et une grande âme.

Toute ma famille, exceptionnellement ma tante Amina, mon oncle amine et sa femme Hind, ils vont trouver ici l'expression de mes sentiments de respect et de reconnaissance pour le soutien qu'ils n'ont cessé de me porter.
à mes cousines.

à Mon binôme MOUSSAOUI Zahra.

Tout mes ami(e)s.

Tous ceux qui me sont chers.

Tous ceux qui m'aiment.

Tous ceux que j'aime.

Sans oublier mon grand père et ma grand-mère (Allah yerhamhoum)

Sachez que vous serez toujours dans mon cœur.

Sarah

Dédicaces

A l'homme de ma vie, mon exemple éternel, mon soutien moral et source de joie et de bonheur, celui qui s'est toujours sacrifié pour me voir réussir, que dieu te garde dans son vaste paradis, à toi mon père.

A la lumière de mes jours, la source de mes efforts, la flamme de mon coeur, ma vie et mon bonheur ; maman que j'adore.

Mon Fiancé a qui je tiens à exprimer ma profonde gratitude et tout mon respect pour son soutien, sa disponibilité et sa patience. Merci à ma belle famille en témoignage de ma reconnaissance envers leur soutien et leurs encouragements.

A mon binôme STAMBOULI Sarah.

Aux personnes dont j'ai bien aimé la présence dans ce jour, à tous mes frères Ahmed, Benamar et Youcef et mes belles soeurs Zahra, Safia et Nadjat, ma nièce Nahoula, et mes neveux Mohamed et abderrahman, je dédie ce travail dont le grand plaisir leurs revient en premier lieu pour leurs conseils, aides, et encouragements.

Aux personnes qui m'ont toujours aidé et encouragé, qui étaient toujours à mes côtés, et qui m'ont accompagnaient durant mon chemin d'études supérieures, mes aimables amies soussitta, noussitta, sousou, nina et khayrou , collègues d'étude.
à mes grands-mères, que dieu vous protège et vous gardes pour moi.

Tous ceux qui me sont chers (mes tantes, mes oncles....).

Tous ceux qui m'aiment.

Tous ceux que j'aime.

à toute la promotion GBM.

Sans oublier mes grands pères (Allah yerhamhoum)

Sachiez que vous serez toujours dans mon coeur.

Zahra

Remerciements

Ce travail n'aurait pu être réalisé sans le soutien du bon dieu. Merci Allah de nous avoir donné la capacité d'écrire et de réfléchir, la force d'y croire, la patience d'aller jusqu'au bout du rêve et la volonté et la patience de mener à terme le présent travail.

Au terme de ce projet Nous exprimons nos profondes gratitudee et respectueuse reconnaissance à notre encadreur, Mr BENAZZOUZ, maitre de conférences au département d'Informatique de la faculté des sciences à l'université de Tlemcen, pour l'assistance qu'il nous a prêté, son soutien et ses conseils avisés pendant toute la durée de ce travail.

Nous tenons à remercier Mr BEHADADA maitre Assistant au département de genie biomedical à la faculté de technologie, université de Tlemcen, qui a bien voulu nous faire l'honneur de présider le jury de notre soutenance.

Nous adressons également nos respectueux remerciements à Mme MOUSSAOUI Djilali pour son aide et sa disponibilité ainsi que son soutien durant la réalisation de ce travail en acceptant d'être examinatrice.

On remercie, aussi, à toute l'équipe du laboratoire STIC spécialement à Mr MERZOUGUI et Mr HAJILA pour le soutien moral et pour leur accueil bienveillant et leur conseils avisés.

Nous adressons nos remerciements, aussi, à Dr SETTOUTI et Dr BENAMMOUR des Gécicologues, et Dr SOUAD docteur en médecine pour leurs disponibilité et leurs aide, ainsi que leurs encouragement.

Notre Vif remerciement s'adresse également à nos enseignants et nos amis, pour leur présence chaleureuse et leur encouragement.

Enfin nous remercions tous ceux qui ont contribué de près ou de loin à la concrétisation de Ce travail, et à toute la promotion de master IBM.

Table des matières

Introduction Générale	10
I Chapitre 1 : L'Intelligence Artificielle	13
I.1 Introduction	13
I.2 Définitions	13
I.3 Génèse de l'I.A	14
I.3.1 Gestation de l'I.A (1943-1955)	14
I.3.2 Naissance d'I.A (1956)	14
I.3.3 Espoirs grandissants (1952-1969)	15
I.3.4 Premières Déceptions (1966-1973)	15
I.3.5 Systèmes Experts (1969-1979)	15
I.3.6 L'I.A dans l'Industrie (1980-présent)	15
I.3.7 L'I.A Moderne (1987-présent)	16
I.4 Les sous-domaines de l'I.A	16
I.4.1 Représentation des connaissances et Raisonnement Automatique	16
I.4.2 Résolution de problèmes généraux	16
I.4.3 Traitement du langage naturel	16
I.4.4 Vision artificielle	16
I.4.5 Robotique	16
I.4.6 Apprentissage automatique	16
I.5 Conclusion	17
II Chapitre 2 : Pathologie Urologique	18
II.1 Introduction	18
II.2 L'appareil urinaire	18
II.2.1 Le rein	18
II.2.2 Le néphron	19
II.2.3 L'uretère	19
II.2.4 La vessie	20
II.2.5 L'urètre	20
II.2.6 L'urine	20
II.3 Pathologie de l'appareil urinaire	20
II.4 Cystite et Néphrite	21
II.4.1 les causes	21
II.4.2 Les symptômes	22
II.4.3 Le diagnostic	22
II.4.4 Evolution de la maladie	23
II.4.5 Options thérapeutiques	23
II.5 Conseils hygiéniques	23
II.6 Conclusion	24
III Chapitre 3 : Le Système Expert	25
III.1 Introduction	25
III.2 Le Système expert	25
III.2.1 Composants d'un SE	26
III.3 Chainage avant :	27
III.3.1 Algorithme du chainage avant	28
III.4 Chainage arrière :	29
III.5 Chainage mixte :	30
III.5.1 Algorithme du chaînage mixte :	30
III.6 Réalisation d'un SE	31

III.6.1 Phase de conception	31
III.6.2 Phase d'implantation	31
III.7 Histoire de MYCIN	31
III.7.1 Qu'est-ce qu'un Mycin ?	32
III.7.2 Objectifs de Mycin	32
III.7.3 Architecture de Mycin	32
III.8 Avantages et Inconvénients :	33
III.8.1 Avantages d'un système expert :	33
III.8.2 Inconvénients d'un système expert :	34
III.9 Clips : L'outil de développement	34
III.10 Conclusion	35
IV Chapitre 4 : Logique Floue Et ANFIS	36
IV.1 Introduction	36
IV.2 Théorie des ensembles flous	36
IV.3 Le Concept d'ensemble flou	36
IV.4 Définition	37
IV.5 Caractérisation d'un ensemble flou	37
IV.6 Implication floue	39
IV.7 Structure interne d'un système d'inférence flou (SIF)	40
IV.7.1 Fuzzification	40
IV.7.2 L'inférence	41
IV.7.3 La défuzzification	41
IV.8 Exemple de système flou	42
IV.9 L'Anfis : Neuro-Flou	45
IV.10 L'apprentissage du classifieur Neuro-flou	46
IV.11 Conclusion	47
V Chapitre 5 : La Conception du Système Expert INFUR	48
V.1 Introduction	48
V.2 Description de la base de donnée	48
V.2.1 Renseignements sur les attributs	49
V.3 Conception de l'application	49
V.4 Générer les règles	49
V.4.1 1ère technique : La logique Floue	49
V.4.2 Les règles obtenues :	51
V.4.3 2ème technique : ANFIS	53
V.4.4 Structure de notre modèle	56
V.5 Les résultats obtenus	56
V.5.1 Evaluation des résultats :	57
V.6 La conception du système expert en CLIPS	57
V.7 L'intégration du système expert en JAVA	59
V.8 Conception du Système	60
Conclusion générale et Perspectives	63

Table des figures

1	Le système urinaire	18
2	Image tirée de : Le Corps Humain, L'Encyclopédie du corps humain en 3D,	19
3	Cystite	22
4	Néphrite	22
5	Composants d'un SE	26
6	Chainage avant	27
7	Exemple de chainage avant	28
8	chainage arrière	29
9	Exemple chainage arrière	30
10	Architecture de Mycin	33
11	Ensemble flou trapézoïdal	38
12	Tangente hyperbolique	39
13	Structure d'un système flou	40
14	Fonctionnement d'un système Flou	40
15	Familles de contrôleurs flous	41
16	Connaissance humaine	42
17	Fonction d'appartenance de l'abscisse d'un point	43
18	Exemple d'interpolation non linéaire	44
19	Architecture d'ANFIS	45
20	Charger la base de donnée	50
21	générer un SIF sans règles	50
22	Génération d'un SIF sans règles :	51
23	les degrés d'appartenance de « Température »	51
24	affecter 1 pour « oui »	52
25	générer les règles	52
26	Génération de conclusion	53
27	Base de cystite	53
28	Base de néphrite	53
29	Lancement d'apprentissage	54
30	les fonctions d'appartenance de « Température »	55
31	Génération de la sortie	55
32	Neuro-Flou	56
33	FIS de la cystite	56
34	FIS de la Néphrite	57
35	lecture de la base des règles	58
36	le Questionnaire	58
37	décision1 : Vous avez la cystite et la néphrite	59
38	décision2 : vous n'avez ni la cystite ni la néphrite	59
39	décision3 : vous n'avez pas la cystite mais vous avez la néphrite	59
40	décision4 : vous avez la cystite mais vous n'avez pas la néphrite	60
41	lire la base de connaissance	60
42	Répondre aux questions pour avoir la décision	60
43	Diagramme de cas d'utilisation	61
44	1ère interface d'INFUR	61
45	accueil	62
46	Gestion des patients	62
47	L'interface de diagnostiquer	62

Liste des tableaux

1	Attributs de la base de données	49
2	Décisions de la base de données	49
3	Erreur pour différentes configurations	54
4	Différence entre les deux méthodes	54
5	Différence entre les deux méthodes	55
6	Resultat de TEST de la base de cystite	57
7	Résultat de TEST de la base de néphrite	57
8	Resultat de TEST de la base de 40 de cystite	57
9	Résultat de TEST de la base de 40 de néphrite	58

Glossaire :

A

ANFIS : Adaptive Neuro-Fuzzy Inference System.

B

BDC : Base De Connaissances.

BDF : Base De Faits.

BDR : Base Des Règles.

F

FC : Fuzzy Controller.

FN : Faux Negatif.

FP : Faux Positif

I

I.A : Intelligence Artificielle.

INFUR : notre SE des Inflammations Urinaires.

M , P

MI : Moteur d'Inférence

PROLOG : Programmation Logique.

Q

QI : Quotient Intellectuel.

S

SBC : Système à Base de Connaissance.

SE : Système Expert.

Se : Sensibilité.

SIF : Système d'Inférence Flou.

Sp : Spécificité.

T

TALN : Traitement Automatique de Langage Naturel.

TSK : Takagy-Sugeno-Kang.

V

VN : Vrai Negatif.

VP : Vrai Positif.

Introduction Générale

Contexte :

De nos jours, les techniques de l'I.A sont devenues de plus en plus utilisées dans le développement des systèmes informatiques ; pour construire des applications dans de nombreux domaines dont la médecine, l'éducation etc...

D'après les chercheurs, la plupart des décisions médicales sont basées sur l'expérimentation et plus particulièrement le cas par cas. Il semble de plus en plus évident que les décisions complexes pourraient être mieux prises grâce à la modélisation qui repose sur de grandes bases de données plutôt que sur la seule intuition.

Objectif :

Des systèmes basés sur l'I.A pourraient énormément améliorer à la fois le coût, et la qualité des soins de santé selon une étude publiée dans la revue *Artificial Intelligence in Medicine*. Parmi les problèmes connus dans la médecine, les infections urinaires sont plus fréquentes chez les femmes mais plus graves chez les hommes.

En France, près de deux millions de femmes sont chaque année victimes d'infections urinaires récidivantes. Les bactéries qui viennent de l'extérieur, ont un trajet plus court à parcourir chez la femme que chez l'homme. Il y en a plusieurs cas pour traiter cette maladie, et des fois les cas se ressemblent.

Contribution :

Pour cela, on a éprouvé le besoin de faire la conception d'un SE, cela serait idéal pour aider le médecin non seulement à gagner du temps pour conclure un cas spécifique mais aussi en proposant de nouvelles règles qui peuvent améliorer sa vision des cas de traitement.

Notre travail consiste à réaliser un SE pour le diagnostic des inflammations urinaires destiné aux médecins spécialistes en urologie.

Ce SE devra permettre au médecin de :

1. Diagnostiquer le type d'inflammation urinaire rapidement.
2. Fournir un apprentissage automatique pour les nouveaux cas.

Ce mémoire contient 5 chapitres, il est organisé comme suite :

- Le premier chapitre débute par définir l'I.A et son impact dans le domaine médicale.
- Le second chapitre brosse une aperçue sur la maladie des infections urinaires. Il décrit brièvement la maladie, les symptômes, les causes, le traitement, le diagnostic ainsi l'évolution de la maladie.

- Le troisième chapitre concerne l'intérêt du SE dans le domaine médicale, nous détaillons par la suite un des prototypes de ce domaine qui est Mycin en exposant ses principes, ses objectifs ainsi que son architecture.
Nous concluons ce chapitre par la définition de CLIPS (l'outil de système de développement et expert de livraison productive).
- Le quatrième chapitre consiste à exposer la technique de la logique floue ainsi que les outils software utilisés.
- Nous terminons ce manuscrit par le cinquième chapitre qui décrit la conception de notre SE INFUR, le résultat obtenu ainsi que la discussion et l'interprétation des conclusions conjointement avec un medecin.
- La dernière partie renferme la conclusion et les perspectives.

Etat De l'Art :

Les principes d'un SE ont subi une grande évolution depuis leurs créations. Ainsi, on a commencé à voir le développement d'un nouveau type de SE capable de s'exprimer en langage naturel et aussi d'expliquer leur raisonnement lors de la réponse à une question utilisateur. Ce type de SE, dit de 2ème génération, est aussi capable de détecter les contradictions et de s'adapter au changement d'avis, et sont dotés de capacités d'apprentissage. Plusieurs sciences et domaines ont bénéficié des capacités des SE depuis la fin des années 60. Nous présentons ci-après une liste non exhaustive de quelques SE les plus connus dans différents domaines :

- En commençant par le premier SE réalisé appelé DENDRAL, il est orienté au chimistes, inventé en 1969 a l'Université de Stanford en U.S.A. Il permet de reconstituer la formule développée d'un composé chimique à partir de sa formule brute et son spectrogramme de masse.
- MYCIN, le plus connu des SE, il a été réalisé entre 1972-1974 dans le domaine de la médecine a l'Université de Stanford en U.S.A. C'est un système d'aide à diagnostiquer et soigner les maladies infectieuses du sang et des méningites. Sa réalisation a fait intervenir des médecins experts en pharmacologie clinique des infections bactériennes, et des informaticiens experts en I.A et informatique médicale. Le résultat est un SE à base de 200 règles initialement, ensuite 500 règles vers la fin de l'année 1978. C'est un système très performant utilisant une communication quasi-naturelle. Il est basé sur la logique floue, probabiliste , pour la représentation de ses règles. voici un exemple : « le site de la culture est le sang, et l'organisme est à Gram négatif, et l'organisme est de forme bâtonnet, et le patient est un hôte à risque alors il est probable (0,6) que l'organisme soit le pseudomonias aeruginosa. »
- Essential MYCIN ou Engine MYCIN ou aussi Empty MYCIN surnommé EMYCIN, est le moteur extrait de MYCIN, sans règles qui a été appliqué à plusieurs applications et a servi à la création d'autres SE, il a été réalisé en 1974 à l'université de Stanford en U.S.A.
- Moteur d'inférence d'un SE appelé OPS, indépendant de toute application. Utilisé pour l'écriture de la connaissance sous forme de règles de production et permettant de générer de nombreux SE, il a été inventé en 1977 à l'université de Carnegie Mellon en U.S.A.
- PROSPECTOR, c'est un système d'aide à évaluer la possibilité d'existence d'un gisement de minerai dans un site étudié en Géologie. Il a été réalisé en 1978 à l'université de Stanford en U.S.A. Il compose des observations sur les caractéristiques géologiques du terrain considéré et la nature des minéraux trouvés en surface pour prédire l'existence de gisement. Le système utilise la logique floue ou probabiliste avec un raisonnement de type bayésien et comporte 1600 règles.
- Un SE développé pour le diagnostic des pannes d'ordinateurs et de systèmes. Il comporte 190 règles. il est surnommé DART destiné aux informaticiens, il est inventé en 1981 à l'IBM en U.S.A

- LPS, SE de résolution des problèmes de géométrie en mathématiques, il a été réalisé en 1979 à l'Université de Keio en Japon.
- SOPHIE, c'est un Système d'aide à l'enseignement des techniques de détection des pannes dans un circuit électronique dans le domaine de l'électricité, il a été réalisé en 1975 à l'Université de Stanford en U.S.A.
- un Système d'étude des dossiers de demande de crédits s'appelle PEGASE destiné aux Gestionnaires, il a été réalisé en 1984 à l'école Polytechnique de lausanne.
- Un Système à 100 règles CAMA pour le diagnostic de pannes d'automobiles et le dépannage automatique. Il a été réalisé en 1982 pour les automobilistes à Politecnio de Milan en Italie.
- PDS, un Système de diagnostic en temps réel d'incident d'exécution de processus automatisés surveillés par ordinateur dans le domaine Contrôle de Processus. Il a été réalisé en 1983 à Westinghouse de l'U.S.A.
- Et finalement un système d'interprétation de sonagrammes pour la reconnaissance de la parole surnommé SONEX dans le domaine Traitement de la parole, il a été inventé en 1984 au Laboratoire LIMSI à l'Université de Paris 11 en France.

I Chapitre 1 : L'Intelligence Artificielle

I.1 Introduction

Depuis les années 1950, deux grandes approches ont été adoptées par les chercheurs pour concevoir des machines « intelligentes ».

Une approche (Making a mind), que l'on peut qualifier d'I.A symbolique, consiste à doter un système d'I.A des mécanismes de raisonnement capables de manipuler les données symboliques qui constituent les connaissances d'un domaine. Cette approche fait appel aux modèles et méthodes de la logique. Elle a donné lieu aux systèmes à bases de connaissances.

Une autre approche (Modeling the brain), que l'on peut qualifier d'I.A connexionniste, consiste à s'inspirer du fonctionnement du cortex cérébral. L'entité de base est un modèle formel du neurone, un système étant formé par l'interconnexion d'un grand nombre de tels « neurones ». Cette approche a donné lieu aux réseaux neuromimétiques actuels. Depuis les années 1990, une tendance prometteuse est de concevoir des modèles hybrides combinant ces deux approches qui présentent des caractères complémentaires. Par ailleurs, des modèles statistiques sont de plus en plus mis à profit pour rendre compte de la grande variabilité des phénomènes étudiés. Ces trois grandes approches de l'I.A (symbolique, connexionniste et statistique) sont décrites dans la suite de ce chapitre.(1)

I.2 Définitions

Plusieurs définitions ont été données à l'I.A par différents chercheurs, on citera quelques unes :

- « L'étude des facultés mentales à l'aide des modèles de type calculatoires. »(Charniak et McDermott, 1985)
- « Conception d'agents intelligents. »(Poole et al., 1998)
- « Discipline étudiant la possibilité de faire exécuter par l'ordinateur des tâches pour lesquelles l'homme est aujourd'hui meilleur que la machine. »(Rich et Knight, 1990)
- « L'automatisation des activités associées au raisonnement humain, telles que la décision, la résolution de problèmes, l'apprentissage,... »(Bellman, 1978)
- « L'étude des mécanismes permettant à un agent de percevoir, raisonner, et agir. »(Winston, 1992)
- « L'études des entités ayant un comportement intelligent. »(Nilsson,1998)

Comme vous pouvez remarquer, ces définitions s'accordent sur le fait que l'objectif de l'I.A est de créer des systèmes intelligents, mais elles diffèrent significativement dans leur façon de définir l'intelligence. Certaines se focalisent sur le comportement du système, tandis que d'autres considèrent que c'est le fonctionnement interne (le raisonnement) du système qui importe. Une deuxième distinction peut être faite entre celles qui définissent l'intelligence à partir de l'être humain et celles qui ne font pas référence aux humains mais à un standard de rationalité plus général. On peut donc décliner quatre façons de voir l'I.A :

- Créer des systèmes qui se comportent comme les êtres humains, cette définition opérationnelle de l'I.A fut promue par Alan Turing, qui introduisit son fameux 'test de Turing' selon lequel une machine est considérée comme intelligente si elle peut converser de telle manière que les

interrogateurs (humains) ne peuvent la distinguer d'un être humain .

- Créer des systèmes qui pensent comme des êtres humains, si l'on adhère à cette deuxième définition, cela implique que l'I.A est une science expérimentale, car il faut comprendre au préalable la façon dont pensent les humains (sinon, comment savoir si une machine pense comme un homme ?) et ensuite évaluer les systèmes par rapport à leurs similarités avec le raisonnement humain.
- Créer des systèmes qui pensent rationnellement, selon cette définition, les systèmes doivent raisonner d'une manière rationnelle, c'est à dire en suivant les lois de la logique. Cette approche peut être critiquée car il semble que certaines capacités (la perception, par exemple) ne sont pas facilement exprimables en logique. De plus, ce standard de rationalité ne peut pas être atteint en pratique car la technologie actuelle ne permet pas de réaliser des calculs aussi complexes.
- Créer des systèmes qui possèdent des comportements rationnels, cette dernière définition de l'I.A concerne le développement des agents qui agissent pour mieux satisfaire leurs objectifs. On remarque que cette définition est plus générale que la précédente car raisonner logiquement peut être une façon d'agir rationnellement mais n'est pas la seule (par exemple, le réflexe de retirer sa main d'un objet brûlant est rationnel mais n'est pas le résultat d'une inférence logique). Ajoutons que dans la réalité, ces distinctions n'ont pas forcément une influence aussi grande que l'on pourrait imaginer sur la façon dont la recherche en I.A est menée. Les chercheurs n'ont pas tous un avis très précis sur ce que doit être l'objectif ultime de l'I.A, mais trouvent tout simplement que les questions soulevées par ce domaine sont intéressantes et méritent d'être étudiées.(2)

I.3 Génèse de l'I.A

Nous citons ci-dessous quelques étapes importantes dans l'histoire de l'I.A :

I.3.1 Gestation de l'I.A (1943-1955)

Pendant cette période furent menés les premiers travaux qui peuvent être considérés comme les débuts de l'intelligence artificielle (même si le terme n'existait pas encore). On peut citer les travaux de McCulloch et Pitts qui ont introduit en 1943 un modèle de neurones artificiels. Quelques années après, Hebb proposa une règle pour modifier des connections entre neurones, et Minsky et Edmonds construisirent le premier réseau de neurones. Ce fut aussi durant cette période que Turing publia son fameux article dans lequel introduit le test de Turing.

