Record Number:
Author, Monographic: Ouellet, M.//Tessier, J. C.
Author Role:
Title, Monographic: Intelligence artificielle et systèmes experts : principes et méthodes
Translated Title:
Reprint Status:
Edition:
Author, Subsidiary:
Author Role:
Place of Publication: Québec
Publisher Name: INRS-Eau
Date of Publication: 1987
Original Publication Date: Janvier 1987
Volume Identification:
Extent of Work: v, 334
Packaging Method: pages incluant 3 annexes
Series Editor:
Series Editor Role:
Series Title: INRS-Eau, Rapport de recherche
Series Volume ID: 226
Location/URL:
ISBN: 2-89146-223-8
Notes: Rapport annuel 1986-1987
Abstract: 50.00$
Call Number: R000226
Keywords: rapport/ ok/ dl

INRS-EAU

INTELLIGENCE ARTIFICIELLE ET SYSTEMES EXPERTS: PRINCIPES ET METHODES

TOME I
CHAPITRES I à VIII

RAPPORT SCIENTIFIQUE 226

Comme exigence partielle
aux études de
doctorat ès sciences (eau).

par

Jocelyn Ouellet
Etudiant au doctorat
INRS-EAU

En collaboration avec

Jean Claude Tessier
Etudes et Recherches écologiques
Direction Environnement
Hydro-Québec

Janvier 1987

TABLE DES MATIÈRES

TABLE DES MATIÈRES (suite)

TABLE DES MATIÈRES (suite)

LISTE DES FIGURES

PAGE

REMERCIEMENT

Je veux spécialement remercier Guy Lefebvre, étudiant en mathématique à l'Université de Montréal pour son travail de moine consacré à l'inventaire des systèmes experts que l'on retrouve en annexe 1, et à l'élaboration du chapitre 7 de la présente étude.

1.0 INTRODUCTION

On a donné le nom "Intelligence Artificielle" aux techniques de programmation qui imitent le comportement de l'intelligence. En fait, l'Intelligence Artificielle (I.A.) est une partie de la science informatique qui est concernée par le design de systèmes informatisés qui exhibent les caractéristiques que nous associons à l'intelligence humaine, comme par exemple le langage, l'apprentissage, le raisonnement, la résolution de problèmes, etc. L'I.A. est une science interdisciplinaire et revêt de plus en plus une importance pratique. Les mathématiciens, les philosophes, les psychologues, les linguistes et d'autres ont contribué par leurs modèles et leurs réflexions à la construction de cette science. Les mathématiques par la théorie sur la logique et l'analyse numérique; les psychologues par le développement de modèles de la pensée basés sur des concepts fondamentaux des systèmes symboliques de l'I.A., les linguistes par des programmes qui comprendront éventuellement le language, les philosophes pour leurs perspectives sur le problème du vieillissement de la connaissance et de la pensée.

Les domaines d'application aujourd'hui de l'I.A., sont de plus en plus variés: le diagnostic médical, la conduite processus industriel, l'interrogation de banques de données, le conseil de placement financier, le pilotage de véhicules autonomes, le design des puces, la chimie interprétative, le design de structure, la vision, la robotique, la reconnaissance de langage parlé et écrit, l'enseignement, etc.

L'I.A. prend son origine dans les années '30 et '40 par le rafinement de la théorie de la logique et des méthodes d'analyse numérique; l'ordinateur n'était pas encore né, mais celles-ci devaient en dicter les principes. La logique étant une méthode permettant de démontrer les théorèmes mathématiques. Les systèmes logico-déductifs ont été implantés avec succès sur ordinateur, et la conception abstraite de

l'analyse numérique est le traitement symbolique. Dès l'apparition de l'ordinateur, nous assistons à la naissance de la cybernitique. Le but était d'expliquer, à l'aide des mathématiques les phénomènes qui mettent en jeu les mécanismes du traitement de l'information. On y amena la participation de différents spécialistes: mathématiciens, psychologues, linguistes, etc. Le but ultime était d'en arriver à un modèle global de l'intelligence. On a échoué dans cette tentative, mais la recherche n'a pas été vaine; elle a donné naissance à l'intelligence artificielle et aux grandes orientations de celle-ci.

Les premiers programmes en I.A. vers la fin des années '50, ont privilégié une approche combinatoire c'est-à-dire qu'ils utilisaient la puissance de l'ordinateur pour examiner toutes les possibilités. Il y eu ensuite l'approche humaine, qui s'inspire de notre comportement avec l'apparition des robots (vers la fin des années '60). Le robot avait comme base de connaissances un modèle de l'univers. Il déterminait un plan pour exécuter les ordres, il était capable de prévoir les conséquences de ses actes, il exécutait le plan qu'il avait créé, et modifiait l'univers, s'il y a avait lieu, lors de l'exécution du plan. Les champs d'actions de ces premiers robots demeuraient quand même très limité. Le principal problème à résoudre se résume à la vision.

Vers le milieu des années 1970, on assiste à des progrès spectaculaires en I.A.: le traitement des langues naturelles, et l'avènement des systèmes experts. La traduction des langues naturelles passe par un intermédiaire celui de l'analyseur qui construit une représentation du sens du texte; par la suite un générateur à l'aide d'un dictionnaire informatisé reprend la représentation du sens pour reconstituer le langage. Pour ce qui est des systèmes experts, la caractéristique commune est la séparation entre la base de connaissance et le programme qui permet d'utiliser les connaissances soit le moteur d'inférence.

L'attrait que suscitent les systèmes experts par l'utilisation de connaissances spécialisées à travers une base de connaissance sont la reproduction du raisonnement humain.

Il semble raisonnable aujourd'hui de distinguer deux types de recherche en I.A., soit la recherche périphérique et la recherche centrale. La recherche centrale est celle qui s'intéresse à la déduction, au raisonnement du bon sens, au plan synthèse, à la compréhension du langage. La recherche périphérique étant celle concernée par la première étape du traitement des données c'est-à-dire celle des capteurs acoustiques, visuels, sensitifs, ceux de la reconnaissance et de la reconstitution de la parole.

La recherche centrale est particulièrement concernée par les formalismes de représentation de la connaissance, et par les techniques de manipulation de ces formalismes. Les mediums conventionnels de représentation des connaissances sont les langues et les notations mathématiques. L'étude de formalismes de représentation en I.A. a donné naissance à de nombreux langages, dont chacun plus ou moins possède la précision et l'interprétabilité (par ordinateur) d'une notation mathématique. La recherche en I.A. s'intéresse, non exclusivement, aux représentations pour lesquelles on peut donner une interprétation déclarative pour exprimer des connaissances non mathématiques.

La présente étude se veut une revue des différents principes et méthodes utilisés en I.A., et plus spécifiquement pour la construction des systèmes experts. Il aurait été possible d'exécuter une analyse strictement sur les principes et méthodes sur lesquels sont fondés les systèmes experts mais celle-ci, à notre avis, aurait manqué de profondeur. Les techniques d'approche pour la résolution de problèmes sont fondamentalement les mêmes que ce soit un système expert, un système que l'on appelle robot ou tout autre système que l'on qualifie d'intelligent: ces systèmes veulent reproduire une forme de raisonnement.

La présente étude aborde donc en premier lieu, au chapitre 2, la résolution de problèmes en introduisant ces différents types de représentation et par la suite les méthodes de résolution.

Le chapitre 3 introduit les différents modes de représentation des connaissances. Certains modes peuvent être mieux adaptés à un problème particulier. Une brève description des principes est donnée pour chacun de ces modes ainsi que de leurs avantages et désavantages.

Le chapitre 4 présente les différents langages utilisés en I.A. Nous verrons que chaque langage est intimement lié à un mode de représentation.

Le chapitre 5 est consacré exclusivement aux systèmes experts; une analyse de leurs composantes, de leurs modes de fonctionnement et de leurs caractères de différenciation permet aux débutants de s'initier rapidement aux techniques existantes.