I.3.2 Naissance d'I.A (1956)

On considère que l'I.A en tant que domaine de recherche, a été créée à la conférence qui s'est tenue sur le campus de Dartmouth College Canada pendant l'été 1956 à laquelle assistaient ceux qui vont marquer la discipline, le mathématicien John McCarthy organise le premier séminaire sur l'I.A. Son confrère Marvin Minsky et Claude E. Shannon, le père de la théorie de l'information, sont présents. Il y a aussi Herbert A. Simon, spécialiste des organisations, et son ami mathématicien Alan Newell. Tous deux créent la surprise en présentant le premier programme d'I.A : Logic Theorist. Ce programme informatique est destiné à démontrer des théorèmes mathématiques. Il fonctionne comme une machine logique capable d'enchaîner et d'articuler entre elles une foule de propositions à partir de quelques prémisses (sur le modèle du syllogisme « si A implique B » et « B implique C » alors « A implique C »).

I.3.3 Espoirs grandissants (1952-1969)

Ce fut une période très active pour le jeune domaine de l'I.A. Un grand nombre de programmes furent développés pour résoudre des problèmes d'une grande diversité. Les programmes Logic Theorist (par Newell et Simon) et Geometry Theorem Prover (Gelernter) furent en mesure de prouver certains théorèmes mathématiques (tous déjà connus, mais en trouvant parfois une preuve plus élégante). Le *General Problem Solver* de Newell et Simon réussissait quant à lui à résoudre des puzzles simples avec un raisonnement semblable au raisonnement humain. Samuel créa un programme jouant (à un niveau moyen) aux dames. Des étudiants de Minsky travaillèrent sur les petits problèmes (« microworlds ») tels que les problèmes d'analogie (problèmes du même type que ceux des tests de QI), donnant naissance au programme ANALOGY, ou encore les manipulations de cubes (le fameux « blocks world ») avec l'idée d'augmenter la complexité petit à petit pour développer des agents intelligents. McCarthy publia un article devenu célèbre dans lequel il traite des programmes qui ont du sens commun. La recherche sur les réseaux de neurones fut également poursuivie. Ce fut aussi l'époque du Shakey, le premier robot à être capable de raisonner sur ses propres actions.

I.3.4 Premières Déceptions (1966-1973)

Il devint durant ces années de plus en plus évident que les prédictions faites par les chercheurs en I.A avaient été beaucoup trop optimistes. Ce fut le cas par exemple pour la traduction automatique. Les chercheurs n'avaient compté que 5 ans pour réaliser un traducteur automatique, mais se sont vite rendu compte que leur approche purement syntaxique n'étaient pas suffisante (pour bien traduire un texte, il faut d'abord le comprendre). Cet échec a provoqué l'annulation en 1966 de tout le financement du gouvernement américain pour les projets de traduction automatique. De grandes déceptions se produisirent également lorsque les chercheurs en I.A essayèrent d'appliquer leurs algorithmes aux problèmes de grande taille, et découvrirent alors qu'ils ne fonctionnaient pas, par manque de mémoire et de puissance de calcul. Ce fut une des critiques adressée à l'I.A dans le rapport de Lighthill de 1973, qui provoqua l'arrêt du financement de la quasi-totalité des projets en I.A de Grande Bretagne. Et comme si cela ne suffisait pas, Minsky et Papert prouvèrent dans leur livre « Perceptrons » de 1969 que les réseaux de neurones de l'époque ne pouvaient pas calculer certaines fonctions pourtant très simples, ce qui mit en cause toute la recherche en apprentissage automatique, entraînant une crise dans cette branche de l'I.A.

I.3.5 Systèmes Experts (1969-1979)

Le premier SE, appelé DENDRAL, fut créé en 1969 pour la tâche spécialisée consistant à déterminer la structure moléculaire d'une molécule étant donné sa formule et les résultats de sa spectrométrie de masse. DENDRAL, comme toutes les systèmes experts, est basé sur un grand nombre de règles heuristiques (nous reviendrons sur ce terme en détail dans la suite du cours) élaborées par des experts humains. Après le succès du DENDRAL, d'autres systèmes d'experts furent créés, notamment le système MYCIN, qui réalisait un diagnostic des infections sanguines. Avec 450 règles, MYCIN réussissait à diagnostiquer à un niveau proches des experts humains et considérablement meilleur que celui les jeunes médecins.

I.3.6 L'I.A dans l'Industrie (1980-présent)

Au début des années 80, l'entreprise DEC commença à utiliser un système expert d'aide à la configuration de systèmes informatiques, ce qui leur permit d'économiser des dizaines de millions de dollars chaque année. Beaucoup de grandes entreprises commencèrent alors à s'intéresser à l'I.A et à former leur propres équipes de recherche. Les Etats-Unis et le Japon financèrent de gros projets en

I.A, et la Grande Bretagne relança son programme de financement.

I.3.7 L'I.A Moderne (1987-présent)

L'intelligence artificielle est devenue au fil du temps une matière scientifique de plus en plus rigoureuse et formelle. La plupart des approches étudiées aujourd'hui sont basées sur des théories mathématiques ou des études expérimentales plutôt que sur l'intuition, et sont appliquées plus souvent aux problèmes issus du monde réel.(2)

I.4 Les sous-domaines de l'I.A

On s'en serait douté, créer des agents intelligents n'est pas si simple. Pour cette raison, l'I.A s'est divisée en de nombreuses sous-disciplines qui essaient chacune de traiter une partie du problème. Voici les principales :

I.4.1 Représentation des connaissances et Raisonnement Automatique

Comme son nom le suggère, cette branche de l'I.A traite le problème de la représentation des connaissances (qui peuvent être incomplètes, incertaines, ou incohérentes) et de la mise en oeuvre du raisonnement.

I.4.2 Résolution de problèmes généraux

L'objectif est de créer des algorithmes généraux pour résoudre des problèmes concrets.

I.4.3 Traitement du langage naturel

Ce sous-domaine vise à la compréhension, la traduction, ou la production du langage (écrit ou parlé).

I.4.4 Vision artificielle

Le but de cette discipline est de permettre aux ordinateurs de comprendre les images et la vidéo (par exemple, de reconnaître des visages ou des chiffres).

I.4.5 Robotique

Cette discipline vise à réaliser des agents physiques qui peuvent agir dans le monde (pour voir les robots humanoïdes les plus avancés aujourd'hui, allez sur le site (3)).

I.4.6 Apprentissage automatique

Dans cette branche de l'I.A, on essaie de concevoir des programmes qui peuvent s'auto-modifier en fonction de leur expérience. Il existe bien entendu des liens très forts entre ces sous-domaines. Par exemple, les langages développés dans la représentation des connaissances peuvent servir de base à des systèmes experts. Ou encore, beaucoup d'algorithmes pour la reconnaissance des formes sont

developpés en utilisant des méthodes d'apprentissage.

Il y a aussi de forts liens entre l'I.A et d'autres domaines tels que la philosophie, la psychologie, les neurosciences, les sciences cognitives, la linguistique, et l'économie.(2)

I.5 Conclusion

Après ce qu'on vient de voir, l'I.A est utile dans plusieurs domaines tels que :

- reconnaissance de formes ;
- TALN ;
- traduction automatique ;
- interfaces intelligentes ;
- apprentissage automatique, fouille de données, fouille de textes ;
- diagnostic, systèmes experts, SBC (médical, bourse, panne, etc.) ;
- jeux (échecs, dames, Go) : échec des algos classiques ;
- ordonnancement, satisfaction de contraintes ;
- optimisation de parcours ou de charge -> NP ;

II Chapitre 2 : Pathologie Urologique

II.1 Introduction

L'urologie est une spécialité médico chirurgicale. Les pathologies liées aux organes urinaires exigent toujours une approche médicale et parfois chirurgicale.

Elle est la spécialité médicale qui traite les voies urinaires des deux sexes, l'appareil génital masculin et traite les maladies et dysfonctionnements qui les concernent, tels que les calculs urinaires, l'incontinence, ou les malformations...

II.2 L'appareil urinaire

Le système urinaire comprend des organes (les reins), différentes structures (la vessie, l'urètre, l'uretère) et de nombreux vaisseaux sanguins permettant d'éliminer les déchets azotés produit par le métabolisme cellulaire. Lors de l'utilisation de molécules, comme les protéines, par les cellules, ces dernières rejettent de l'azote, une substance toxique pour le corps si elle est très concentrée. On se doit donc de l'éliminer, sous forme d'urée. L'urée voyage dans le système circulatoire jusqu'au rein, où le sang est filtré. L'urée ainsi qu'un peu d'eau se retrouve dans le rein lui-même, puis descend l'uretère jusqu'à la vessie, où l'urine est stockée. Lorsqu'accumulée en grande quantité, l'urine descend l'urètre vers l'extérieur du corps. Les glandes surrénales, situées juste au dessus des reins, ne font pas directement partie du système urinaire, bien qu'elles ont un effet indirect sur lui, comme sur le reste du corps. (4)

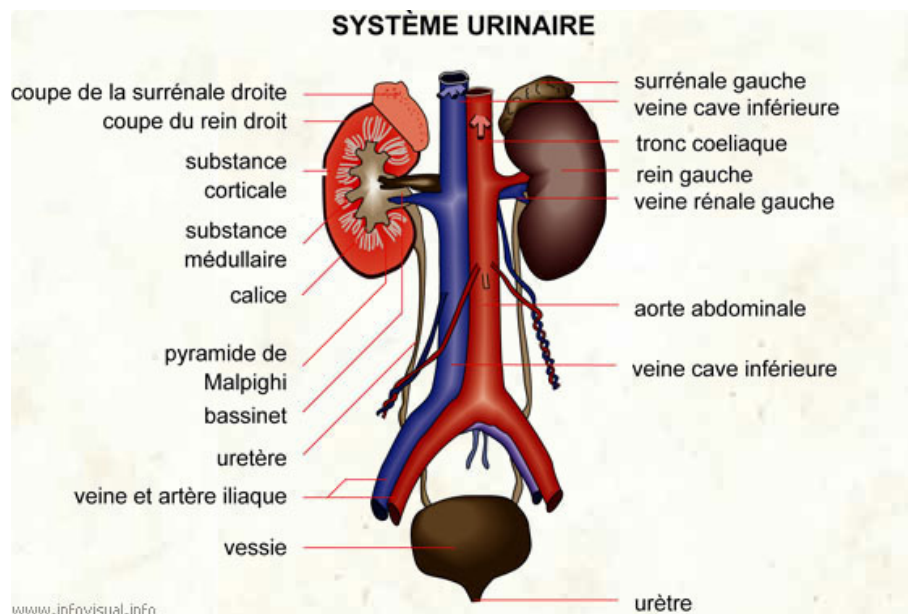


Figure 1 – Le système urinaire
(4)

II.2.1 Le rein

Les reins sont deux organes en forme de haricot situés dans la partie postérieure de l'abdomen, de part et d'autre de la colonne vertébrale, approximativement entre la douzième vertèbre dorsale et

la troisième vertèbre lombaire. Il arrive souvent que le rein gauche soit situé jusqu'à 2,5 centimètres plus haut que le rein droit. Le rein mesure 10 à 12,5 centimètres de long et 5 centimètres d'épaisseur environ, pour un poids de 100 à 150 grammes en moyenne chez l'adulte. Les reins sont de couleur rouge-brun foncé en raison de l'abondante présence de vaisseaux sanguins.

Le rein comporte un bord latéral concave sur lequel s'implantent l'artère rénale, la veine rénale, des fibres nerveuses, et l'uretère au niveau d'une ouverture appelée le hile. Le rein contient une cavité, le sinus rénal, qui renferme les unités fonctionnelles de la filtration, les néphrons. Chaque rein en contient plus d'un million. (4)

II.2.2 Le néphron

Le sang passe par les capillaires glomérulaires, vaisseaux sanguins au début du néphrons. À cet endroit, tout ce qui peut passer à travers la paroi des capillaires se retrouve dans le néphrons, soit l'eau, l'urée, les sels minéraux et le glucose; c'est la filtration. Par la suite, cette « urine » passe dans le tube contourné proximal, l'anse et le tube contourné distal. À ces endroits, « l'urine » subit une réabsorption, c'est-à-dire que l'eau, le glucose et les sels minéraux retournent dans le sang par les capillaires péri-tubulaires. Il y a aussi sécrétion, c'est-à-dire que le sang se débarrasse d'autres déchets, comme les ions acides H^+ , et les envoie dans l'urine. Le liquide se rend ensuite dans le tube collecteur, puis vers le calice et le bassinnet du rein où il est emmagasiné.(4)

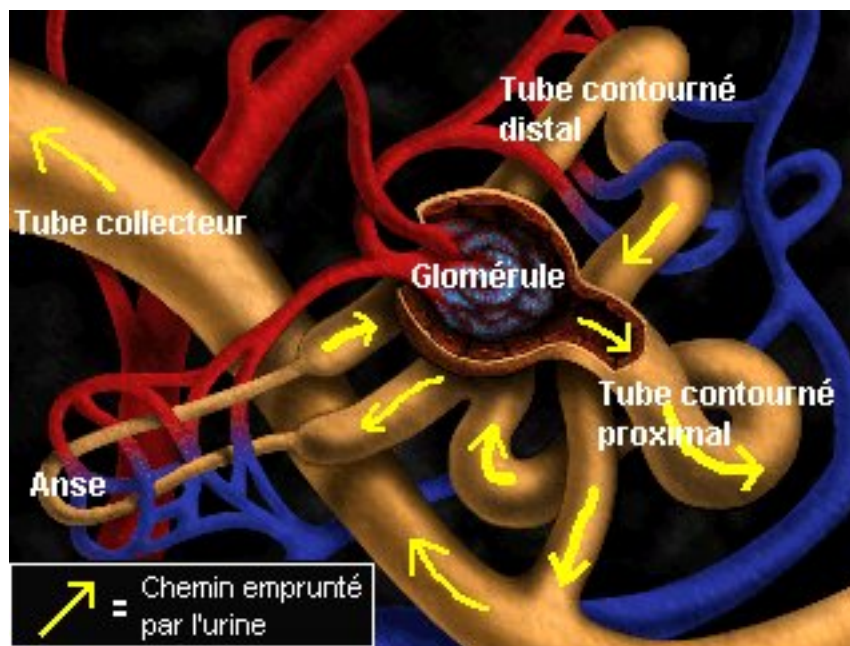


Figure 2 – Image tirée de : Le Corps Humain, L'Encyclopédie du corps humain en 3D,
(4)

Pour résumer brièvement, le sang qui circule dans le rein libère les déchets solubles dans les néphrons, et le liquide qui en résulte, l'urine, passe ensuite dans l'uretère pour séjourner dans la vessie en attendant d'être évacué lors de la miction.

II.2.3 L'uretère

Les uretères sont des tubes étroits mesurant de 35 à 45 centimètres de long et 3 millimètres de diamètre environ. Elles prennent naissance dans le bassinnet du rein, quittent le rein par le hile pour descendre le long de la paroi postérieure de la cavité abdominale et aboutir dans la vessie. Les uretères recueillent l'urine produite par les reins pour la conduire dans la vessie, où elle est stockée

jusqu'à la miction. L'urine ne s'écoule pas des reins en un flux ininterrompu, mais est chassée dans les uretères toutes les 20 à 30 secondes environ grâce à des ondes péristaltiques (contractions périodiques) commençant dans le bassinet. Ces ondes péristaltiques commandent l'ouverture de l'orifice urétéral (qui met l'uretère en communication avec la vessie), qui reste ouvert pendant quelques secondes pour laisser passer l'urine, puis se referme jusqu'à la prochaine onde.(4)

II.2.4 La vessie

La vessie est une structure en forme de sac composée de fibres musculo-membraneuses. Située dans le bassin, la vessie accumule l'urine jusqu'à son excrétion. L'urine arrive dans la vessie par les uretères, qui prennent naissance dans les reins, grâce à des mouvements péristaltiques. Durant la miction, l'orifice urétral situé à la base de la vessie s'ouvre et l'urine s'écoule dans l'urètre. Bien que l'envie d'uriner survienne généralement lorsque la vessie contient environ 250 à 300 millilitres d'urine (un peu plus d'une tasse liquide), la vessie humaine a une capacité moyenne égale à près de deux fois ce volume. L'être humain élimine en moyenne de un à deux litres d'urine par jour, bien que ce chiffre varie fortement selon l'état de santé, le régime alimentaire et les activités de l'individu.(4)

II.2.5 L'urètre

La vessie est mise en communication avec l'extérieur par un canal membraneux, l'urètre. L'urètre permet de transporter l'urine de la vessie jusqu'au méat à l'extrémité du pénis chez l'homme, ou jusqu'à un orifice allongé situé au milieu de la vulve, chez la femme. Chez l'homme, le sperme, mélange de spermatozoïdes provenant des testicules et de sécrétions provenant des vésicules séminales, emprunte également ce canal lors de l'éjaculation.(4)

II.2.6 L'urine

D'ordinaire, l'urine est claire ou jaune, quoique cela dépende du régime et de la santé du sujet. Par exemple, la prise de diurétique (certains aliments comme l'alcool, ou médicaments) augmente le volume d'urine produite en augmentant la quantité d'eau éliminée, diminue la concentration de l'urée et par le fait même, rend l'urine plus pâle. L'urine dégage une odeur d'ammoniaque caractéristique qui est due principalement aux déchets azotés formant 5 pour cent de son volume. Les déchets sont principalement constitués d'urée, bien que l'on trouve également dans l'urine de l'ammoniaque, de l'acide urique, de la créatinine, et de nombreux autres déchets. L'eau, élément principal de l'urine, est généralement éliminée dans les quatre heures qui suivent sa consommation.(4)

II.3 Pathologie de l'appareil urinaire

Parmi la grande variété des pathologies de l'appareil urinaire, liée à la diversité des organes et de leur fonction, on peut distinguer 4 groupes pathologiques principaux :

- La pathologie tumorale bénigne (adénome prostate, kyste rein,..) ou maligne (cancer rein, vessie, prostate, testicule).
- La pathologie infectieuse (cystite, prostatite, pyélonéphrite, parasitose).
- La pathologie fonctionnelle (incontinence, troubles sexuels, douleurs chroniques).
- La pathologie lithiasique (colique néphrétique, lithiase rénale).

Mais l'urologue est aussi confronté au traitement des malformations de l'appareil urinaire et génital masculin (phimosis, ectopie, malformation rénale, urétérale) ou traumatique.

II.4 Cystite et Néphrite

Du point de vue localisation, Il existe 2 types d'infection urinaire : l'infection urinaire haute (touchant le rein) et l'infection urinaire basse, appelée également cystite (inflammation de la vessie).

L'idée principale de cet ensemble de données est de préparer l'algorithme du SE, qui effectuera le diagnostic présomptif de deux maladies de l'appareil urinaire. Ce sera l'exemple de diagnostic des inflammations aiguës de la vessie et aiguë de rein. Pour une meilleure compréhension du problème considérons définitions de deux maladies données par des médecins.

L'infection des voies urinaires, également appelée inflammation de la vessie ou cystite, fait référence à une inflammation des tissus qui tapissent les voies urinaires. Le plus souvent, les bactéries sont en cause. Parmi les voies urinaires, les uretères sont les conduits urinaires qui transportent l'urine des bassins rénaux vers la vessie ; l'urètre quant à lui, est le canal de sortie de la vessie, qui amène l'urine à l'extérieur. Lorsque les bactéries se multiplient de façon abondante au sein de ce conduit, une inflammation se produit, ce qui se manifeste par des brûlures lors de la miction .

Chez la femme, les infections des voies urinaires et de la vessie sont beaucoup plus fréquentes parce que les voies urinaires de la femme sont plus courtes. Pour cette raison, les bactéries accèdent plus rapidement à la vessie, particulièrement chez les femmes jeunes sexuellement actives.

En raison de l'hypertrophie de la prostate, les hommes âgés sont plus souvent atteints d'infections urinaires (cystite) liées à l'obstacle lors de l'écoulement de l'urine.(5)

On distingue deux formes d'infections des voies urinaires : l'infection des voies urinaires basses et l'infection des voies urinaires hautes. Les infections des voies urinaires basses concernent l'urètre et la vessie ; les infections de voies urinaires hautes concernent le bassinet rénal.

La néphrite correspond à une inflammation du rein qui peut être d'origine infectieuse ou auto-immune, c'est-à-dire provoquée par une activité anormale du système immunitaire. Le rein joue un rôle essentiel en procédant à la destruction d'un certain type de déchets toxiques. Mais lorsque le rein est enflammé, il laisse passer un nombre trop important de toxines qui se retrouvent alors dans le sang en quantité anormale.

II.4.1 les causes

Différents facteurs favorisent la multiplication d'agents pathogènes au sein des voies urinaires :

- Sexe féminin.
- Grossesse et accouchement.
- Déficience en oxygène au cours de la ménopause.
- Chez l'homme, hypertrophie de la prostate.
- Calculs rénaux présentant un obstacle à l'écoulement de l'urine.
- Rétrécissement de l'urètre.
- Diabète de type 1 et de type 2 et goutte.
- Sonde urinaire.
- Mauvaise hygiène (nettoyage des régions génitales en partant de l'anus vers le vagin, ce qui favorise la pénétration des agents pathogènes en provenance du gros intestin).
- Faiblesse immunitaire chez les nourrissons, les enfants, les malades chroniques et les patients traités par certains médicaments comme la cortisone.
- Propagation par rapports sexuels intenses.

II.4.2 Les symptômes

- Douleurs et brûlures à la miction.
- Difficultés à la miction (dysurie).
- Miction impérieuse permanente avec émission de petites quantités d'urine (pollakiurie).
- Urines sombres, éventuellement mêlées à du sang.
- Douleurs et crampes dans le bas ventre.



Figure 3 – Cystite
(6)

Tout symptôme de cystite nécessite une consultation médicale comme l'infection peut s'étendre aux voies urinaires supérieures. Lorsque les symptômes apparaissent, il s'agit la plupart du temps déjà d'une infection rénale (néphrite) :

- Fièvre élevée, frissons.
- Douleurs dans le dos (« mal aux reins »).
- Affaiblissement et altération importante de l'état général(5).

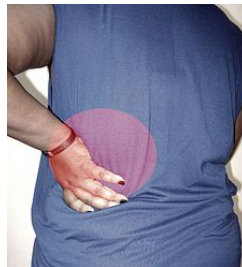


Figure 4 – Néphrite
(7)

II.4.3 Le diagnostic

- Anamnèse avec prise en compte des symptômes du patient.
- Examen physique.
- Mesure de la température pour exclure une inflammation du rein.
- Analyses des urines : recherche de bactéries, de globules blancs et rouges.
- Eventuellement, échographie pour exclure une autre maladie du petit bassin.
- A un stade avancé, un examen approfondi de la vessie, appelé cystoscopie, permet de visualiser l'intérieur de la vessie.
- Lors de l'examen des urines, il est impératif de collecter l'urine à mi-jet. Le premier jet d'urine doit être fait dans les toilettes ; ensuite, il convient de la collecter dans un récipient stérile et de finir la miction dans les toilettes.

II.4.4 Evolution de la maladie

L'infection urinaire fait courir de nombreux risques. Les infections urinaires basses (cystite, urétrite) peuvent, en cas de retard thérapeutique, évoluer vers une infection urinaire haute, c'est-à-dire touchant le rein (pyélonéphrite). Toute infection urinaire avec fièvre (pyélonéphrite, prostatite) peut se compliquer de septicémie (passage de germes dans la circulation sanguine), avec un risque de choc septique (défaillance des organes vitaux) qui nécessite une prise en charge en réanimation. L'infection peut provoquer un abcès du rein voire sa destruction.

Ces infections sont d'autant plus graves que le patient est fragile (diabète, déficit immunitaire, âge avancé ...).

Le risque ultime est l'insuffisance rénale terminale dont le traitement est l'hémodialyse (rein artificiel) voir la greffe rénale.

La prostatite possède des complications particulières qui sont la rétention aiguë d'urine (blocage complet de la vessie) et la dissémination de l'infection aux testicules (orchite).

En cas de grossesse, toute infection urinaire est dangereuse pour la mère mais également pour le fœtus. Il y a un risque d'accouchement prématuré.(8)

II.4.5 Options thérapeutiques

Le traitement de l'infection urinaire repose sur l'antibiothérapie, la majoration des boissons et le traitement des facteurs favorisants.

En cas de cystite simple ou d'urétrite (infection urinaire basse isolée non récidivante) :

- Soit un antiseptique urinaire pendant 10 jours.
- Soit un traitement antibiotique court sur 3 jours : antibiotiques de la famille des pénicillines ou des quinolones.
- Soit un traitement mono dose en prise unique. Cette dose unique est efficace sur la majorité des germes impliqués dans les cystites aiguës.

En cas de pyélonéphrite aiguë ou de prostatite aiguë, il est habituel de prescrire des antibiotiques pendant 2 à 3 semaines. L'antibiothérapie doit être adaptée aux résultats de l'antibiogramme.

En cas de fièvre importante, d'altération de l'état général ou de complication (septicémie, abcès rénal), une hospitalisation est nécessaire pour surveillance médicale intensive. Une double antibiothérapie est souvent prescrite par voie intraveineuse les 2 ou 3 premiers jours, avant de relayer par des comprimés.

Dans le cas d'une pyélonéphrite par obstacle (rein obstrué par un calcul par exemple), une intervention de drainage est indiquée en urgence afin de sauver la vitalité du rein. Celle-ci est réalisée par les voies naturelles le plus souvent.

II.5 Conseils hygiéniques

Comme de multiples facteurs interviennent pour faciliter la contamination de la vessie, de simples conseils hygiéniques suffisent souvent à faire disparaître les cystites chez la femme en période d'activité sexuelle :

- Boire plus de 1,5 litres par jour ;
- Effectuer des mictions complètes en étant détendue ;
- Pratiquer une toilette vulvaire au savon ;
- Effectuer une bonne toilette après les selles ;
- Lutter contre la constipation ;
- S'essuyer avec le papier hygiénique en allant d'avant vers l'arrière ;

- Eviter les toilettes périnéales trop énergiques ;
- Uriner immédiatement après les rapports sexuels ;
- Eviter les pantalons serrés et les sous-vêtements en fibres synthétiques qui favorisent la transpiration et la multiplication des germes.

II.6 Conclusion

Le diagnostic médical semble s'adapter difficilement aux contraintes de l'intelligence artificielle. Ceci se comprend aisément car il n'existe pas une seule forme de la même maladie, ce qui est appelé « forme clinique » dans le langage médical. L'aide au diagnostic consiste en la description d'un cas clinique à l'aide des signes issus de l'interrogation ou d'examens de laboratoire. L'approche exhaustive est difficile vu que le cerveau humain n'est pas fait pour garder un trop grand nombre de données, présent en esprit en même temps.

Aujourd'hui, vu l'intérêt croissant de faire gagner du temps, de conserver les données, de diagnostiquer, de donner le traitement en médecine et pas mal d'autres raisons. Dans ce cadre s'inscrit notre projet de fin d'études qui consiste à la conception et la réalisation d'un système expert pour le diagnostic des inflammations urinaires.

III Chapitre 3 : Le Système Expert

III.1 Introduction

Les experts humains sont capables d'effectuer un niveau élevé de raisonnement à cause de leur grande expérience et connaissance sur leurs domaines d'expertise. Un SE utilise la connaissance correspondante à un domaine spécifique afin de fournir une performance comparable à l'expert humain. En général, les concepteurs des SE effectuent l'acquisition de connaissance grâce à un ou plusieurs interviews avec l'expert ou les experts du domaine. Les humains qui enrichissent le système avec leurs connaissances ne fournissent pas seulement leur connaissance théorique ou académique mais aussi des heuristiques qu'ils ont acquises grâce à l'utilisation de leurs connaissances.

Contrairement à la modélisation cognitive, les SE n'ont pas comme finalité de s'inspirer des théories du fonctionnement du cerveau humain mais ce sont des programmes qui utilisent des stratégies heuristiques pour la résolution des problèmes spécifiques.

Le raisonnement effectué par un SE doit être objet à l'inspection, et ceci en fournissant d'information sur l'état de la résolution du problème et des explications sur les choix et les décisions du système.

D'un autre côté, la solution fournie par le système doit être évaluée par un expert humain et ceci dans le but de modifier l'information contenue dans la base de connaissances.

III.2 Le Système expert

Les machines sont-elles capables de penser ? (9) Une question qui est débattu depuis tous les premiers jours de naissance de l'I.A. L'intelligence d'une personne se mesure non pas seulement de ce qu'elle sait, mais par ce qu'elle peut faire de ce savoir, la façon dont elle l'utilise et l'applique. Ce savoir doit être réel et efficace. Les experts ont donc un passé enregistré et reconnu prouvons qu'ils sont capables d'utiliser ce savoir, ils sont connus par leur informations qu'ils fournissent, leurs connaissances de leurs limites, leur capacité à résoudre les problèmes et leurs explications qu'ils donnent.

Le domaine de l'I.A qui a eu le plus d'impact pratique est celui des systèmes basés sur le savoir intelligent appelé aussi systèmes experts.

D'une manière générale, un SE est un outil capable de reproduire les mécanismes cognitifs d'un expert, dans un domaine particulier. Il s'agit de l'une des voies tentant d'aboutir à l'I.A.

Plus précisément, un SE est un logiciel capable de répondre à des questions, en effectuant un raisonnement à partir de faits et de règles connus. Il peut servir notamment comme outil d'aide à la décision.

Plusieurs définitions existents dans la littérature sont :

Un SE comme le définit (Welbank 83) est un programme qui a une large connaissance dans un domaine restreint et qui utilise des raisonnements d'inférence complexes pour accomplir des tâches dont pourrait s'acquitter un expert humain.

Louis F.Frenzel définit un SE comme un programme d'I.A incorporant une base de connaissance et un moteur d'inférence.

C'est un logiciel ou circuit intégré destiné à remplacer ou assister l'homme dans des domaines où est demandée une expertise humaine considérable. Il permet l'expression facile et directe de la connaissance d'un expert humain, et l'exploitation de cette connaissance avec logique au cours d'un dialogue avec l'utilisateur. Il est capable de faire un enchaînement de déductions permettant ainsi d'inférer ou de produire de nouvelles connaissances. Le système expert peut manipuler indifféremment des connaissances de type variés pouvant être des idées, des jugements, des décisions, des propositions,

des prédictions.... etc.

Par ailleurs, un SE supporte facilement les révisions, les ajouts et les suppressions de faits ou de règles.

Les SE sont alors des programmes capables de simuler la démarche d'un expert humain pour la recherche de solutions à un problème donné. Cependant, un SE possède l'avantage (par rapport à l'expert humain) d'être duplicable et disponible. En plus, les connaissances d'un SE peuvent être tirées de plusieurs experts humains et sont enregistrées dans la mémoire du SE sans jamais être oubliées.(10)

III.2.1 Composants d'un SE

Architecture de base :

Un SE est composé de deux parties complémentaires :

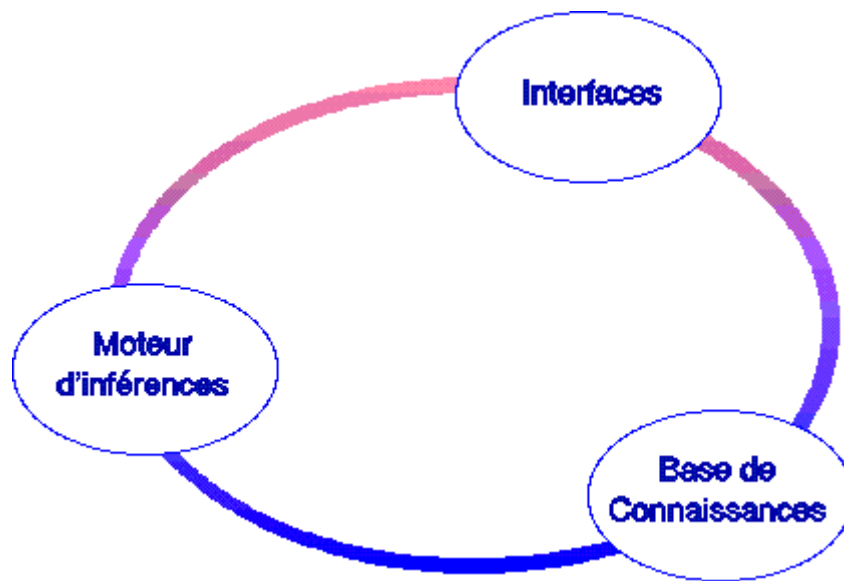


Figure 5 – Composants d'un SE
(10)

- **Une Base de Connaissances** : qui contient l'ensemble de la connaissance acquise dans un domaine d'application. Cette base de connaissances est constituée elle-même de deux parties :
 - **Une Base De Faits (BDF)** : qui contient la séquence de faits établis et ayant une valeur de vérité vraie. Elle constitue la partie statique de la base de connaissances.
 - **Une Base De Règle (BDR)** : qui contient l'ensemble des règles de production pouvant être appliqués aux faits pour déduire de nouveaux faits non existant préalablement dans la BDF. Il s'agit de la partie dynamique et fondamentale de la base de connaissance. En effet, un système expert peut démarrer son raisonnement avec une BDF vide, mais il ne peut faire aucun pas de raisonnement si sa BDR ne contient aucune règle.

La base de connaissances constitue ainsi la mémoire d'un SE.

- **Un Moteur d'Inférence (MI)** : il s'agit du cerveau du système expert. C'est un programme ou procédure qui simule le raisonnement humain par des applications répétitives du savoir-faire de l'expert, les règles, aux faits d'un problème pour déduire de nouveaux faits ou résultats. Au cours du traitement d'un problème, le MI peut générer plusieurs faits intermédiaires qui serviront de base pour l'application de nouvelles règles et ainsi de suite jusqu'à l'aboutissement

au résultat cherché ou à une impasse.

Le MI est une partie du SE qui est indépendante de la connaissance. Ainsi, un même Moteur d'inférence peut être appliqué à des bases de connaissances variées pour constituer à chaque fois un SE différent. C'est le cas de moteurs tels que EMYCIN et OPS, ainsi que le langage Prolog qui possède au sein de lui-même un moteur d'inférence, ce qui explique sa capacité de déduction est d'adaptation à différentes bases de connaissances.(10)

- **L'interface utilisateur** : sert à simplifier la communication, elle peut utiliser la forme question-réponse, Un moteur d'inférence fonctionnant en saturation (ou chaînage avant) est un programme qui applique à une base de faits toutes les règles non marquées d'une base de règle, afin d'obtenir une nouvelle base de faits. Ce principe est appliqué sur la nouvelle base de faits obtenue, jusqu'à ce que plus rien ne soit ajouté à la base. Toute règle non marquée, sélectionnée par le moteur et modifiant effectivement la base de fait, est marquée.

III.3 Chainage avant :

Le principe du chaînage avant est simple, il requiert l'accès aux prémisses (standards d'engagement) afin de déclencher les règles d'inférence adéquates définies par les metarules. L'application des règles (évaluations) donnent des résultats, ceux-ci sont évalués (par les metarules) afin de savoir si l'on a accédé à une solution finale potentielle. Si c'est le cas, on arrête et cette solution est proposée

- Si c'est le cas, la solution est proposée à l'utilisateur. S'il la valide, la solution est enregistrée dans la base de faits comme solution, sinon comme simple résultat et on continue dans le cas suivant.
- Si cela n'est pas le cas ou si la solution est refusée, la solution est enregistrée dans la base de faits comme simple résultat et le moteur d'inférence tente d'y appliquer d'autres règles jusqu'à trouver une solution potentielle validée, ou jusqu'à ce qu'il n'y ait plus de règle.

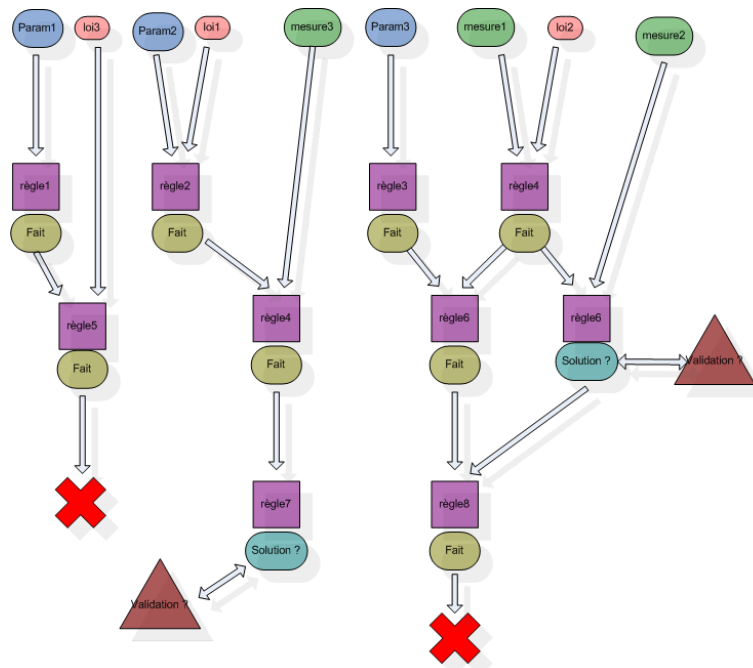


Figure 6 – Chainage avant
(10)

Un moteur fonctionnant en chaînage arrière procède à l'inverse du précédent : son objectif est de vérifier qu'un fait est vrai ou pas. Si le fait à démontrer est dans la base, le fait est trivialement vérifié. S'il n'est pas dans la base, alors le moteur recherche s'il existe au moins une règle qui conclut sur ce fait, et vérifie alors tous les faits prémisses. Si toutes les prémisses sont vérifiées, la règle est applicable, et l'exécution des actions de la règle rend le fait vérifié. Si l'un des prémisses n'est pas vérifié, essaie une autre règle. Si aucune règle ne convient, le fait n'est pas vérifié.

III.3.1 Algorithme du chaînage avant

Entree : BF, BR, F

Debut :

Tant que F n'est pas dans BF ET QU'il existe dans BR une règle applicable FAIRE choisir une règle ap

Si F appartient à BF

Alors F est établi

Sinon F n'est pas établi

Fin SI

Fin Tant que

Fin

Exemple :

1. Règle 1 : IF Y AND D THEN Z
2. Règle 2 : IF X AND B AND E THEN Y
3. Règle 3 : IF A THEN X
4. Règle 4 : IF C THEN L
5. Règle 5 : IF L AND M THEN N
6. La base de données comprend initialement les faits A, B, C, D et E

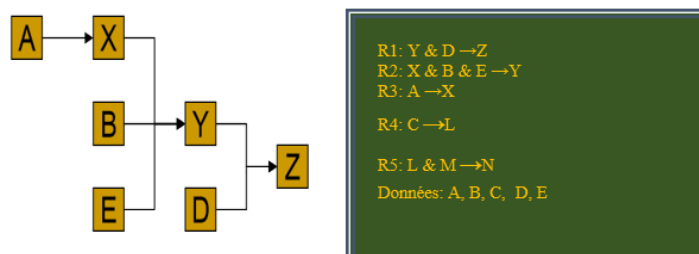


Figure 7 – Exemple de chaînage avant
(10)

Cycle 1 : on passe les règles jusqu'à ce qu'on en trouve une dont l'antécédent est vrai. R3 et R4 sont donc activées. X et L sont maintenant dans la base de données.

Cycle 2 : On repasse les règles qui n'ont pas été déclenchées au cycle précédent. R2 est donc déclenchée, puisque X est dans la base de données. Y est maintenant dans la base de données.

Cycle 3 : On repasse les règles qui n'ont pas été activées. R1 est déclenchée, et alors le but, Z, est activé.

III.4 Chainage arrière :

Le principe du chaînage arrière est plus compliqué, il s'agit dans ce cas de partir d'un effet ou d'une solution et de tenter de remonter la chaîne afin de déterminer les causes d'un effet (fait). La procédure est à partir d'un fait, de déterminer, grâce aux metarules, les règles d'inférence qui auraient pu être à l'origine de ce fait et de déterminer les paramètres les plus probables. A partir de là, on analyse les paramètres :

- Si le paramètre est un fait enregistré dans la base de faits, c'est qu'il est le résultat d'une règle (évaluation). La procédure précédemment décrite est donc relancée.
- Si le paramètre n'est pas un fait de la base de faits, on en reste là. On relève alors tous les faits et données retrouvés. Ils représentent les causes probables de la conséquence étudiée.

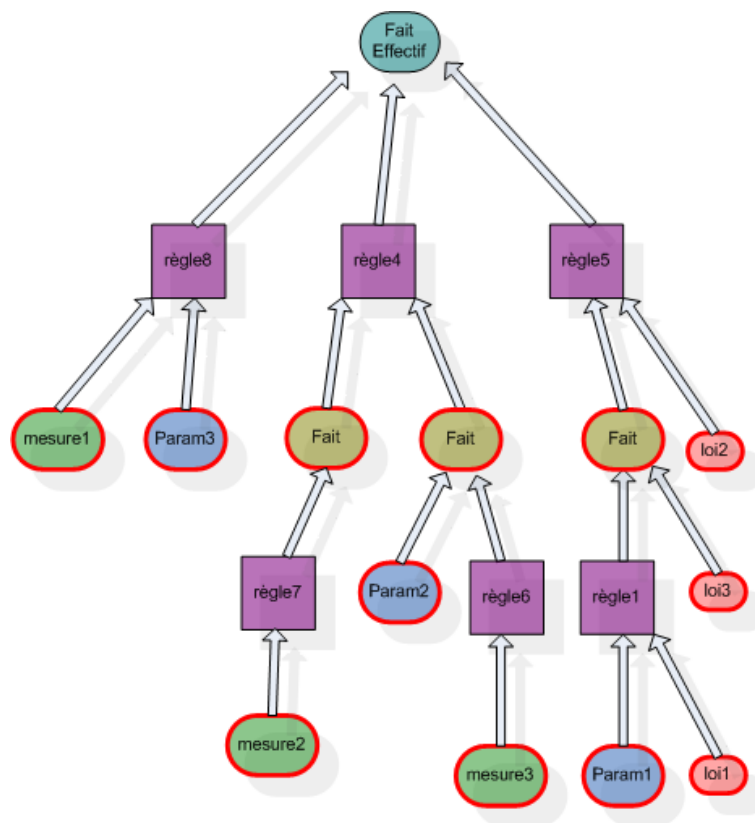


Figure 8 – chaînage arrière
(10)

Exemple :

Cycle 1 : on cherche une règle où Z est dans l'action. On le trouve (R1); le fait D est dans la base de données, mais il faut trouver Y.

Cycle 2 : on cherche une règle où Y est dans l'action. On le trouve (R2); les faits E et B sont dans la base de données, mais pas X.

Cycle 3 : on cherche une règle où X est dans l'action. On le trouve (R3); le fait A est dans la base de données.

Cycle 4 : On cherche A dans la base de données, et donc R3 est activée : X est dans la base.

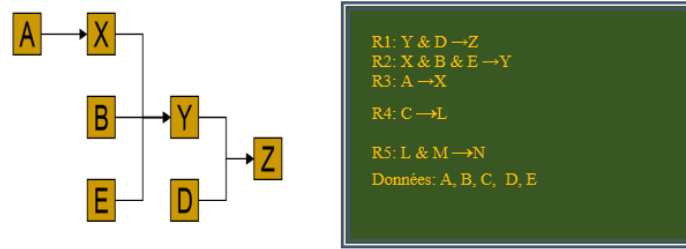


Figure 9 – Exemple chaînage arrière
(10)

Cycle 5 : On cherche à déclencher R2; X, B et E sont dans la base de données, alors Y est activé.

Cycle 6 : On cherche à déclencher R1; Y et D sont dans la base; alors Z est prouvé.

III.5 Chaînage mixte :

Il existe un dernier mode de fonctionnement dit chaînage mixte qui combine les 2 chaînages précédents. De prime abord, il fonctionne comme le chaînage avant avec pour but de déduire un fait donné. Mais applique un chaînage arrière sur chaque fait trouvé afin de déterminer les paramètres les plus probables et les plus optimisés. Ce mécanisme permet l'ouverture sur de nouvelles combinaisons encore non envisagées par les règles d'inférence et de déterminer les facteurs discriminants lors de la recherche d'une solution.(10)

III.5.1 Algorithme du chaînage mixte :

Entree : F (à déduire)

Debut :

Tant que F n'est pas déduit mais peut encore l'être FAIRE Saturer la base de faits par chaînage AVANT

Fin Tant que

Fin

Faire en sorte que toutes les informations puissent être entrées à partir d'un fichier texte.

Exemple : voici une petite base de règles :

1. règle 1 : si (poilu = oui) et (pelage = rayé) et (moustaches = oui) alors (animal = tigre)
2. règle 2 : si (poilu = non) et (poids > 2t) alors (Animal = éléphant)
3. règle 3 : si (poilu = non) alors (plumage = oui)
4. règle 4 : si (plumage = oui) et (poids < 50g) alors (Animal = moineau)

Si une base de faits contient au départ les faits (poilu=non) et (poids=35g), le moteur d'inférence ajoute, après application de toutes les règles possibles, le fait (Animal=moineau).