Le chapitre 6 aborde les différentes techniques utilisées dans les principaux champs d'applications de I.A., tels que la reconnaissance du langage parlé et écrit, la vision et l'apprentissage.

Le chapitre 7 est voué à l'inventaire des systèmes experts commercialisés ou en voie de développement. Nous traitons d'abord des sources d'information sur I.A., pour donner ensuite une description des particularités des tableaux sur les systèmes experts que l'on retrouve en annexe 1. Au-delà de 400 systèmes experts sont inventoriés et catégorisés selon leur domaine d'application.

Le chapitre 8 complète l'étude en présentant d'abord une série de recommandations sur la démarche à adopter pour la construction de systèmes experts. Nous terminons ce chapitre en abordant les différents avantages dont la direction Environnement d'Hydro-Québec pourrait bénéficier en utilisant ces techniques de programmation des connaissances expertes que sont les systèmes experts.

Finalement, nous terminons notre introduction en donnant une brève description des annexes que nous retrouvons à la fin de cette étude. L'annexe 1 est consacrée aux tableaux inventaires des systèmes experts. Ces systèmes experts sont inventoriés et catégorisés selon leur domaine d'application. L'annexe 2 donne sous forme de tableau les principaux fournisseurs de matériel informatique en I.A., leur adresse ainsi qu'une brève description de l'équipement et des logiciels disponibles. L'annexe 3 présente une brève description des différentes caractéristiques du langage PROLOG, ainsi qu'un programme écrit en turbo-prolog en rapport avec le problème des déversements accidentels de polluants.

2.0 LA RÉSOLUTION DE PROBLÈMES

2.1 REVUE

L'I.A. comprend un grand ensemble d'idées telles que, la déduction, l'inférence, la planification, le raisonnement de bon sens, la preuve de théorèmes, et tous les processus qui y sont reliés. Les applications de ces idées générales sont retrouvées dans des programmes sur la compréhension du langage naturel, les systèmes experts, en robotique, l'analyse de scène, les jeux et la preuve mathématique des théorèmes.

On distingue trois composantes dans un système de résolution de problèmes: une base de données, un ensemble d'opérateurs et une stratégie de contrôle.

Une base de données: décrit la situation du domaine des tâches et le but à atteindre ou le problème à résoudre. La base de données peut être constituée d'un ensemble de données structurées différentes, soit des réseaux, des listes, des expressions de calculs, des propriétés sur les listes. Par exemple pour la résolution de théorèmes, la base de données est constituée d'assertions représentées par des axiomes, des lemmes, des théorèmes dont le but est une assertion représentant le théorème à prouver. Un autre exemple; dans la résolution de problèmes des robots, la base des données est un modèle de l'univers composé d'assertions décrivant l'entourage physique du robot, et le but est une description qui doit être validée par une suite d'actions du robot.

L'ensemble des opérateurs: manipule la base des données. Par exemple dans la résolution de théorèmes, les opérateurs sont les règles d'inférence telles que le modus ponens et le principe de résolution.

Dans notre exemple du robot, ceux-ci sont les règles pour son déplacement. Quelquefois l'ensemble des opérateurs se résume à quelques règles générales d'inférence qui génèrent de nouvelles assertions de celles existantes; habituellement il est plus efficace d'utiliser un grand nombre d'opérateurs spécialisés qui génèrent de nouvelles assertions.

La stratégie de contrôle: décide quel opérateur doit être appliqué et où l'appliquer. Le choix de la stratégie de contrôle affecte le contenu et l'organisation de la base de données. En général il faut arriver au but en appliquant une suite appropriée d'opérateurs. Chaque application d'un opérateur modifie la situation de la base de données d'une façon ou d'une autre. En général si plusieurs suites d'opérateurs sont pertinentes, la représentation maintient la structure des données montrant les effets sur la situation de la tâche pour chaque suite alternative. Une telle représentation permet une stratégie de contrôle qui investigue différentes suites d'opérateurs en parallèle et dont l'attention est mise sur les suites qui ont l'air relativement prometteuses.