III.6 Réalisation d'un SE

III.6.1 Phase de conception

Comme pour le développement de tout logiciel, il est nécessaire de passer par une phase de conception pendant laquelle on dégage tous les éléments de l'application et le raccordement logique entre ces éléments. Pendant cette phase on détermine aussi les besoins et le type d'entrées/sorties nécessaire à l'application : quel type de problèmes peut rencontrer l'utilisateur, comment peut-il formuler ses besoins à l'ordinateur et quel genre de réponse souhaite-il obtenir. Nous pouvons donner ainsi un plan d'actions, sous forme d'un ensemble d'étapes à réaliser, cependant, nous ne pouvons pas prétendre qu'il existe une démarche rigoureuse est finie comme le cas de Merise appliquée aux systèmes de gestion classiques. Mais, le cogniticien ou l'ingénieur de la connaissance peut mener sa démarche en s'inspirant d'une méthode d'analyse classique ou moderne quelconque qu'il applique au domaine de l'expertise étudiée en passant par les techniques de l'informatique symbolique.

La phase de conception d'un SE peut alors être définie à l'aide des étapes suivantes :

1. Etablir un contact avec un ou plusieurs experts dans le domaine.
2. Etablir la liste des données logiques disponibles.
3. Etablir la liste des données à déduire et l'ensemble des règles de déduction.
4. Associer à chaque règle un degré de certitude dans le cas où les règles de déduction n'offrent pas des résultats sûrs.
5. Préparation de la base de connaissances équivalente : choix et rassemblement des prédicats et représentation symbolique ou graphique de l'ensemble des règles
6. Définition de l'interface utilisateur : sous le terme d'interface, on regroupe à la fois le moyen de communication (entrées/sorties en langage naturel) et la représentation des écrans et boîtes de dialogue.

III.6.2 Phase d'implantation

La partie critique et la plus intéressante dans un SE est son noyau (BDC + MI). Pour développer un SE, il est alors indispensable d'utiliser un système informatique spécifique qui offre au minimum un moyen souple de représentation de la connaissance. Deux choix se présentent :

1. Un premier choix consiste à utiliser un langage disposant déjà d'un MI indépendant est une souplesse de représentation des connaissances. On peut alors utiliser le langage PROLOG qui a donné de très bons résultats avec de nombreuses applications.
2. Un deuxième choix consiste à utiliser un langage spécifique à l'I.A avec lequel on peut faire un choix approprié du MI à utiliser et de la méthode de représentation de la connaissance. Dans ce cas, le programmeur doit réaliser en premier lieu un MI ensuite une implantation adéquate de la BDC. Pour ceci, on se base en général sur un langage tel que LISP. Le deuxième choix nécessite une maîtrise de la structure des MI mais possède l'avantage de l'adaptabilité aux besoins et au type de l'expertise à implanter. L'objectif de la deuxième partie de ce chapitre est de présenter une étude détaillée de la structure d'un MI et des différents algorithmes d'inférence possibles.(10)

III.7 Histoire de MYCIN

Le premier SE fut Dendral en 1965, créé par les informaticiens Edward Feigenbaum, Bruce Buchanan, le médecin Joshua Lederberg et le chimiste Carl Djerassi.

Il permettait d'identifier les constituants chimiques d'un matériau à partir de spectrométrie de masse et de résonance magnétique nucléaire, mais ses règles étaient mélangées au moteur. Il fut par la suite modifié pour en extraire le moteur de SE nommé Meta-Dendral

Le plus connu, peut-être, fut Mycin en 1972-73

III.7.1 Qu'est-ce qu'un Mycin ?

Un des premiers SE était Mycin développé à Stanford à partir de 1972 pour la diagnostique médicale. Le but de ce système expert interactif est d'aider les médecins à proposer la meilleure thérapie antimicrobienne pour leur patient atteints d'infections bactériologiques. A partir des symptômes, de l'histoire du patient et de résultats de tests de laboratoire, le système diagnostique la cause de l'infection et propose un traitement selon l'expertise de praticiens des maladies infectieuses. Le processus de sélection de traitement antibiotique est décomposé en quatre parties en faisant appel à une connaissance exprimée sous forme de règles de production :

- Recherche de l'indication d'un traitement de l'infection ;
- Si oui, identification du ou des micro-organismes susceptibles de donner une telle infection ;
- Sélection d'un ensemble de médicaments pouvant être prescrits ;
- Sélection dans cette liste du ou des médicaments à prescrire.

III.7.2 Objectifs de Mycin

L'équipe de shortliffe ont défini dès le début, un ensemble d'objectifs que Mycin doit l'atteindre et qui se résume dans les différents points suivants :

- Facilité d'utilisation.
- Fiabilité, flexibilité.
- Manipulation d'un grand nombre de connaissance.
- Utilisation des renseignements inexacts ou incomplets.
- Explication et justification de ses conseils : A tout instant, l'utilisateur peut demander :
WHY ? - Pourquoi ? - trace de la chaine de raisonnement en cours.
HOW ? - Comment ? - trace de source d'un fait.
- Apprentissage de connaissance.

III.7.3 Architecture de Mycin

Mycin est un Système expert qui est écrit en lisp, raisonne en utilisant le chaînage arrière et adapte la recherche en profondeur d'abord. Son architecture est décrite dans le schéma suivant :

- **Module d'interface** : sert à simplifier la communication.
- **Module d'explication** : réponds aux questions d'utilisateur et tente d'expliquer ses avis.
- **Base de connaissance** : la base de connaissance statique comporte des règles sous forme « si...alors ».
- **La base de faits** : contient les données spécifiques liées à l'application traitée. Elle peut contenir aussi les solutions intermédiaires ou les conclusions partielles trouvées lors de l'inférence.
- **Module d'acquisition de connaissance** : permet à l'expert d'ajouter des règles, ou de modifier celles existantes.(11)

MYCIN a une tâche en quatre étapes :

1. Décider quels organismes, le cas échéant, sont à l'origine des maladies significative.

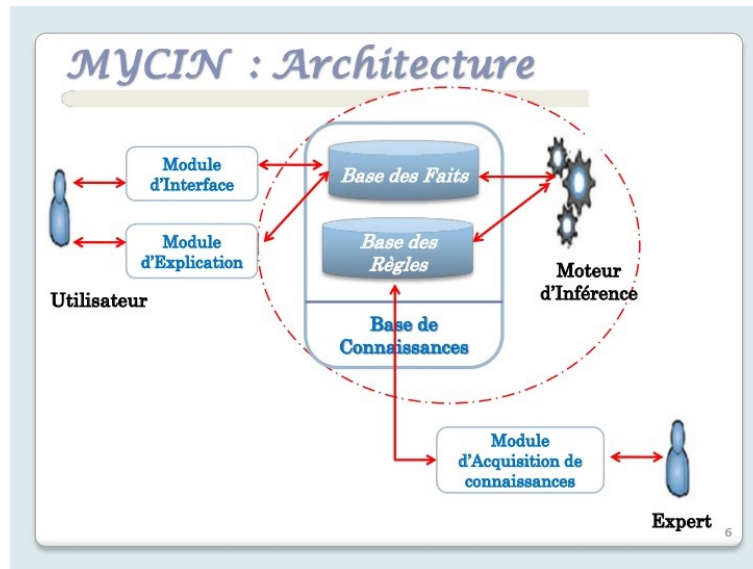


Figure 10 – Architecture de Mycin
(11)

2. Déterminer l'identité probable des organismes importants.
3. Décident quels médicaments sont potentiellement utiles.
4. Sélectionner le meilleur médicament, ou une combinaison de médicaments.

III.8 Avantages et Inconvénients :

III.8.1 Avantages d'un système expert :

Les systèmes experts ont plusieurs caractéristiques attrayantes :

- **Grande disponibilité** : L'expérience est disponible pour tout matériel de traitement adéquat. Dans un sens plus réel, un système expert est la production massive de l'expérience.
- **Coût réduit** : Le coût de mettre l'expérience à la disposition de l'utilisateur est réduit énormément.
- **Danger réduit** : Les systèmes experts peuvent être utilisés dans des environnements qui pourraient être dangereux pour un être humain.
- **Permanence** : L'expérience est permanente. Contrairement aux spécialistes humains qui peuvent se retirer, renoncer ou mourir, la connaissance d'un système expert durera indéfiniment.
- **Expérience multiple** : La connaissance des plusieurs spécialistes peut être disponible pour travailler simultanément et continuellement sur un problème, à toute heure de la nuit ou du jour. Le niveau d'expérience combinée de beaucoup de systèmes experts peut dépasser celui d'un seul spécialiste humain.
- **Explication** : Le système expert peut expliquer clairement et en détail le raisonnement qui conduit à une conclusion, cela augmente la confiance que la décision prise était correcte. Un être humain peut être fatigué, peut renoncer ou ne pas être capable de le faire toujours.
- **Réponse rapide** : Une réponse rapide ou en temps réel peut être nécessaire pour certaines applications. En fonction du logiciel ou matériel utilisé, un système expert peut répondre plus rapidement et être pus disposé qu'un spécialiste humain, de telle sorte qu'un système expert en temps réel constitue un bon choix.
- **Réponses solides, complètes et sans émotions, en tout moment** : Ceci peut être très important en temps réel et en des situations d'urgence, quand un spécialiste humain ne fonctionnera pas avec toute sa capacité à cause de la pression et de la fatigue.

- **Enseignement intelligent** : Un système expert peut agir comme un enseignant intelligent en laissant que l'étudiant exécute des programmes d'exemple et en expliquant le raisonnement du système.
- **Base de données intelligente** : Les systèmes experts peuvent être utilisés pour avoir accès à une base de données de manière intelligente.

III.8.2 Inconvénients d'un système expert :

Les deux principaux inconvénients des SE sont :

- Ils créent le chômage parce qu'ils émulent les humains ;
- Dans les systèmes experts, on fait inférence à des connaissances même si elles sont dépassées.(12)

III.9 Clips : L'outil de développement

CLIPS est un outil de système de développement et expert de livraison productive qui fournit un environnement complet pour la construction de la règle et/ou systèmes experts basés objet. Créée en 1985, CLIPS est maintenant largement utilisée à travers le gouvernement, l'industrie et le milieu universitaire. Ses principales caractéristiques sont :

- **Représentation des connaissances** : CLIPS fournit un outil cohésif pour la gestion d'une grande variété de connaissances compatible avec trois différents paradigmes de programmation : à base de règles, orienté objet et de procédure. La programmation basée sur des règles permet de représenter les connaissances heuristiques, ou « règles de base », qui spécifient un ensemble d'actions à exécuter pour une situation donnée. La programmation orientée objet permet à des systèmes complexes pour être modélisés comme des composants modulaires (qui peut être facilement utilisé pour modéliser d'autres systèmes ou pour créer de nouveaux composants). Les capacités de programmation procédurale fournies par CLIPS sont similaires à celles trouvées dans des langages tels que C, Java, Ada et Lisp.
- **Portabilité** : CLIPS est écrit en C pour la portabilité et la vitesse et a été installé sur de nombreux systèmes d'exploitation différents, sans modification de code. Les systèmes d'exploitation sur lesquels CLIPS a été testé comprennent Windows XP, Mac OSX, et Unix. CLIPS peuvent être portés à tout système qui a une ANSI C conforme ou le compilateur C de. CLIPS est livré avec tout le code source qui peut être modifié ou adapté pour répondre aux besoins spécifiques d'un utilisateur.
- **Intégration / Extensibilité** : clips peuvent être intégrés dans le code de procédure, appelé comme sous-programme, et intégrées à des langages tels que C, Java, Fortran et ADA. Les clips peuvent être facilement étendues par un utilisateur à travers l'utilisation de plusieurs protocoles bien définis.
- **Développement Interactive** : La version standard de CLIPS fournit un environnement interactif de texte axé sur le développement, y compris les aides de débogage, aide en ligne, et un éditeur intégré. Interfaces offrant des fonctionnalités telles que des menus déroulants, des éditeurs intégrés, et plusieurs fenêtres ont été développés pour Mac OS, Windows XP, et les environnements X Window.
- **Vérification / Validation** : CLIPS comprend un certain nombre de fonctionnalités pour soutenir la vérification et la validation de systèmes experts y compris le soutien pour la conception modulaire et le partitionnement d'une base de connaissances, statique et contrainte dynamique vérification des valeurs de sous et arguments de la fonction, et l'analyse sémantique des modèles de règles de déterminer si des incohérences pourraient empêcher une règle de tir ou générer une erreur.
- **Entièrement documenté** : CLIPS est livré avec une documentation complète, y compris un manuel de référence et un guide de l'utilisateur.

- **À Bas Prix** : CLIPS est maintenu en tant que logiciel de domaine public.(13)

III.10 Conclusion

Nous avons présenté dans ce chapitre le domaine des systèmes experts, ses composantes et les types de chaînages existents.

L'intelligence artificielle en médecine est principalement liée à la construction des programmes d'I.A qui effectuent le diagnostic et le traitement recommandé. Depuis plus de 20ans et grâce au développement de cette science pluridisciplinaires et de ces applications, il devient possible de développer des systèmes centrés sur l'action médicale, permettant aux cliniciens et aux médecins de bénéficier des possibilités offertes par cette approche et ces différentes méthodes avancées pour améliorer leurs connaissances, leurs décisions et maîtriser leurs activités.

IV Chapitre 4 : Logique Floue Et ANFIS

IV.1 Introduction

L'objectif de ce chapitre est de présenter une synthèse sur les systèmes flous. Pour ce faire, nous présentons les notions de base de la théorie des ensembles flous ainsi que les outils mathématiques nécessaires à leur manipulation. Un des objectifs du chapitre est de montrer l'intérêt que présente la logique floue dans l'identification et la commande de processus. La structure interne d'un système d'inférence floue, ainsi que les différents types de contrôleurs existants seront présentés et analysés.

IV.2 Théorie des ensembles flous

Jusqu'à une date récente, l'I.A a eu un impact limité sur les applications industrielles, parce qu'elle a mis l'accent, de façon exclusive, sur le traitement symbolique de la connaissance, par opposition à la modélisation numérique utilisée traditionnellement dans les sciences de l'ingénieur. Plus récemment, on a assisté à un retour du numérique dans ces problèmes d'I.A, avec les réseaux de neurones et la logique floue. Les réseaux de neurones proposent une approche implicite de type « boîte noire » de la représentation des connaissances, très analogue à la démarche de l'identification des systèmes en automatique. La logique floue est quant à elle plus conforme à l'intelligence artificielle symbolique, qui met en avant la notion de raisonnement, et où les connaissances sont codées explicitement, Elle permet de faire le lien entre modélisation numérique et représentation symbolique, ce qui a permis des développements industriels spectaculaires à partir d'algorithmes de traduction de connaissances symboliques en entités numériques et inversement. La théorie des ensembles flous a également donné naissance à un traitement original de l'incertitude, fondé sur l'idée d'ordre et non plus sur l'additivité comme dans les approches probabilistes. Les ensembles flous ont également eu un impact sur les techniques de classification et ont contribué à la proposition de nouvelles approches pour l'aide à la décision.

C'est à L.Zadeh, spécialiste renommé en automatique et théorie des systèmes, que revient le mérite de faire la jonction en 1965 entre les logiques multivaluées des années 1930 et les travaux de Black sur les concepts du « vague », grâce à la notion d'ensemble flou et aux logiques associées. Celles-ci sont en fait une extension de la logique Booléenne, C'est aussi Zadeh qui a établi en 1978, le lien entre la théorie des ensembles flous et la modélisation non probabiliste de l'incertitude, grâce à la théorie des possibilités. Dans cette dernière, les fonctions d'appartenance des ensembles flous servent à représenter les idées de valeurs incomplètement spécifiées, mal connues, et de contraintes flexibles. Les ensembles flous et la théorie des possibilités sont à la base d'une méthodologie de représentation et d'utilisation de connaissances imprécises, vagues et incertaines, que Zadeh appelle théorie du raisonnement approché, plus connue sous le vocable de « logique floue »,⁽¹⁴⁾

IV.3 Le Concept d'ensemble flou

Ce concept permet de considérer des classes d'objets dont les frontières ne sont pas clairement déterminées, par l'introduction d'une fonction d'appartenance des objets à la classe, prenant des valeurs entre 0 et 1, contrairement aux ensembles « Booléens » dont la fonction caractéristique ne prend que deux valeurs possibles 0 ou 1,

Les classes d'objets rencontrées dans le monde réel ne possèdent pas de critères d'appartenance bien définis. Ce constat ne fait que souligner le fossé qui sépare les représentations mentales de la réalité, des modèles mathématiques usuels (à base de logique binaire, de nombres réels, d'équations différentielles, etc...). Les classes d'objets auxquelles Zadeh fait allusion n'existent qu'au travers

de ces représentations mentales et correspondent à des termes vagues du langage naturel, tels que « température élevée », « puissance moyenne », etc.

La notion d'ensemble classique semble mal adaptée pour représenter des classes de ce type. Par exemple, si l'on considère le concept « homme jeune », il est difficile de proposer un seuil en dessous duquel un homme sera considéré comme « jeune ». L'idée de Zadeh a été de suggérer qu'au lieu de chercher, à tout prix, un seuil unique pour l'appartenance à l'ensemble des âges « jeunes » dans un contexte donné, il semblait plus réaliste de considérer deux seuils s_1 et s_2 , tels que le terme jeune s'applique parfaitement aux âges plus petits que s_1 (par exemple 19 ans), et ne s'applique plus du tout au dessus de s_2 . Les âges plus petits que s_1 auront le degré d'appartenance maximal (en général supposé égal à 1) et les âges plus grands que s_2 (par exemple 40 ans) auront un degré d'appartenance minimal (en général égal à 0). Entre s_1 et s_2 , les degrés d'appartenance seront intermédiaires, par convention entre 0 et 1. (14)

IV.4 Définition

Soit X , un ensemble dénombrable ou non. Un sous-ensemble flou A de X est caractérisé par sa fonction d'appartenance f_A , telle que :

$$f_A : X \rightarrow [0, 1]$$

$$x \rightarrow f_A(x) \quad (1)$$

où $f_A(x)$ représente le degré d'appartenance avec lequel x appartient à l'ensemble flou A . Soit $F(X)$ l'ensemble des flous de X . (14)

Exemple

L'ensemble flou I couramment utilisé pour représenter l'ensemble des réels ayant des bornes non rigides, est décrit par la fonction d'appartenance f_I admettant une représentation trapézoïdale. La notation $I = (a, b, \alpha, \beta)$, est souvent utilisée dans les applications informatiques pour représenter une telle

$$f_I(x) \begin{cases} 0 & \text{si } (x \leq a - \alpha) \text{ ou } (x \geq b + \beta) \\ 1 & \text{si } (a \leq x \leq b) \\ 1 + (x - a/\alpha) & \text{si } (a - \alpha \leq x \leq a) \\ 1 - (b - x/\beta) & \text{si } (b \leq x \leq b + \beta) \end{cases} \quad (2)$$

IV.5 Caractérisation d'un ensemble flou

Les ensembles flous ont le grand avantage de constituer une représentation mathématique de labels linguistiques largement utilisés dans l'expression de connaissances expertes, qualitatives et manipulées dans un raisonnement approché.

Ils apparaissent donc comme un moyen de réaliser l'interface entre l'information numérique (quantitative) et l'information symbolique (linguistique, qualitative). Pour montrer dans quelle mesure un sous-ensemble flou A d'un référentiel X diffère d'un sous-ensemble classique de X , nous définissons

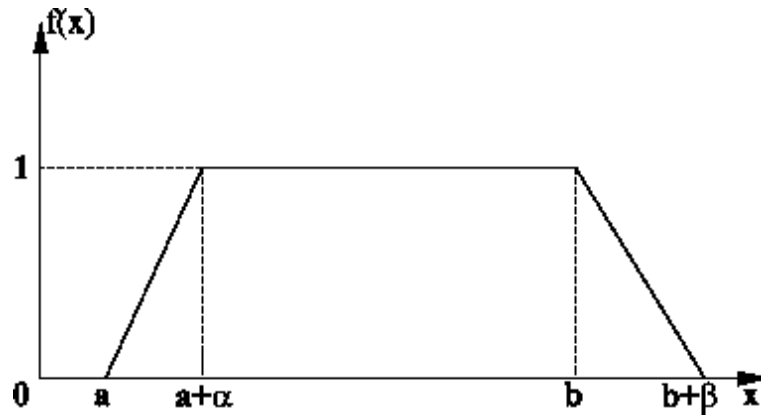


Figure 11 – Ensemble flou trapézoïdal
(14)

les caractéristiques essentielles suivantes :(14)

Notations

$$A = \sum_{x \in X} f_A(x)/x \quad \text{si l'ensemble de rfrence } X \text{ est fini}$$

$$A = \int f_A/x \quad \text{si l'ensemble de rfrence } X \text{ est infini}$$

(3)

Définitions

1. Support d'un sous-ensemble flou : Le support d'un sous-ensemble flou A est caractérisé par l'ensemble des éléments de X, qui sont à des degrés divers dans A, et que l'on note par $\text{Supp}(A)$ défini par :

$$\text{Supp}(A) = \{x \in X / f_A(x) \neq 0\}$$

(4)

2. Noyau d'un sous-ensemble flou : Le noyau d'un sous-ensemble flou A est caractérisé par l'ensemble des éléments de X qui sont réellement dans A, et que l'on note par $N(A)$ défini par :

$$N(A) = \{x \in X / f_A(x) = 1\}$$

(5)

3. Hauteur d'un sous-ensemble flou : La hauteur H d'un sous-ensemble flou A est définie à l'aide de l'opérateur Sup , telle que :

$$(6) \quad H(A) = \sup_{x \in X} f_A(x)$$

Remarque

Le sup n'est pas forcément atteint par la fonction d'appartenance f_A . En prenant l'exemple de la fonction $\tanh(x)$, les caractéristiques du sous-ensemble flou A sont décrits de la façon suivante :

(14)

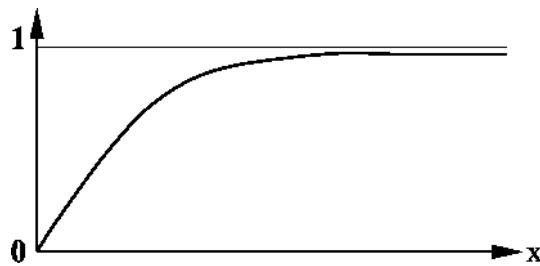


Figure 12 – Tangente hyperbolique
(14)

1. $H(A) = 1$... Sans qu'il soit atteint ;
2. $N(A) = \emptyset$;
3. $\text{Supp}(A) = \mathbb{R}$;

IV.6 Implication floue

Une implication floue est une proposition floue qui possède la forme suivante :

$$\text{Si } (V \text{ est } A_i) \text{ alors } (W \text{ est } A_j)$$

D'une manière générale, les propositions floues sont construites à l'aide de différents opérateurs, comme la disjonction, la conjonction ou bien l'implication portant sur des propositions floues quelconques. La particularité d'une proposition floue est qu'elle utilise une implication floue entre deux propositions floues ayant la forme suivante :

$$\begin{aligned} &\text{Si } (I \text{ est } A_i) \text{ et } (II' \text{ est } A_j) \\ &\quad \text{alors } ((\text{ est } A_k) \end{aligned}$$

On parle également dans ce cas de règle floue,

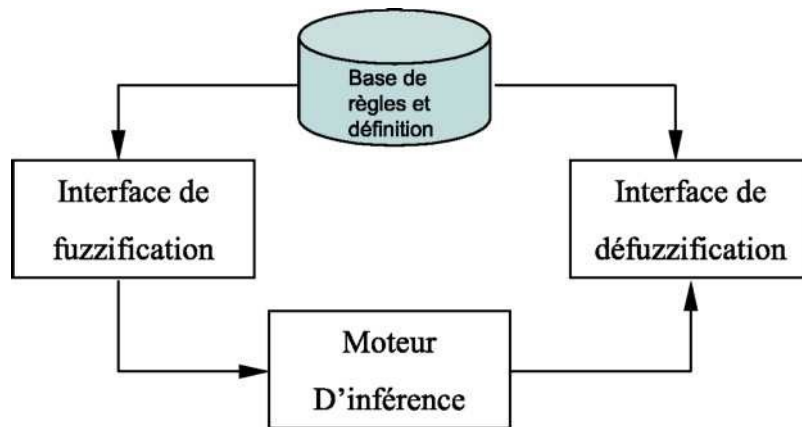


Figure 13 – Structure d'un système flou

IV.7 Structure interne d'un système d'inférence flou (SIF)

Le traitement réalisé par un système d'inférence flou (SIF) est divisé en trois tâches essentielles :

1. La fuzzification transforme une valeur numérique XQ de l'entrée en une valeur floue,
2. L'inférence, produit l'image de la partie floue issue de la fuzzification par une relation floue R , généralement construite à partir de règles.
3. La défuzzification transforme la partie floue issue de l'inférence en une valeur numérique y en sortie, La défuzzification constitue alors une prise de décision. La mise en oeuvre de ces trois étapes de traitement dépend de la nature des sous-ensembles flous utilisés. En d'autres termes, ces sous-ensembles peuvent être de type numérique ou symbolique. De même, la relation floue R peut être numérique (liant des nombres entre eux), symbolique (liant des symboles entre eux), ou bien hybride (numérique et symbolique). Dans ce qui suit, nous procédons à une étude plus détaillée des parties constituant un SIF et établissons les différents types de contrôleurs flous,

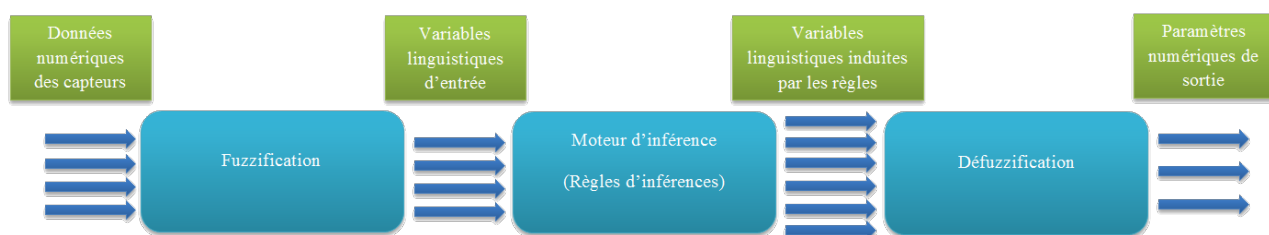


Figure 14 – Fonctionnement d'un système Flou

IV.7.1 Fuzzification

L'étape de fuzzification a pour but de transformer une donnée numérique en variable linguistique. Pour cela, le concepteur du système flou doit créer des fonctions d'appartenances. Une fonction d'appartenance est une fonction qui permet de définir le degré d'appartenance d'une donnée numérique à une variable linguistique.

IV.7.2 L'inférence

Rappelons d'abord que l'inférence transforme, à l'aide d'un jeu de règles, la partie issue de la fuzzification en une nouvelle partie floue. Cette dernière caractérise la sortie du contrôleur. Comme pour la fuzzification, on peut considérer deux cas : une caractérisation numérique ou une caractérisation symbolique de la sortie du contrôleur. En d'autres termes, la partie floue issue de l'inférence peut être soit une partie floue de l'ensemble de référence U , soit une partie floue de $L(IJ)$ (ensemble des symboles linguistiques utilisés dans les conclusions des règles).

Comme nous avons mis en évidence deux types de fuzzification, on aboutit à quatre familles de contrôleurs, comme le montre la figure 1,6. Dans ce qui suit, nous appelons contrôleur flou de type FCi (FC pour Fuzzy Controller) un contrôleur dont le mécanisme d'inférence est noté par g_i ,

L'inférence repose sur l'utilisation du principe de combinaison-projection (Figure 1,7). Ce principe proposé par Zadeh, n'est rien d'autre que l'extension au cas flou, (14)

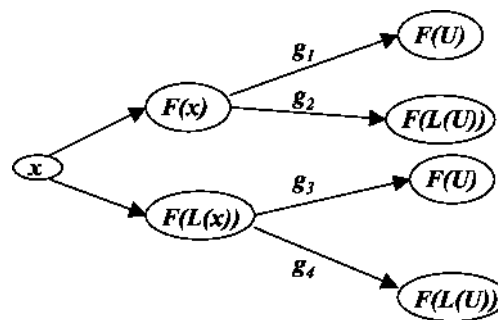


Figure 15 – Familles de contrôleurs flous
(14)

IV.7.3 La défuzzification

Le fonction de défuzzification consiste à transformer la partie floue issue de l'inférence en une grandeur numérique de commande. Le problème se pose de façon différente selon que le résultat de l'inférence est une partie floue de U (contrôleurs FCi et FC3) ou de $L(IJ)$ (contrôleurs de type FC2 et FC 4). Pour classifier les deux approches, on parlera de défuzzification numérique et de défuzzification symbolique. Dans ce qui suit, on considère, que l'ensemble U est fini et que F est la partie floue issue de l'inférence.

a) La défuzzification numérique

Il existe différentes méthodes de défuzzification, parmi lesquelles on peut citer celle du centre de gravité, de la moyenne des maximums, etc.

La méthode du centre de gravité considère la partie floue issue de l'inférence, comme une surface dont on calcule la projection sur l'axe horizontal de son centre de gravité.

La méthode de la moyenne des maximums consiste quant à elle, à prendre la moyenne des éléments ayant le plus grand degré d'appartenance. Il s'agit d'un calcul de moyenne tout à fait classique. Pour ce faire, on applique la relation précédente à la partie nette obtenue par l'acoupe de niveau de la partie F , notée F_a , où a est le plus grand degré d'appartenance.

b) la défuzzification symbolique

Il s'agit d'un problème moins classique et très peu traité dans la littérature. Le principe consiste à calculer une valeur numérique à partir de la partie floue de symboles, issue de l'inférence. Une solution envisageable consiste à transformer la partie de symboles en une partie floue du domaine numérique U , puis à appliquer les méthodes de défuzzification numérique. Une autre solution consiste à effectuer cette opération en construisant un graphe flou à partir de la signification des symboles :

$fr(L,u) = /t(l)(w)$ On applique ainsi le principe de combinaison-projection pour calculer l'image de la partie F à partir du graphe précédent.

Dans ce qui suit, nous présentons le contrôleur de Sugeno sur lequel nous nous appuierons pour notre approche,

IV.8 Exemple de système flou

Pour finir cet article, voici un petit exemple très simple de système flou qui a pour avantage d'avoir une défuzzification assez simple puisque les variables de sorties sont déjà des variables numériques et non linguistiques. Il suffit, pour défuzzifier de fusionner ces variables numériques sans manipuler des fonctions d'appartenances.

L'exemple en question est un système flou qui a pour but de faire de l'interpolation de points. Les connaissances que l'on a du système sont les coordonnées de cinq points :

Point 1 : (0, 0)

Point 2 : (0.4, 0.25)

Point 3 : (0.5, 0.5)

Point 4 : (0.6 ; 0.75)

Point 5 : (1, 1)

On souhaite donc interpoler un point quelconque dont l'abscisse est comprise entre 0 et 1. En entrée

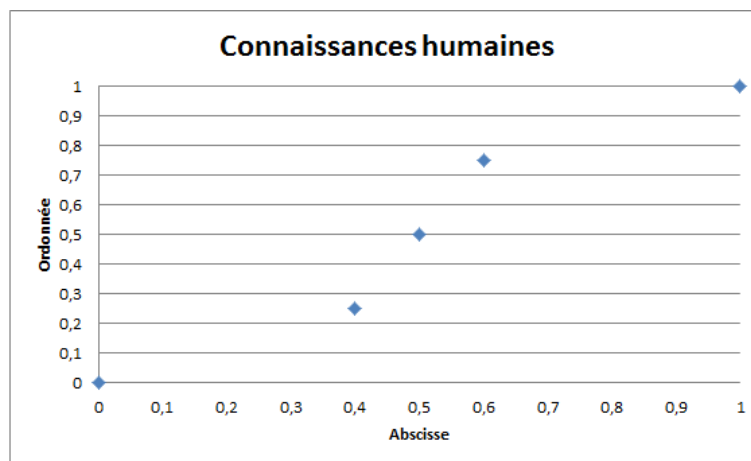


Figure 16 – Connaissance humaine
(15)

du système, on a l'abscisse d'un point compris entre 0 et 1 et en sortie, on veut l'ordonnée du point interpolé.

Pour cet exemple, nous voulons essayer d'interpoler trois points dont les abscisses sont 0.3, 0.55 et 0.95

Etape 1. La fuzzification

La première chose à faire, c'est de définir les variables linguistiques qui nous seront utiles ainsi que leurs fonctions d'appartenances. Pour cet exemple, je vais choisir cinq variables linguistiques qui caractériseront l'abscisse d'un point. Chacune des variables linguistiques caractérisera un des cinq points que l'on connaît. Les variables linguistiques seront nommées ainsi : low, medium_low, medium, medium_high, high. Leurs fonctions d'appartenance seront des fonctions triangles qui seront centrées en chacune des abscisses des points connus. De plus, je choisis que la somme de toutes les fonctions d'appartenances en un point donné donne 1. (Cette propriété n'est pas obligatoire, c'est

juste pour simplifier les choses par la suite.)

Voila les cinq fonctions d'appartenances résumés ci-dessous :

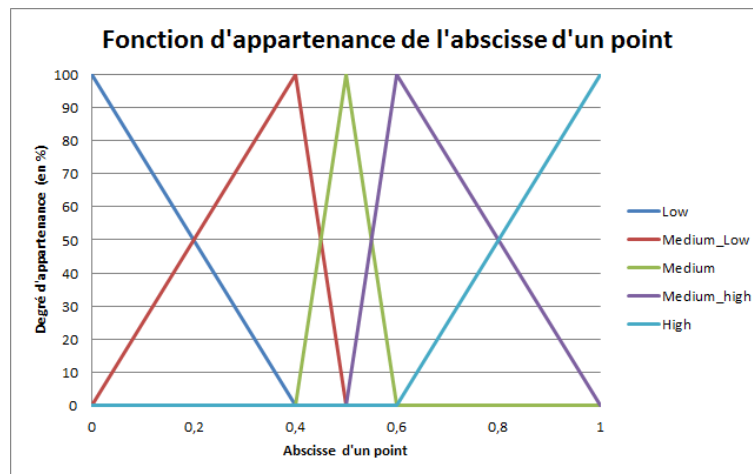


Figure 17 – Fonction d'appartenance de l'abscisse d'un point
(15)

Avec ces fonctions d'appartenances, on remarque que chaque point des connaissances est caractérisé à 100% par une seule et unique variable linguistique. (Par exemple, le point 4 est $medium_{high} 100\%$)

Nous pouvons maintenant fuzzifier les trois points interpoler!

Pour $x_1 = 0.3$, x_1 est low 25% et medium_low 75%

Pour $x_2 = 0.55$, x_2 est medium 50% et medium_high 50%

Pour $x_3 = 0.95$, x_3 est medium_high 12.5% et high 87.5%

Etape 2. Le moteur d'inférence

Dans cette partie, on doit tout d'abord définir un ensemble de règles d'inférences dérivé des connaissances humaines. Ici, les règles d'inférences sont plutôt simples. Elles sont au nombre de cinq :

Si x est low, alors y = 0

Si x est medium_low, alors y = 0.25

Si x est medium, alors y = 0.50

Si x est medium_high, alors y = 0.75

Si x est high, alors y = 1

On se rend compte que dans notre cas, la sortie n'est pas vraiment à 100% une variable linguistique, car elle comporte déjà une valeur. C'est cette astuce qui va nous soulager des calculs complexes de la défuzzification.

Si on passe les trois points à interpoler dans le moteur d'inférence, on a :

Pour $x_1 = 0.3$: $y_1 = 0$ à 25% et $y_1 = 0.25$ à 75%

Pour $x_2 = 0.55$: $y_2 = 0.5$ à 50% et $y_2 = 0.75$ à 50%

Pour $x_3 = 0.95$: $y_3 = 0.75$ à 12.5% et $y_3 = 1$ à 87.5%

Etape 3. La défuzzification

Vient enfin la dernière étape de défuzzification. Pour cette étape, pas besoin de calculs complexes ! Il suffit de faire la moyenne pondérée des sorties et comme la somme des valeurs des fonctions d'appartenance vaut 1 (25%+75%, 50%+50% et 12.5%+87.5%), pas besoin de diviser par 1 :)

Pour $x_1 = 0.3$: $y_1 = 0 * 25\% + 0.25 * 75\% = 0.188$

Pour $x_2 = 0.55$: $y_2 = 0.5 * 50\% + 0.75 * 50\% = 0.625$

Pour $x_3 = 0.95$: $y_3 = 0.75 * 12.5\% + 1 * 87.5\% = 0.969$

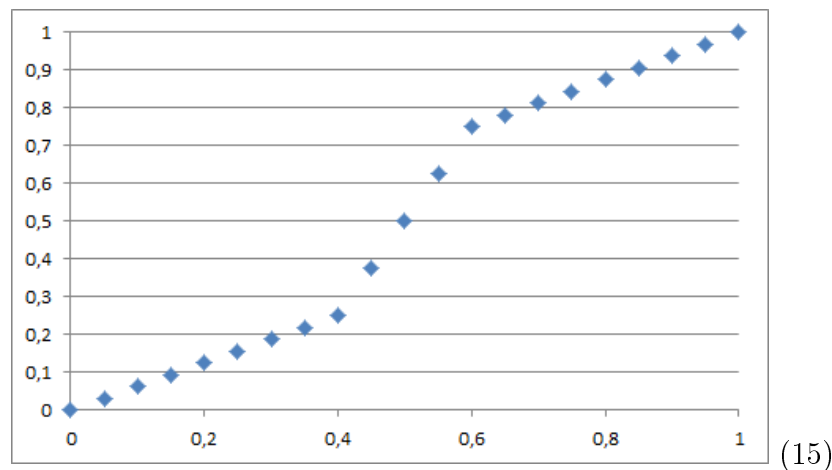
On obtient donc l'interpolation de nos trois points grâce à un système flou :

(0.3, 0.19)

(0.55, 0.63)

(0.95, 0.97)

On peut ainsi interpoler n'importe quelle point d'abscisse comprise entre 0 et 1.



Dans notre cas, on remarque que l'interpolation est linéaire entre deux points 'connaissances'. Pour rendre l'interpolation non linéaire, il suffit de redéfinir les fonctions d'appartenance afin que la somme des fonctions ne fassent plus forcément.

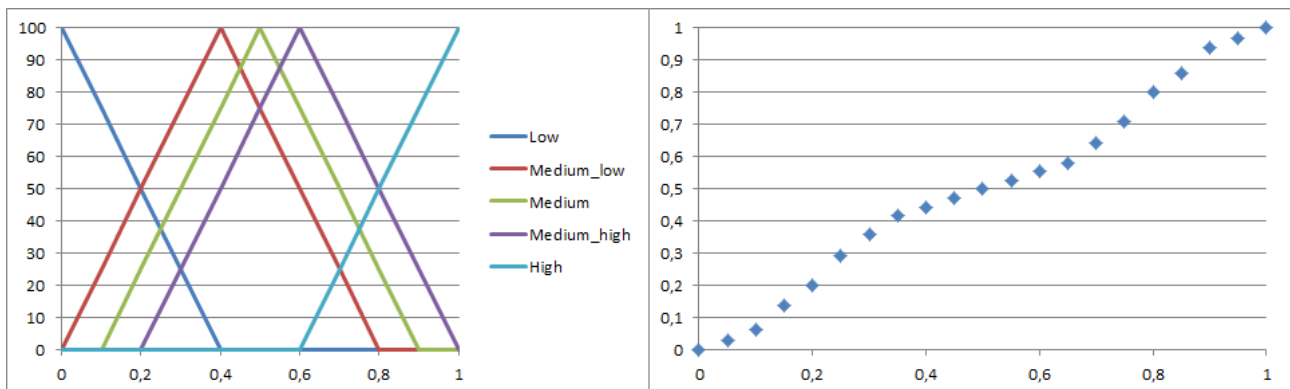


Figure 18 – Exemple d'interpolation non linéaire
(15)

En conclusion, pour réaliser un système flou, il faut dans un premier temps modéliser les données d'entrées en variables linguistiques grâce à des fonctions d'appartenance, dans un second temps, dresser une liste de règles d'inférences qui représentent les connaissances que l'on a du système et enfin, il faut choisir quels sont les opérateurs logiques à utiliser et quel type de defuzzification on souhaite utiliser.

Un système flou est entièrement customisable. Ses performances seront à l'image des réglages que le concepteurs du système à bien voulu faire. Un système flou peut être très performant comme

totallement inutile ! (Par exemple, mon dernier exemple d'interpolation non linéaire n'est pas du tout performant car elle déforme les valeurs connaissances ! Ainsi, l'interpolation de $x=0.4$ ne fait plus du tout $0.25 : s$)(15)

IV.9 L'Anfis : Neuro-Flou

Les systèmes flous ont été conçus pour exploiter les informations linguistiques des connaissances d'experts pour l'étude d'un système [JS95, PY98]. Toutefois, au cours des années, le concept s'est élargi et les systèmes adaptatifs flous ont pris de l'importance pour l'extraction automatique des connaissances (en l'absence d'un expert humain) à partir d'un ensemble de données. Ils ont été mis en oeuvre pour construire des modèles efficaces avec des algorithmes d'apprentissage sophistiqués.

Bon nombre de ces systèmes adaptatifs sont développés par hybridation avec d'autres méthodes, par exemple réseaux de neurones, algorithmes génétiques, programmation évolutive, des méthodes probabilistes, etc [Bon97]. Jusqu'à présent, le plus populaire de ces systèmes hybrides, est celui du neuro-flou (les réseaux de neurones flous) qui exploite la force d'apprentissage des réseaux de neurones ainsi que la facilité de compréhension linguistique du système à base de règles floues. L'apprentissage est utilisé de manière adaptative, afin d'ajuster les règles dans la base de connaissances, et de produire ou d'optimiser les fonctions d'appartenance du système flou.

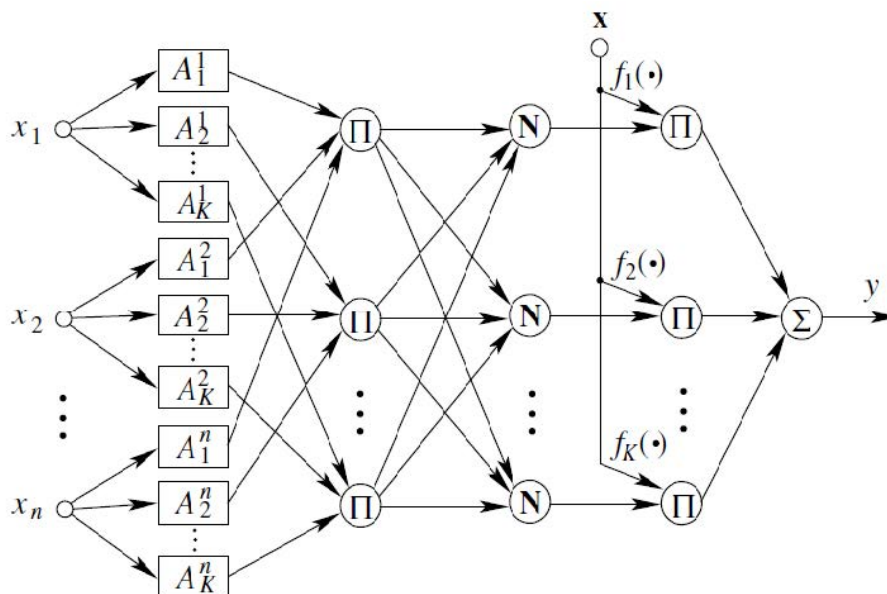


Figure 19 – Architecture d'ANFIS

Jang dans [Jan93] a mis au point ANFIS (Adaptive Neuro-Fuzzy Inference System) qui est un système neuro-flou. Avec une structure de réseau de neurones, où chaque couche est un composant du système neuro-flou (figure 12). Il applique le modèle de règle de TSK (Takagi-Sugeno-Kang) flou [SK88] dans lequel les règles ont un caractère flou juste dans la partie $\ddot{S}I$; tandis que dans la partie $\ddot{A}lors$; il y a des dépendances fonctionnelles.

Algorithm 4 Principe de l'architecture d'ANFIS a deux entrées

1 : 1ère couche : Fuzzyfication
 $O_i^1 = u_{A_i}(X)$ avec $i = 1, 2$

$$O_i^1 = u_{Bi-2}(Y) \text{ avec } : i = 3, 4$$

2 : 2^{mecouche} : Pondrationdesrglesfloues

$$O_i^2 = w_i = u_{Ai}(X)u_{Bi-2}(Y) i = 1, 2$$

3 : 3^{mecouche} : Normalisation

$$O_i^3 = \bar{w}_i = w_i / w_1 + w_2 i = 1, 2$$

4 : 4^{mecouche} : Dfuzzyfication

$$O_i^4 = w_i f_i \text{ avec } f = a_i.X + b_i.Y + c_i, \text{ ou } , i = 1, 2$$

5 : 5^{mecouche} : Calculdelasortie

$$O_i^5 = \sum w_i f_i = \sum w_i f_i / w_1 + w_2$$

Toutefois, dans la conception des systèmes neuro-flous, on relève plusieurs difficultés :

- Détermination du nombre de fonctions d'appartenance : il est lié à l'identification de la structure (16),
- Choix des centres et de la largeur des fonctions d'appartenance,
- Besoin d'une base a large exemples pour tester les modèles,
- Perte de la capacité de généralisation pour une base de données importante,
- Explosion combinatoire du nombre de règles floues (temps d'apprentissage),
- Fixation des paramètres de fonctions d'appartenance.

Une partition optimale de l'espace d'entrée peut diminuer le nombre de règles et augmenter la vitesse d'apprentissage. Par conséquent, choisir le nombre de fonctions d'appartenance et trouver les paramètres optimaux de ces dernières deviennent des questions ouvertes.