Les stratégies de contrôle comprennent deux types distincts de raisonnement: le raisonnement avant et le raisonnement arrière. Le raisonnement avant dont l'objectif est de conduire la situation, ou l'état du problème vers l'avant, c'est-à-dire de sa configuration initiale à une configuration satisfaisant le but. Par exemple le jeu d'échecs, la configuration initiale est la situation décrivant les pièces sur l'échiquier au début de la partie, et le but est n'importe laquelle configuration qui met échec et mat. Le raisonnement arrière est celui dont les opérateurs sont appliqués au but. Le but où le problème est converti en sous-problèmes qui sont, on espère, plus faciles à résoudre, et dont leurs solutions sont satisfaisantes à résoudre le problème original. En combinant ces deux types de raisonnement nous obtenons le raisonnement mixte ou bidirectionnel. La méthode implique, à chaque étape de la résolution, une comparaison entre le but à atteindre et l'état de la résolution du problème pour

en extraire une différence. Cette différence est alors utilisée pour indexer l'opérateur le plus pertinent pour réduire la différence. Si cet opérateur particulièrement pertinent ne peut être appliqué immédiatement à l'état présent du problème, des sous-buts sont établis pour changer l'état du problème (par des opérateurs); après que ces sous-buts sont atteints, l'opérateur pertinent est appliqué et la situation modifiée devient un nouveau point de départ pour résoudre le but original.

2.2 LES REPRÉSENTATIONS DU PRINCIPE

2.2.1 L'espace d'états

Un système de résolution de problèmes qui utilise le raisonnement avant et dont chacun des opérateurs travaille en produisant un nouvel objectif unique (un état nouveau) dans la base des données est dit représenter des problèmes dans la représentation par espace d'états. Un système qui utilise le raisonnement arrière dont chaque application d'un opérateur à un problème donne exactement un nouveau problème (dont le degré de difficulté, on espère, est un peu moins élevé que le problème précédent) est aussi dit représenter des problèmes dans la représentation par espace d'états. Ce sont des systèmes à entrée et sortie uniques.

La représentation par espace d'états d'un problème utilise deux types d'entités: les états et les opérateurs. Les états sont des structures de données donnant un aperçu du problème à chaque étape de sa solution, les opérateurs agissant comme moyen de transformer un état à un autre.

Un exemple simple d'un problème avec une représentation par espace d'états est le casse-tête suivant:

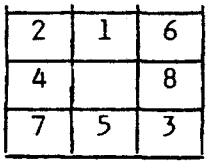


FIGURE 2.2.1

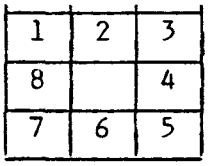


FIGURE 2.2.2

Une tablette carrée comprenant huit plaquettes d'égales dimensions numérotées de 1 à 8. L'espace pour la neuvième plaquette est vacante.

Une plaquette peut être déplacée horizontalement ou verticalement dans l'espace vide. Le problème est de transformer une configuration particulière dans une autre configuration telle que montrée aux figures 2.2.1 et 2.2.2.

L'ensemble de tous les états possibles d'un problème est appelé l'espace des états. Dans l'exemple précédent on peut dénombrer 181 440 états possibles. Chaque opérateur (le déplacement d'une plaquette) si est applicable à un état, donne exactement un nouvel état comme résultat.

La représentation par espace d'états est complètement définie par trois composantes. La première est l'ensemble des opérateurs O, la deuxième est le ou les états initiaux S, la troisième l'ensemble des états buts G. Cette représentation est alors décrite par le triplet (S, O, G). Une solution d'un problème est une suite finie d'applications d'opérateurs qui change un état initial à un état but.

L'espace des états peut être représenté aussi par un graphe où les noeuds sont les états et où les arcs sont les opérateurs transformant un état à un autre. L'état résultant d'une transformation est appelé successeur.

Une représentation graphique d'une partie de l'espace des états de l'exemple de la tablette à huit plaquettes est donnée en figure 2.2.3.