Pour remédier aux problèmes suscités, nous exploitons les méthodes de regroupement pour la répartition des données et permettre ainsi la régularisation des contraintes qui s'appliquent sur les paramètres des fonctions d'appartenance floues. Cela en éliminant la similarité par la fusion des partitions floues afin d'améliorer leurs distinctions et de diminuer le nombre de sous-ensembles flous (17). De ce fait, automatiser et optimiser la structure et les paramètres des fonctions d'appartenance. Pour résoudre ce problème, nous proposons dans cette partie l'implémentation de l'algorithme Fuzzy C-means et à titre comparatif l'algorithme Soustractive clustering

IV.10 L'apprentissage du classifieur Neuro-flou

Dans ce travail, deux phases d'apprentissage sont nécessaires : Premièrement, une phase d'apprentissage structurel est appliquée pour déterminer le bon partitionnement de l'espace d'entrée (nombre de fonctions d'appartenance pour chaque entrée). Deuxièmement, une phase d'apprentissage paramétrique est utilisée pour affiner les fonctions d'appartenance et les paramètres en conséquence. Il y a plusieurs façons de combiner l'apprentissage structurel et paramétrique dans un classifieur neuroflou. Elles peuvent être réalisées de manière séquentielle : apprentissage de la structure en premier lieu pour trouver la structure appropriée d'un système neuro-flou et l'apprentissage paramétrique est ensuite réalisé pour réajuster les paramètres. Dans d'autres cas, seul l'apprentissage paramétrique ou structurel est nécessaire lorsque la structure (règles floues) ou les paramètres (fonctions d'appartenance) sont donnés par des experts. Par ailleurs la structure de certains modèles neuroflous est fixée à priori.

ANFIS ajuste les paramètres de fonction d'appartenance en adoptant soit un algorithme de rétro propagation seul ou en combinaison avec une estimation des moindres carrés, pour réduire une certaine mesure d'erreur définie par la somme d'écart quadratique entre les résultats réels et désirés.(18)

IV.11 Conclusion

Nous avons présenté dans ce chapitre une synthèse sur la théorie des ensembles flous, ainsi que les outils mathématiques nécessaires à leur manipulation. Nous avons ainsi montré l'intérêt de cette théorie pour formaliser le traitement de l'inconsistance et de l'incertitude sur le modèle d'un système et pour en interpréter le comportement dynamique par un ensemble de règles linguistiques, traduites en langage simple et interprétable. La commande de systèmes complexes constitue un domaine d'application intéressant pour l'application de cette théorie. Dans ce cadre, nous avons présenté et analysé la structure interne d'un système d'inférence flou, ainsi que les différents types de contrôleurs existants ; chaque type s'appliquant à une classe donnée de problèmes. La synthèse d'un contrôleur flou pour un système MIMO, constitue une problématique de recherche non triviale et nécessite l'emploi d'une méthodologie pour la génération de la base de connaissances. L'identification des règles de décision qui doivent être encapsulées dans un contrôleur flou ne peut être réalisée manuellement par observation du ou des experts pilotant le système. Notre objectif est de proposer une méthodologie et une algorithmique pour la génération automatique de contrôleurs flous MIMO, Dans le chapitre suivant, nous présentons et analysons les différentes méthodes de génération de règles floues.

V Chapitre 5 : La Conception du Système Expert INFUR

V.1 Introduction

Bellman a dit en 1978 : « automatisation des activités associées au raisonnement humain, telles que la décision, la résolution de problèmes, l'apprentissage... », c'est ce que nous allons prouver dans ce dernier chapitre.

Notre objectif est la conception et la réalisation d'un SE interactif, fiable et flexible pour aider les médecins ou les urologues spécialement à proposer la meilleure thérapie pour leur patient atteints les infections Urinaires, à partir des symptômes de l'histoire du patient et du diagnostic proposé par le Système. Nous Commençons par la description de la base de données utilisée ainsi que les étapes fondamentaux de notre étude.

Cette conception est en général organisée comme suite :

1. Description et analyse de la base de donnée ;
2. Nous proposons deux techniques pour Générer les règles, la 1ère technique est basée sur un modèle Neuro-Flou (ANFIS) et la deuxième technique est basée sur la logique Flou en utilisant l'outil FISPRO.
3. Comparer le résultat des deux techniques conjointement à un expert pour prendre le meilleur.
4. La Conception d'un Système Expert en utilisant l'outil CLIPS puis construire l'interface de l'application pour faciliter l'utilisation de l'expert.
5. Programmation de ce système expert en java, cela pour avoir un logiciel qui ne dépend pas de clips.

V.2 Description de la base de donnée

La confection de la base de donnée peut prendre des mois voir des années pour avoir plus de patients et par conséquent valider les résultats, donc nous avons opté pour une base de donnée qui est destinée aux systèmes experts parmi les bases de données UCI.(19)

Les données a été créées par un expert médical comme un ensemble de données pour tester le système expert, qui effectuera le diagnostic présomptif de deux maladies du système urinaire. Elle est composée de 120 instances et 6 Attributs, elle n'a aucune valeur manquante et elle a été récoltée en 11/02/2009

Les données sont dans un fichier ASCII. Les attributs sont séparés par tabulation. chaque ligne du fichier de données commence par un chiffre qui indique la température du patient.(19)

Lignes d'attributs :

Par exemple : « 35, 9 non non oui oui oui oui non », Où :

- « 35,9 »Température de patients ;
- « non »Présence des nausées ;
- « non »la douleur lombaire ;
- « oui »urine poussée (besoin continu de la miction) ;
- « oui »douleurs miction ;

- « oui »incendie de l'urètre, les démangeaisons, le gonflement de la sortie de l'urètre ;
- « oui »la décision : inflammation de la vessie ;
- « non »la décision : néphrite rénale de l'origine du bassin.

V.2.1 Renseignements sur les attributs

Attribut	Significaton
a1	Température de patients (35C-42C)
a2	Présence des nausées (yes, no)
a3	la douleur lombaire (yes, no)
a4	urine poussée (besoin continu de la miction) (yes, no)
a5	douleurs miction (yes, no)
a6	incendie de l'urètre, les démangeaisons, le gonflement de la sortie de l'urètre (yes, no)

Tableau 1 – Attributs de la base de données

Decision	Significaton
d1	Inflammation de la vessie (yes, no)
d2	néphrite d'origine bassin (yes, no)

Tableau 2 – Décisions de la base de données

V.3 Conception de l'application

Nous présentons dans cette partie, les trois phases essentielles pour avoir une bonne conception d'une interface graphique qui représente le Système Expert :

1. **Analyse** : cette étape nécessite une étude détaillée du système a réaliser. tous les objectifs, exigences et contraintes doivent être justifiés et discutés en profondeur.
2. **Conception** : une fois la première tâche est accomplie, le passage du modèle logique à la description technique représente l'objectif de cette phase. Différents points sont pris en considération pour la réalisation de cette tâche :
 - Répondre aux besoins d'utilisateur,
 - la prise en compte du coté cognitive,
 - La rédaction d'un éventuel prototype. Reste, le crayon et le papier un excelent moyen d'explorer pour la conception d'une interface graphique(20).
3. **Implémentation** : cette phase comprend le codage de l'application.

V.4 Générer les règles

V.4.1 1ère technique : La logique Floue

Dans cette partie, on a généré les règles avec la logique Floue, cela pour avoir des résultats interprétables en utilisant l'outil FISPRO.

Vu qu'on a deux sorties (deux classes) dans notre base de donnée, on a divisé la base en 2. Toutes les deux possèdent : 6 entrée, 1 sortie et 18 règles.

Les entrées sont les attributs (les symptômes), la sortie est la maladie (la décision). Les règles font la décision en fonction des symptômes.

Inférer :

L'option Inférer du menu SIF permet d'explorer le fonctionnement du système. Le raisonnement à base de règles floues est représenté graphiquement.

Les valeurs des variables d'entrée sont saisies manuellement (déplacement du curseur ou saisie de valeur). Le système affiche la valeur inférée pour la sortie, avec plusieurs informations intermédiaires permettant de comprendre les étapes du raisonnement flou :

Pour chaque règle : un degré d'appartenance de la valeur à chaque SEF présent dans les prémisses de la règle, visualisé comme un taux de remplissage du SEF. Il est visualisé sous forme de valeur numérique car la sortie est nette.

Pour inférer on a passé par plusieurs étapes :

1ère étape : Charger la base de données

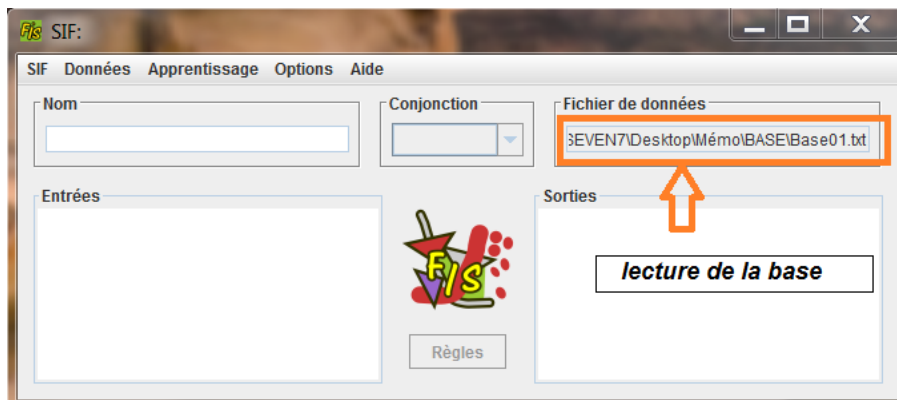


Figure 20 – Charger la base de donnée

2ème et 3ème étape :initialisation et génération d'un SIF sans règles

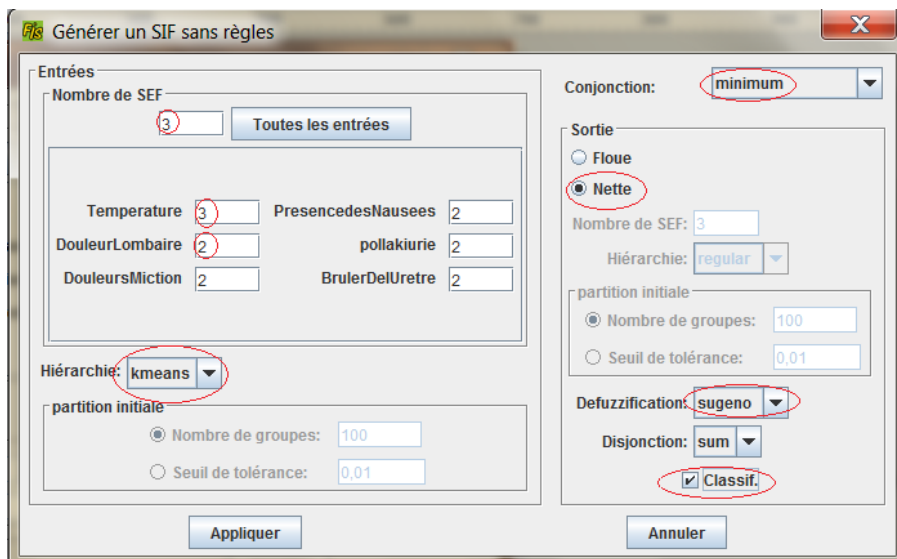


Figure 21 – générer un SIF sans règles

Choix des paramètres

Dans cette partie, nous avons choisi les différents degrés d'appartenance de chaque attribut. Pour l'attribut « Température » on a 3 degrés d'appartenance qui sont basse, normal et élevée, cela selon l'avis d'un expert. (voir la figure 23)

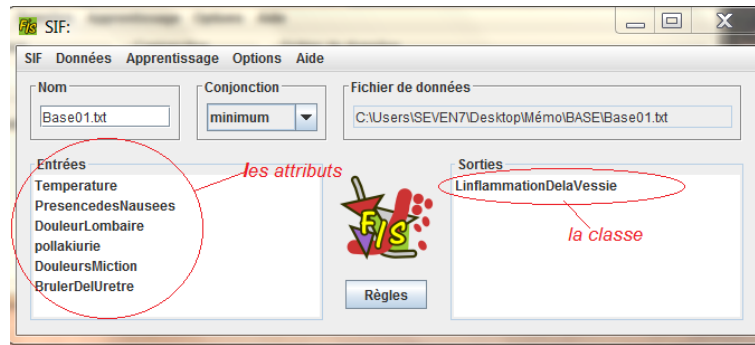


Figure 22 – Génération d'un SIF sans règles :

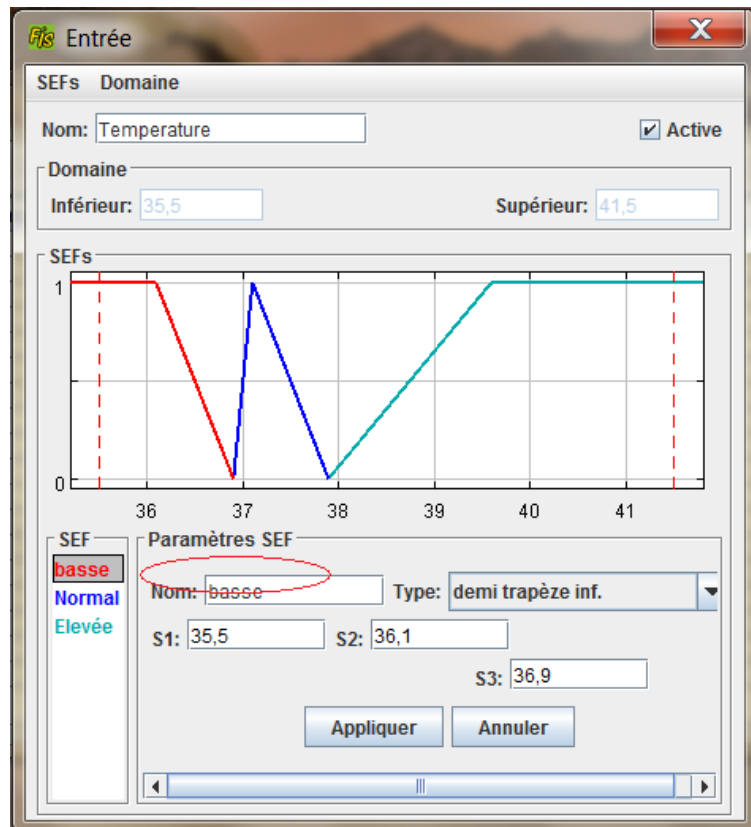


Figure 23 – les degrés d'appartenance de « Température »

Et pour les attributs binaires on a affecté le 0 pour le *Non* et le 1 pour le *Oui* (voir la figure 24).

Génération de règles : Dans cette partie, on a lancé la génération des règles (voir la figure 25).

Génération de conclusion : dans la figure 26

V.4.2 Les règles obtenues :

Nous présentons dans la figure 27 les règles obtenues de la base de cystite et dans la figure 28 les règles obtenues de la base de néphrite .

Analyse des résultats

Les résultats obtenus par FISPRO restent toujours pauvre car elle traite que les cas présents dans notre base de données pour cette raison on cherche d'enrichir notre resultat par une autre technique

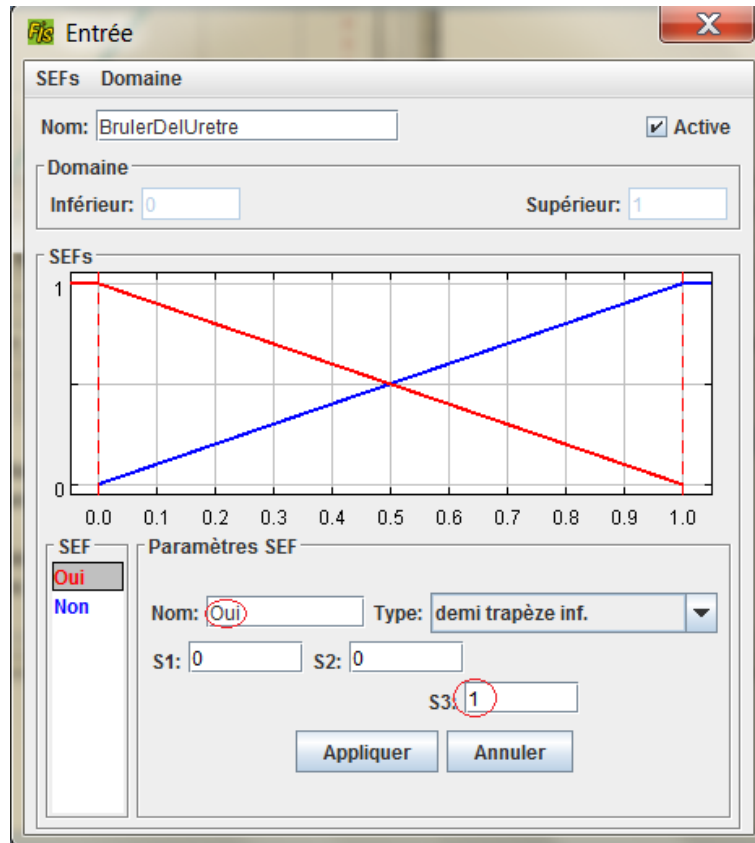


Figure 24 – affecter 1 pour « oui »

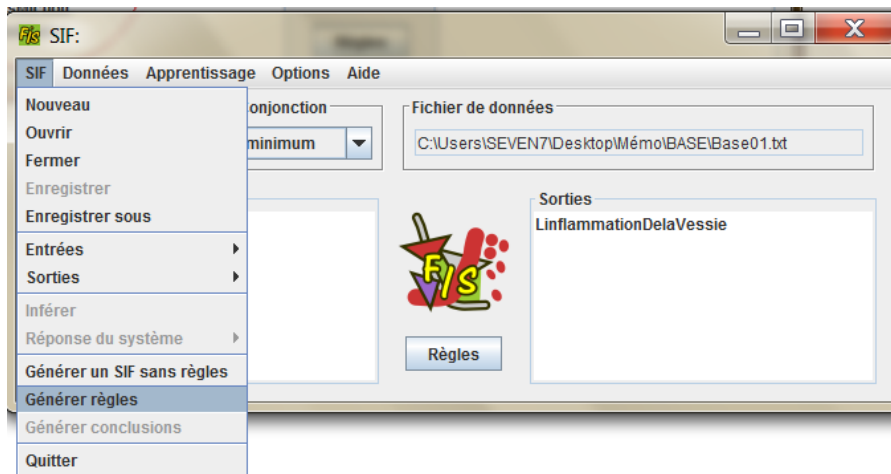


Figure 25 – générer les règles

qui est le Neuro-Flou (ANFIS).

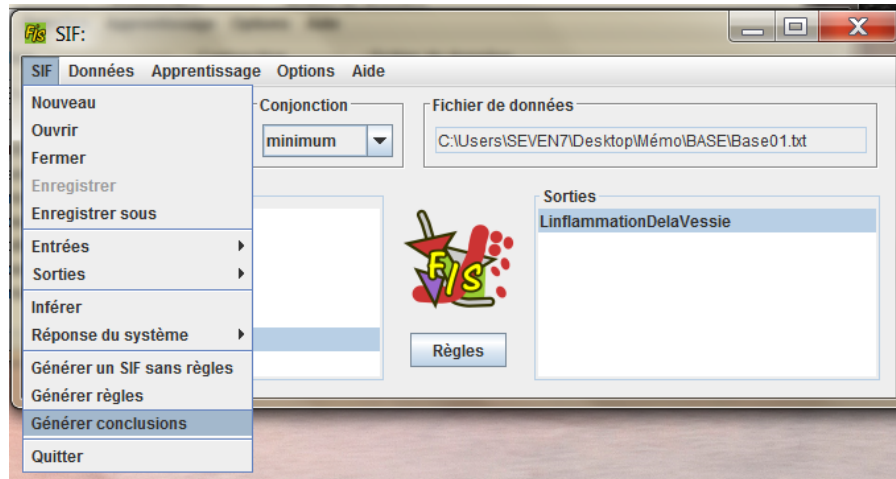


Figure 26 – Génération de conclusion

The screenshot shows the 'Règles' window with a table of 18 rules. The table has columns for 'Règle', 'Active', 'SI Temperature', 'ET PresencedesNaus...', 'ET DouleurLombaire', 'ET pollakiurie', 'ET DouleursMiction', 'ET BrulerDelUretre', and 'ALORS LinflammationDelaVessie'.

Règle	Active	SI Temperature	ET PresencedesNaus...	ET DouleurLombaire	ET pollakiurie	ET DouleursMiction	ET BrulerDelUretre	ALORS LinflammationDelaVessie
1	<input checked="" type="checkbox"/>	Basse	Non	Non	Oui	Non	Non	1
2	<input checked="" type="checkbox"/>	Basse	Non	Non	Oui	Oui	Non	1
3	<input checked="" type="checkbox"/>	Basse	Non	Non	Oui	Oui	Oui	1
4	<input checked="" type="checkbox"/>	Basse	Non	Oui	Non	Non	Non	0
5	<input checked="" type="checkbox"/>	Normal	Non	Non	Non	Non	Non	0
6	<input checked="" type="checkbox"/>	Normal	Non	Non	Oui	Non	Non	1
7	<input checked="" type="checkbox"/>	Normal	Non	Non	Oui	Oui	Non	1
8	<input checked="" type="checkbox"/>	Normal	Non	Non	Oui	Oui	Oui	1
9	<input checked="" type="checkbox"/>	Normal	Non	Oui	Non	Non	Non	0
10	<input checked="" type="checkbox"/>	Normal	Non	Oui	Oui	Non	Oui	0
11	<input checked="" type="checkbox"/>	Normal	Oui	Oui	Non	Oui	Non	0
12	<input checked="" type="checkbox"/>	Normal	Oui	Oui	Oui	Oui	Non	1
13	<input checked="" type="checkbox"/>	Normal	Oui	Oui	Oui	Oui	Oui	1
14	<input checked="" type="checkbox"/>	Elevee	Non	Non	Non	Non	Non	0
15	<input checked="" type="checkbox"/>	Elevee	Non	Oui	Oui	Non	Oui	0
16	<input checked="" type="checkbox"/>	Elevee	Oui	Oui	Non	Oui	Non	0
17	<input checked="" type="checkbox"/>	Elevee	Oui	Oui	Oui	Oui	Non	1
18	<input checked="" type="checkbox"/>	Elevee	Oui	Oui	Oui	Oui	Oui	1

Figure 27 – Base de cystite

The screenshot shows the 'Règles' window with a table of 18 rules. The table has columns for 'Règle', 'Active', 'SI Temperature', 'ET PresencedesNaus...', 'ET DouleurLombaire', 'ET pollakiurie', 'ET DouleursMiction', 'ET BrulerDelUretre', and 'ALORS NephritedOrigineBassinnet'.

Règle	Active	SI Temperature	ET PresencedesNaus...	ET DouleurLombaire	ET pollakiurie	ET DouleursMiction	ET BrulerDelUretre	ALORS NephritedOrigineBassinnet
1	<input checked="" type="checkbox"/>	Basse	Non	Non	Oui	Non	Non	0
2	<input checked="" type="checkbox"/>	Basse	Non	Non	Oui	Oui	Non	0
3	<input checked="" type="checkbox"/>	Basse	Non	Non	Oui	Oui	Oui	0
4	<input checked="" type="checkbox"/>	Basse	Non	Oui	Non	Non	Non	0
5	<input checked="" type="checkbox"/>	Normal	Non	Non	Non	Non	Non	0
6	<input checked="" type="checkbox"/>	Normal	Non	Non	Oui	Non	Non	0
7	<input checked="" type="checkbox"/>	Normal	Non	Non	Oui	Oui	Non	0
8	<input checked="" type="checkbox"/>	Normal	Non	Non	Oui	Oui	Oui	0
9	<input checked="" type="checkbox"/>	Normal	Non	Oui	Non	Non	Non	0
10	<input checked="" type="checkbox"/>	Normal	Non	Oui	Oui	Non	Oui	1
11	<input checked="" type="checkbox"/>	Normal	Oui	Oui	Non	Oui	Non	1
12	<input checked="" type="checkbox"/>	Normal	Oui	Oui	Oui	Oui	Non	1
13	<input checked="" type="checkbox"/>	Normal	Oui	Oui	Oui	Oui	Oui	1
14	<input checked="" type="checkbox"/>	Elevee	Non	Non	Non	Non	Non	0
15	<input checked="" type="checkbox"/>	Elevee	Non	Oui	Oui	Non	Oui	1
16	<input checked="" type="checkbox"/>	Elevee	Oui	Oui	Non	Oui	Non	1
17	<input checked="" type="checkbox"/>	Elevee	Oui	Oui	Oui	Oui	Non	1
18	<input checked="" type="checkbox"/>	Elevee	Oui	Oui	Oui	Oui	Oui	1

Figure 28 – Base de néphrite

V.4.3 2ème technique : ANFIS

Notant que pour un medecin (un Expert), 18 règles ne sont pas suffisantes pour traiter tous les cas possibles, pour cela dans cette partie on a implémenté un modèle Neuro-Flou pour resoudre cette problématique. On a divisé la base de données en deux, les deux ont : six attributs et une sortie.

Choix des paramètres :

Nous avons discuté avec un expert pour fixer les fonctions d'appartenance pour chaque attribut, on a opté pour les configurations $(3*2*2*2*2*2)$ et $(4*2*2*2*2*2)$, le tableau ci dessus montre la

Configuration	Nombre-Règles	Erreur-Apprentissage
3*2*2*2*2*2	96	2.162645e-007
4*2*2*2*2*2	128	2.55e-3

Tableau 3 – Erreur pour différentes configurations

Configuration	Méthode	Erreur-Apprentissage
3*2*2*2*2*2	Hybride	2.162645e-007
3*2*2*2*2*2	Backpropa	2.21723e-3

Tableau 4 – Différence entre les deux méthodes

différence entre les deux configurations :

Avant lancer l'apprentissage, on a opté la première configuration car elle a l'erreur d'apprentissage minimal et pour un expert la température peut se présenté sous trois fonctions d'appartenance. On lance l'apprentissage en utilisant :

- deux méthodes (algorithmes) :
 1. Rétro-propagation
 2. Hybride (rétro-propagation et moindre carré)
- on fixe le seuil d'erreur à 0.001
- on fixe trois itérations pour l'apprentissage

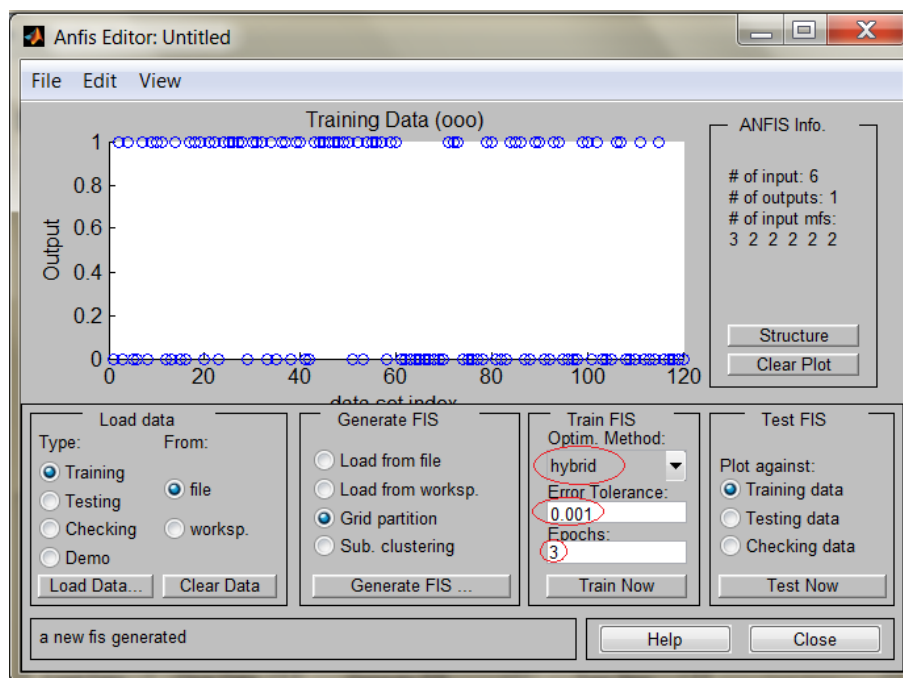


Figure 29 – Lancement d'apprentissage

En comparant ces deux méthodes dans le tableau ci dessous.

Remarque :

La méthode hybride donne des résultats meilleurs que celle de backpropa (retro- propagation) car celle de hybride à l'erreur d'apprentissage moins que celle de backpropa.

Configuration	NmbrIndividus	Méthode	TauxCl%Cys	TauxCl%Ne	TauxEr%Cys	TauxEr%Ne
3*2*2*2*2*2	120	Hybride	100	92.5	0	7.5
3*2*2*2*2*2	120	Backpropa	100	92.5	0	7.5

Tableau 5 – Différence entre les deux méthodes

Nous avons opté pour l'attribut « Température » sur 3 degrés d'appartenance (basse, normal, élevée), cela selon un expert.(voir la figure 30)

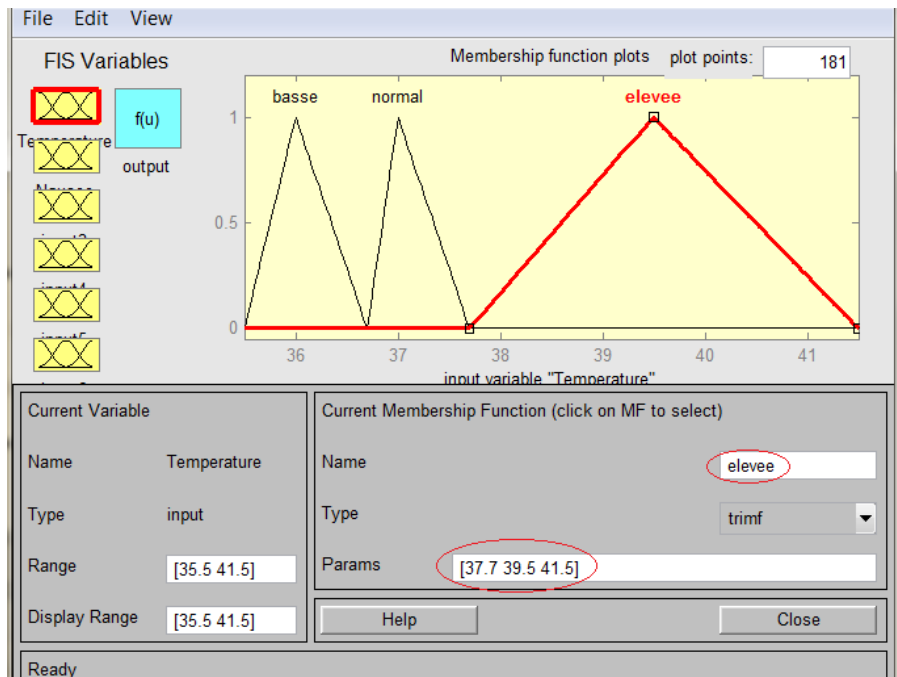


Figure 30 – les fonctions d'appartenance de « Température »

Et pour les attributs binaires on a affecté le 0 pour le *Non* et le 1 pour le *Oui*. Et pour la sortie on a affecté le 0 pour le *Non* et le 1 pour le *Oui* (voir la figure 31).

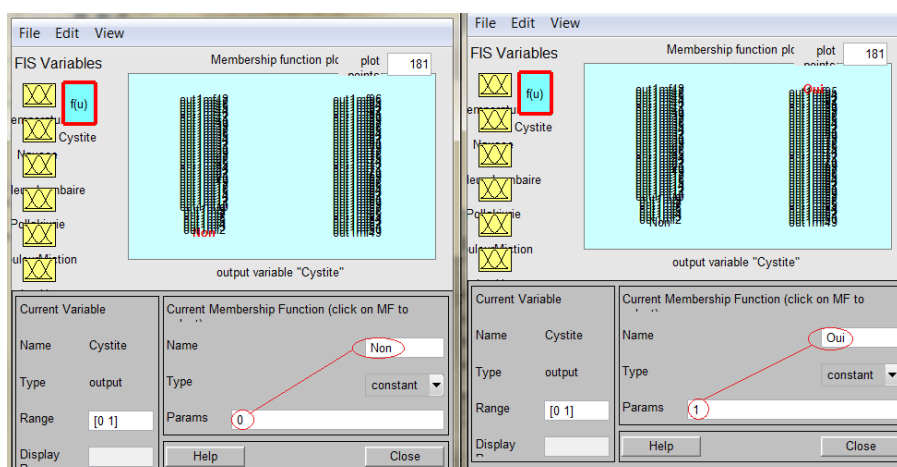


Figure 31 – Génération de la sortie

V.4.4 Structure de notre modèle

nous présentons dans la figure 32 la structure neuro-flou de notre modèle, dans la figure 33 la structure FIS de la base de Cystite et dans la figure 34 la structure FIS de la base de nephrite.

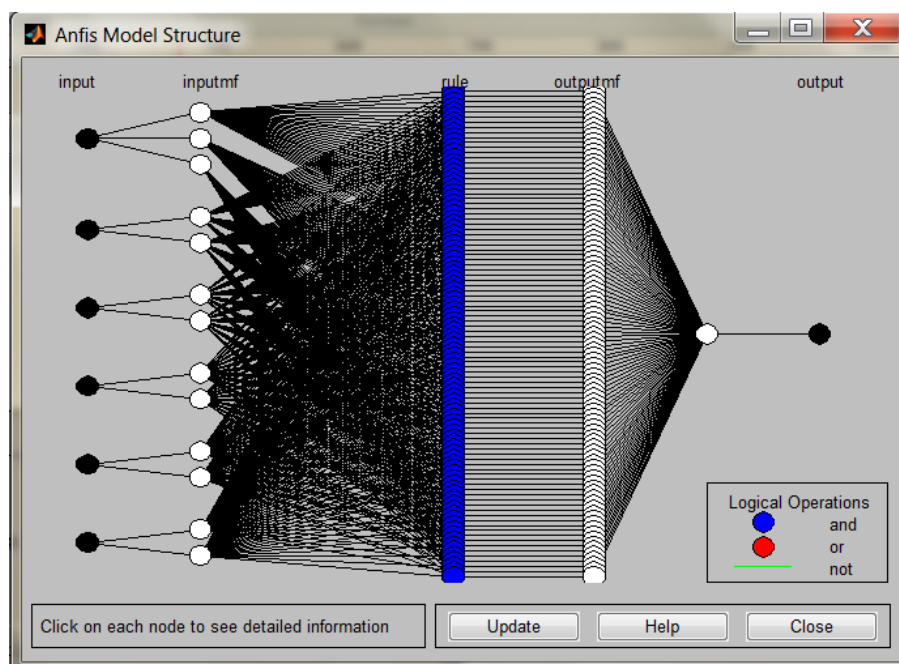


Figure 32 – Neuro-Flou

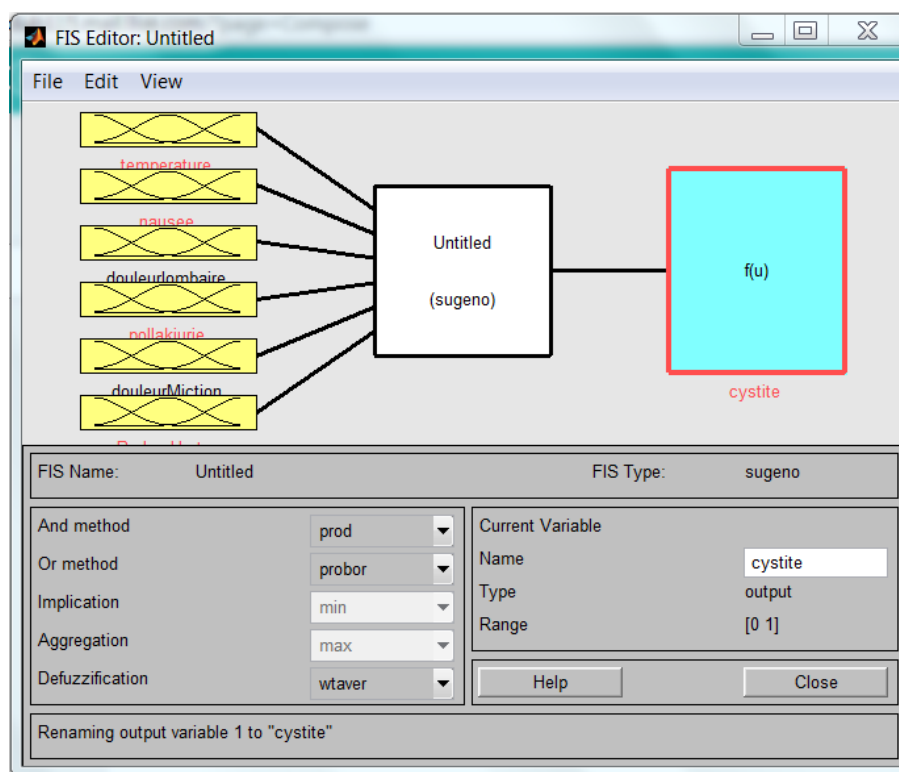


Figure 33 – FIS de la cystite

V.5 Les résultats obtenus

Les résultats obtenus sont les 96 règles générées par l'ANFIS. vous trouverez ces 96 règles en Annexe.

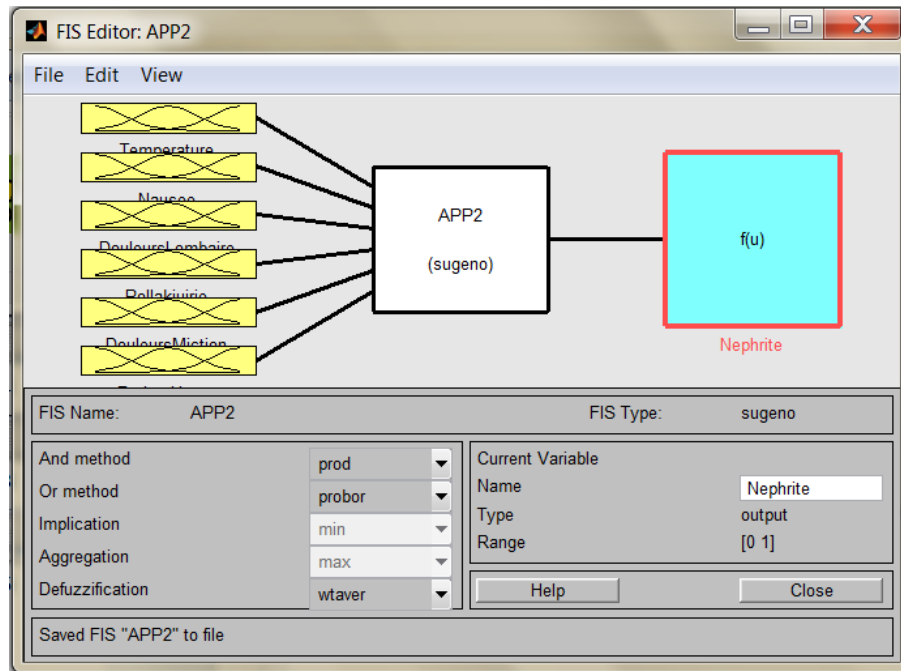


Figure 34 – FIS de la Nephrite

V.5.1 Evaluation des resultats :

les résultats de TEST obtenus :

Pour tester l'apprentissage de notre Système, nous avons lancer le test sur la même base d'apprentissage, les tableaux ci dessous montre cela :

NbrIndividus	TauxClassif%	TauxErreur%	SP%	SE%	VP%	Vn%	FP %	FN %
120	100	0	100	100	100	100	NAN	0

Tableau 6 – Resultat de TEST de la base de cystite

NbrIndividus	TauxClassif%	TauxErreur%	SP%	SE%	VP%	Vn%	FP %	FN %
120	92.5	7.50	100	87.14	100	84.75	NAN	12.86

Tableau 7 – Résultat de TEST de la base de néphrite

Dans cette partie, pour tester on a créé une autre base de données de 40 patients qui contient des cas qui n'existent pas dans la base de données d'apprentissage et des cas qui existent, cela pour tester l'apprentissage.

NbrIndividus	TauxClassif%	TauxErreur%	SP%	SE%	VP%	Vn%	FP %	FN %
40	95	5	100	91.30	100	89.47	NAN	8.70

Tableau 8 – Resultat de TEST de la base de 40 de cystite

V.6 La conception du système expert en CLIPS

Dans cette partie, nous avons programmer notre système expert en Clips, et utiliser les règles obtenus pour notre conception.

NbrIndividus	TauxClassif%	TauxErreur%	SP%	SE%	VP%	Vn%	FP %	FN %
40	95	5	100	91.30	100	89.47	NAN	8.70

Tableau 9 – Résultat de TEST de la base de 40 de néphrite

Notre Système expert sera présenté sous formes « Question-Réponse », c'est une aide au diagnostic aux médecins pour les différents cas d'infections urinaires, il lui permet de savoir si le patient est malade ou non a partir de ses différents symptômes et sa proposition de la décision. L'outil le plus connu pour la conception des systèmes expert est clips :

Nous avons utiliser Clips pour notre conception du système expert, il commence par la lecture de la base des règles qu'on a construit en exécutant cette dernière, la figure 35 montre cela.

```

Dialog Window
Defining defrule: determinee-BrulureUret30 +j+j
Defining defrule: determinee-BrulureUret31 +j+j
Defining defrule: determinee-BrulureUret32 +j+j
Defining defrule: determinee-BrulureUret33 +j+j
Defining defrule: determinee-BrulureUret34 +j+j
Defining defrule: determinee-BrulureUret35 +j+j
Defining defrule: determinee-BrulureUret36 +j+j
Defining defrule: determinee-BrulureUret37 +j+j
Defining defrule: determinee-BrulureUret38 +j+j
Defining defrule: determinee-BrulureUret39 +j+j
Defining defrule: determinee-BrulureUret40 +j+j
Defining defrule: determinee-BrulureUret41 +j+j
Defining defrule: determinee-BrulureUret42 +j+j
Defining defrule: determinee-BrulureUret43 +j+j
Defining defrule: determinee-BrulureUret44 +j+j
Defining defrule: determinee-BrulureUret45 +j+j
Defining defrule: determinee-BrulureUret46 +j+j
Defining defrule: determinee-BrulureUret47 +j+j
Defining defrule: determinee-BrulureUret48 +j+j
TRUE
CLIPS> (run)
*****
*Auteur: Stambouli et Moussaoui *
*****
Tapez la touche ENTREE pour continuer ...
    
```

Figure 35 – lecture de la base des règles

Nous avons construit notre base de connaissance sous forme de questionnaire pour aider le médecin d'avoir une décision à partir des symptômes cliniques(voir la figure 36).

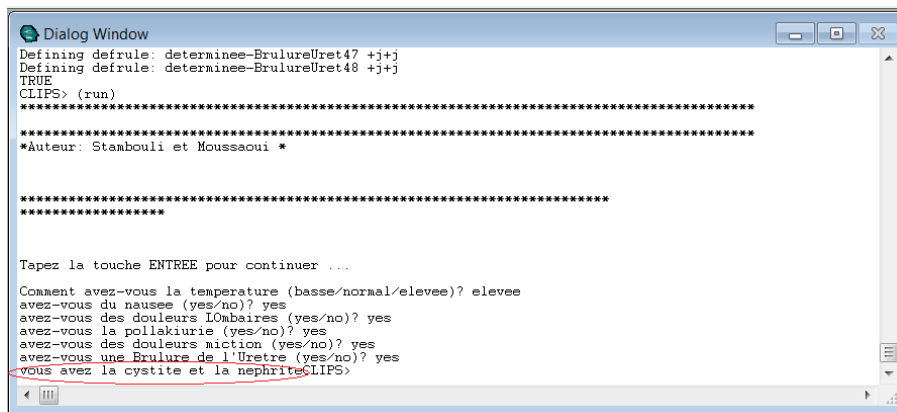
```

CLIPS 6.3
File Edit Buffer Execution Browse Window Help
Dialog Window
Defining defrule: determinee-BrilureUret41 +j+j
Defining defrule: determinee-BrilureUret42 +j+j
Defining defrule: determinee-BrilureUret43 +j+j
Defining defrule: determinee-BrilureUret44 +j+j
Defining defrule: determinee-BrilureUret45 +j+j
Defining defrule: determinee-BrilureUret46 +j+j
Defining defrule: determinee-BrilureUret47 +j+j
Defining defrule: determinee-BrilureUret48 +j+j
TRUE
CLIPS> (run)
*****
*Auteur: Stambouli et Moussaoui *
*****
Tapez la touche ENTREE pour continuer ...
Comment avez-vous la temperature (basse/normal/elevee)?
    
```

Figure 36 – le Questionnaire

On peut avoir quatre décisions : « vous avez la cystite et la néphrite », « vous n'avez ni la cystite

ni la néphrite », « vous avez la cystite mais vous n'avez pas la néphrite » et « vous avez la néphrite mais vous n'avez pas la cystite » (voir les figures 37,38,39,40).



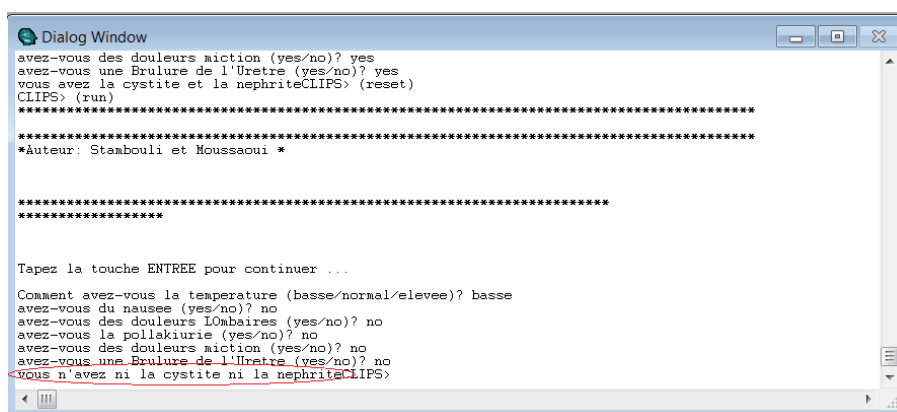
```

Dialog Window
Defining defrule: determinee-BrulureUret47 +j+j
Defining defrule: determinee-BrulureUret48 +j+j
TRUE
CLIPS> (run)
*****
*Auteur: Stambouli et Moussaoui *
*****

Tapez la touche ENTREE pour continuer ...

Comment avez-vous la temperature (basse/normal/elevee)? elevee
avez-vous du nausée (yes/no)? yes
avez-vous des douleurs LOmbaires (yes/no)? yes
avez-vous la pollakiurie (yes/no)? yes
avez-vous des douleurs miction (yes/no)? yes
avez-vous une Brulure de l'Uretre (yes/no)? yes
vous avez la cystite et la nephriteCLIPS>
    
```

Figure 37 – décision1 : Vous avez la cystite et la néphrite



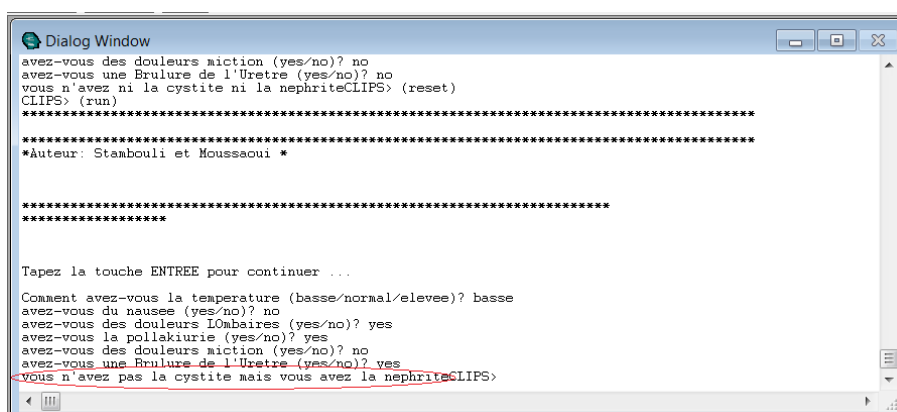
```

Dialog Window
avez-vous des douleurs miction (yes/no)? yes
avez-vous une Brulure de l'Uretre (yes/no)? yes
vous avez la cystite et la nephriteCLIPS> (reset)
CLIPS> (run)
*****
*Auteur: Stambouli et Moussaoui *
*****

Tapez la touche ENTREE pour continuer ...

Comment avez-vous la temperature (basse/normal/elevee)? basse
avez-vous du nausée (yes/no)? no
avez-vous des douleurs LOmbaires (yes/no)? no
avez-vous la pollakiurie (yes/no)? no
avez-vous des douleurs miction (yes/no)? no
avez-vous une Brulure de l'Uretre (yes/no)? no
vous n'avez ni la cystite ni la nephriteCLIPS>
    
```

Figure 38 – décision2 : vous n'avez ni la cystite ni la néphrite



```

Dialog Window
avez-vous des douleurs miction (yes/no)? no
avez-vous une Brulure de l'Uretre (yes/no)? no
vous n'avez ni la cystite ni la nephriteCLIPS> (reset)
CLIPS> (run)
*****
*Auteur: Stambouli et Moussaoui *
*****

Tapez la touche ENTREE pour continuer ...

Comment avez-vous la temperature (basse/normal/elevee)? basse
avez-vous du nausée (yes/no)? no
avez-vous des douleurs LOmbaires (yes/no)? yes
avez-vous la pollakiurie (yes/no)? yes
avez-vous des douleurs miction (yes/no)? no
avez-vous une Brulure de l'Uretre (yes/no)? yes
vous n'avez pas la cystite mais vous avez la nephriteCLIPS>
    
```

Figure 39 – décision3 : vous n'avez pas la cystite mais vous avez la néphrite

V.7 L'intégration du système expert en JAVA

Nous avons intégré le SE CLIPS en JAVA pour avoir une interface graphique à l'aide d'une librairie JESS :(voir les figures 41 et 42)

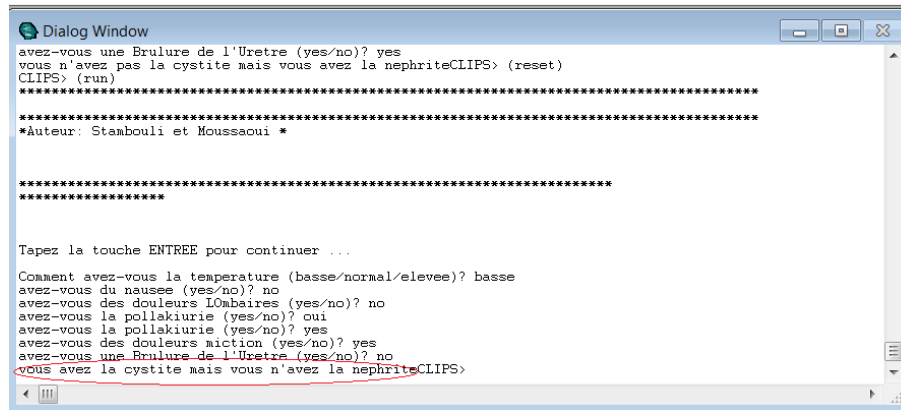


Figure 40 – décision4 : vous avez la cystite mais vous n'avez pas la néphrite

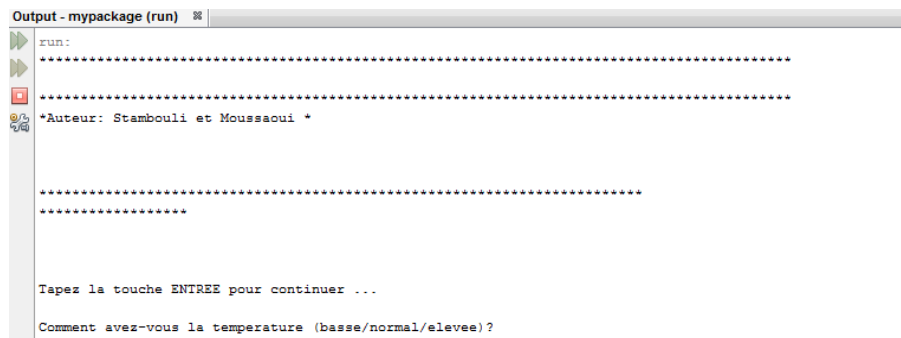


Figure 41 – lire la base de connaissance

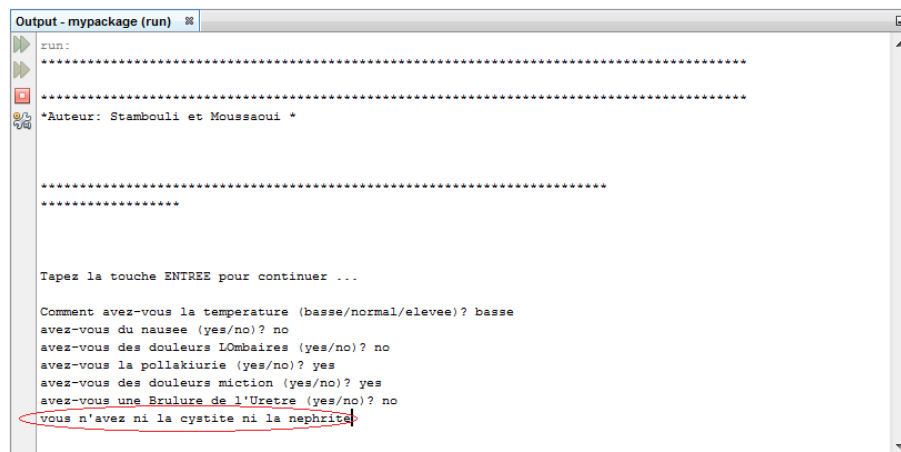


Figure 42 – Répondre aux questions pour avoir la décision

V.8 Conception du Système

Nous avons opté pour la programmation en JAVA comme deuxième alternative ; tout simplement parce que le package n'est pas libre.

Nous avons programmé notre base de connaissance en java pour avoir une interface graphique qui facilite l'utilisation aux médecins et urologues.

Nous avons construit un logiciel qui est exécutable et facile à manipuler. C'est un système d'aide au diagnostic interactif, fiable et flexible, il contient deux phases :

1. **La phase de la gestion des patients** : le médecin ou l'urologue peut ajouter, supprimer, modifier et rechercher un patient. Cela pour enrichir la base de données.

2. **La phase du diagnostic** : le medecin ou l'urologue peut entrer les symptômes du patient pour qu'il obtient la décision. S'il n'est pas satisfait de cette décision, il peut proposer une autre décision, cela permet au système d'apprendre de ses erreurs. comme il peut créer des nouveaux cas.

Nous présentons dans la figure 43 un diagramme de cas d'utilisations qui montre cela .

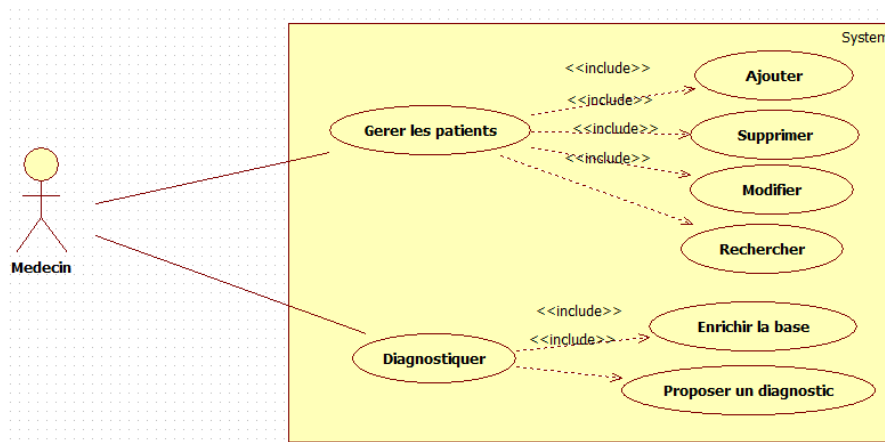


Figure 43 – Diagramme de cas d'utilisation

Le logiciel est présenté d'abord avec une interface graphique sécurisée, il demande à l'utilisateur de rentrer le nom d'utilisateur et le mot de passe. s'il y'a une erreur quelque part, il affiche cela (voir la figure 44)



Figure 44 – 1ère interface d'INFUR

Après avoir entrer le nom d'utilisateur et le mot de passe, il va afficher une autre interface qui permet de choisir entre faire le diagnostic a un patient ou gérer les patients.(voir la figure 45)

1. Si l'utilisateur choisit de gérer les patients pour ajouter, il permet de rentrer le informations du patient comme son nom et prénom, sa date de naissance et ses symptomes ainsi que sa maladie, pour supprimer, modifier ou rechercher un patient a partir de son nom et prénom, nous présentons ces différentes tâches dans la figure 46
2. Si l'utilisateur choisit d'ajouter des nouveaux cas pour les diagnostiquer, il affiche une interface qui lui facilite cette tâche, elle lui permet d'entrer les différents informations du patient ainsi que ses symptômes et clique sur le bouton « tester » pour avoir la décision générée par notre système. S'il a des propositions a propos de la décision, il peut ajouter sa proposition, cela permet au système d'apprendre de ses erreurs , la figure 47 montre ceci.



Figure 45 – accueil

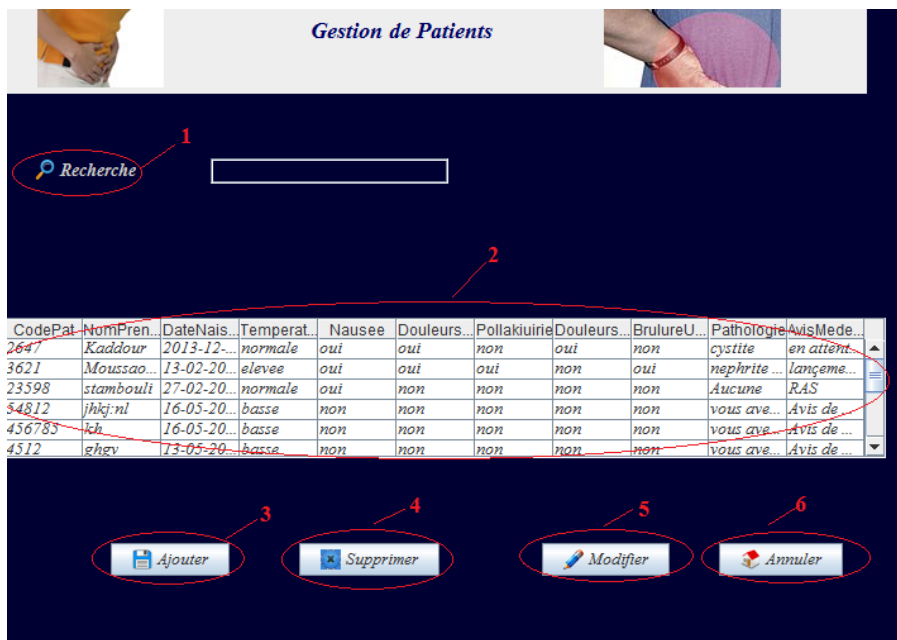


Figure 46 – Gestion des patients



Figure 47 – L'interface de diagnostiquer

Conclusion générale et Perspectives

A travers ce mémoire, nous avons pu voir que l'intelligence artificielle est une discipline qui ne cesse d'évoluer depuis déjà la moitié du 20ème siècle jusqu'à présent et il n'est pas prêt de s'arrêter. Nous avons constaté qu'il est présent dans différents domaines (celui de la santé, de l'informatique, de l'industrie...). Actuellement, il est capable de réaliser de nombreuses choses comme opérer des patients, assembler des pièces pour aboutir à la construction d'automobiles..

Dans ce modeste travail, nous avons développé différentes techniques intelligentes pour les implémenter dans notre système INFUR afin d'aider les médecins et les urologues à la détection précoce des maladies des infections urinaires.

En premier lieu, nous avons implémenté une interface basée sur la génération automatique des règles qui ont été générées par le neuro-Flou. D'après les résultats présentés dans le dernier chapitre, cette technique a prouvé ses capacités de résoudre cette problématique, les résultats sont plus interprétable et compréhensible par l'expert humain. Les résultats obtenus par la technique utilisée ont été assez satisfaisantes.

Nous avons intégré les règles obtenus en CLIPS dans java pour une interface graphique en utilisant la librairie JESS mais malheureusement cette librairie n'est pas libre alors nous avons développé une application pour avoir un système expert gratuit, ayant comme finalité la gestion des patients et le diagnostic de ses patients en entrant leurs symptômes et la proposition de l'expert pour que le système apprend de ses erreurs.

Finalement, ce travail constitue un premier pas vers un système expert d'aide au diagnostic dans le domaine médical. Il peut être enrichi dans le futur en intégrant d'autres modules qui seront utiles pour une meilleure décision médicale comme :

- Vulgariser ce système (il sera destiné au grand public).
- Donner la possibilité de faire l'apprentissage en ligne.
- Enrichir le système pour qu'il traite plusieurs maladies et donne leurs traitements.

88. **Règle 88 :**
If (Temperature is elevee) and (Nausee is Oui) and (DouleursLombaire is Non) and (pollakiurie is Oui) and (DouleursMiction is Oui) and (BrulureUretre is Oui) then (Cystite is Non) and (Nephrite is Non)
89. **Règle 89 :**
If (Temperature is elevee) and (Nausee is Oui) and (DouleursLombaire is Oui) and (pollakiurie is Non) and (DouleursMiction is Non) and (BrulureUretre is Non) then (Cystite is Non) and (Nephrite is Non)
90. **Règle 90 :**
If (Temperature is elevee) and (Nausee is Oui) and (DouleursLombaire is Oui) and (pollakiurie is Non) and (DouleursMiction is Non) and (BrulureUretre is Oui) then (Cystite is Non) and (Nephrite is Non)
91. **Règle 91 :**
If (Temperature is elevee) and (Nausee is Oui) and (DouleursLombaire is Oui) and (pollakiurie is Non) and (DouleursMiction is Oui) and (BrulureUretre is Non) then (Cystite is Non) and (Nephrite is Oui)
92. **Règle 92 :**
If (Temperature is elevee) and (Nausee is Oui) and (DouleursLombaire is Oui) and (pollakiurie is Non) and (DouleursMiction is Oui) and (BrulureUretre is Oui) then (Cystite is Non) and (Nephrite is Non)
93. **Règle 93 :**
If (Temperature is elevee) and (Nausee is Oui) and (DouleursLombaire is Oui) and (pollakiurie is Oui) and (DouleursMiction is Non) and (BrulureUretre is Non) then (Cystite is Non) and (Nephrite is Non)
94. **Règle 94 :**
If (Temperature is elevee) and (Nausee is Oui) and (DouleursLombaire is Oui) and (pollakiurie is Oui) and (DouleursMiction is Non) and (BrulureUretre is Oui) then (Cystite is Non) and (Nephrite is Non)
95. **Règle 95 :**
If (Temperature is elevee) and (Nausee is Oui) and (DouleursLombaire is Oui) and (pollakiurie is Oui) and (DouleursMiction is Oui) and (BrulureUretre is Non) then (Cystite is Oui) and (Nephrite is Oui)
96. **Règle 96 :**
If (Temperature is elevee) and (Nausee is Oui) and (DouleursLombaire is Oui) and (pollakiurie is Oui) and (DouleursMiction is Oui) and (BrulureUretre is Oui) then (Cystite is Oui) and (Nephrite is Oui)

Références

- [1] Jean-Paul Haton and Marie-Christine Haton. *L'intelligence artificielle*. Presses universitaires de France, 1989.
- [2] Stuart Russell et Peter Norvig. *Artificial intelligence : A modern approach*.
- [3] Intelligence artificielle. url = <http://www.world.honda.com/ASIMO/>.
- [4] Acute inflammations. url = <http://archive.ics.uci.edu/ml/machine-learning-databases/acute/diagnosis.names>.
- [5] Inflammation de la vessie, cystite. url = http://sante-web.ch/santeweb/Maladies/khb.php?Inflammation_dela_vessie_cystite_khb_ingid=2khh_cotent_id=2115.
- [6] Pyélonéphrite aiguë. <http://fr.wikipedia.org/wiki/Py>
- [7] Les infections urinaires. url = http://www.doctissimo.fr/html/sante/encyclopedie/sa520_infection_urinaire.htm
- [8] INTELLIGENCE ARTIFICIELLE. *Intelligence artificielle*. Teknea, 1988.
- [9] N.chenfour. *Intelligence artificielle et Système expert. chapitre5 : Systèmes Experts et Moteurs d'Inférence*. 2013.
- [10] Melle Fatima bekaddour. *La conception d'un Système au diagnostic Médical*. PhD thesis, 2010.
- [11] KANDE Kanumuambidi. *Réalisation d'un système expert pour le diagnostic et la thérapeutique des maladies de la tuberculose*. PhD thesis, Juillet 2010.
- [12] What is clips? url = <http://clipsrules.sourceforge.net/WhatIsCLIPS.html>.
- [13] El-Hadi Rebaa. *L'Université Paris XII Val de Marne*. PhD thesis, université Paris, 2003.
- [14] Le fonctionnement d'un système flou. url = <http://www.ferdinandpiette.com/blog/2011/08/les-systemes-flous-le-fonctionnement/>.
- [15] C.-T Sun. *Rulebase structure identification in an adaptive-network-based fuzzy inference system. IEEE Trans. Fuzzy Systems*. 1994.
- [16] U. Kayamak M. Setnes, R. Babuska and van Nauta. *Similarity measures in fuzzy rule base simplification. IEEE Transactions on Systems, Man and Cybernetics*. 1998.
- [17] Nesma Settouti. *Renforcement de l'apprentissage structurel pour la reconnaissance du diabete*. 2011.
- [18] Uci. url = <http://archive.ics.uci.edu/ml/>.
- [19] Patrick MARCHAND O.THOMAS HOLLAND. *Graphics and guis with matlab third edition*. 2003.

Conception et Réalisation d'un système expert médical INFUR

Résumé

De nos jours, les infections urinaires sont des maladies plus fréquentes chez les femmes et plus graves chez les hommes. L'aide au diagnostic dans ce cas précis consiste à proposer des décisions à partir de la description d'un cas clinique, ce dernier est obtenu à l'aide des signes issus de l'interrogatoire combinés aux examens cyto bactériologique.

Dans ce mémoire, nous avons construit un système expert appelé INFUR qui permet d'assister les médecins urologues de déceler d'éventuelles infections urinaires ; ce système est basé sur des modèles de l'I.A .

Nous avons réussi à implémenter un système évolutif qui apprend de ses erreurs et donne la possibilité d'enrichir la base de données à partir de nouveaux patients.

Les résultats obtenus indiquent qu' INFUR est un Système interactif, fiable et flexible d'aide au diagnostic.

Mots clés : Aide au diagnostic, Infection Urinaire, Système expert, Logique Floue, Neuro-Flou.

Abstract

Nowadays , urinary tract infections are more frequent diseases in women and more severe in men. The diagnostic aid in this case is to propose decisions from the description of a clinical case, the latter is obtained using the signs from the combined with cyto bacteriological examinations.

In this brief, we have built an expert system that allows INFUR called to assist urologists to detect any urinary infections ; This system is based on models of A.I .

We managed to implement a scalable system that learns from its mistakes and gives the opportunity to enrich the database from new patients.

The results obtained indicate that INFUR is for interactively , reliable and flexible diagnostic aid .

Keywords: computer assisted diagnostic, Urinary infections, expert System, fuzzy logic, neuro-fuzzy

ملخص:

في الوقت الحاضر، التهابات المسالك البولية هي أمراض أكثر تواترا عند النساء وأكثر شدة عند الرجال. مساعدة التشخيص في هذه الحالة هو اقتراح القرارات من وصف هذه الحالة السريرية، يتم الحصول هذه الأخيرة باستخدام دراسة موحدة للأعراض و الامتحانات الجرثومية.

في هذه المدكرة، قمنا ببناء نظام خبير يسمى أنفور الذي يسمح بمساعدة أطباء المسالك البولية للكشف عن أي التهابات المسالك البولية. ويستند هذا النظام على نماذج الذكاء الاصطناعي.

تمكنا من تنفيذ نظام قابل أن يتعلم من أخطائه ويعطي الفرصة لإثراء قاعدة البيانات من المرضى الجدد.

الكلمات المفاتيح: المساعدات في التشخيص الطبي، عدوى المسالك البولية، نظام

خبير، المنطق الضبابي، الشبكة العصبية الضبابية