Le choix de l'ensemble des opérateurs est le suivant:

H = haut : déplacement du carré vacant d'un carré vers le haut.
B = bas : déplacement du carré vacant d'un carré vers le bas.
G = gauche: déplacement du carré vacant d'un carré vers la gauche.
D = droite: déplacement du carré vacant d'un carré vers la droite.

L'état initial est l'état tel que montré à la figure 2.2.1, l'état final est celui montré à la figure 2.2.1.

Une solution du problème est par exemple la suite des applications des opérateurs suivants:

GHDBGBDDHGHDBBGHDBGGHDHDBBGH.

Cette suite n'est pas nécessairement la plus courte mais en est une parmi toutes les suites possibles menant à une solution.

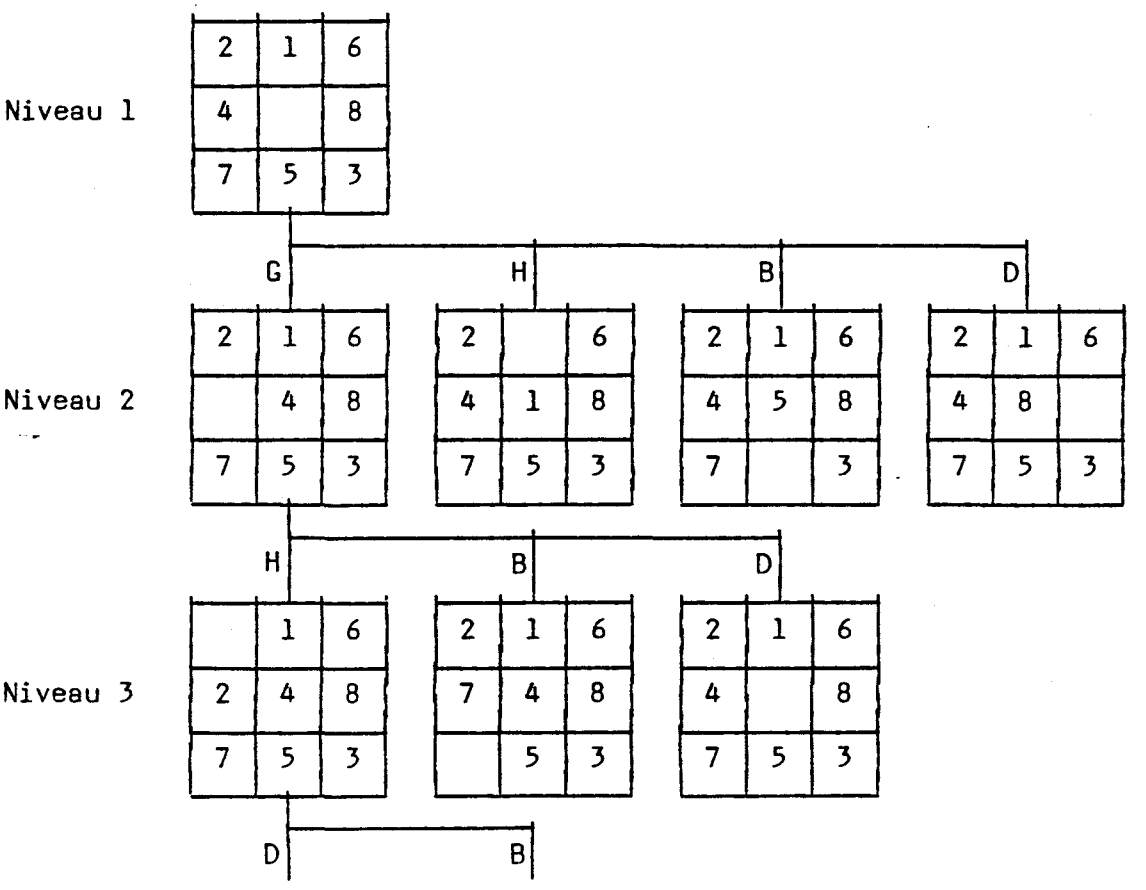


FIGURE 2.2.3
EXEMPLE D'ESPACE D'ÉTAT DU CASSE-TÊTE DE LA TABLETTE À HUIT PLAQUETTES

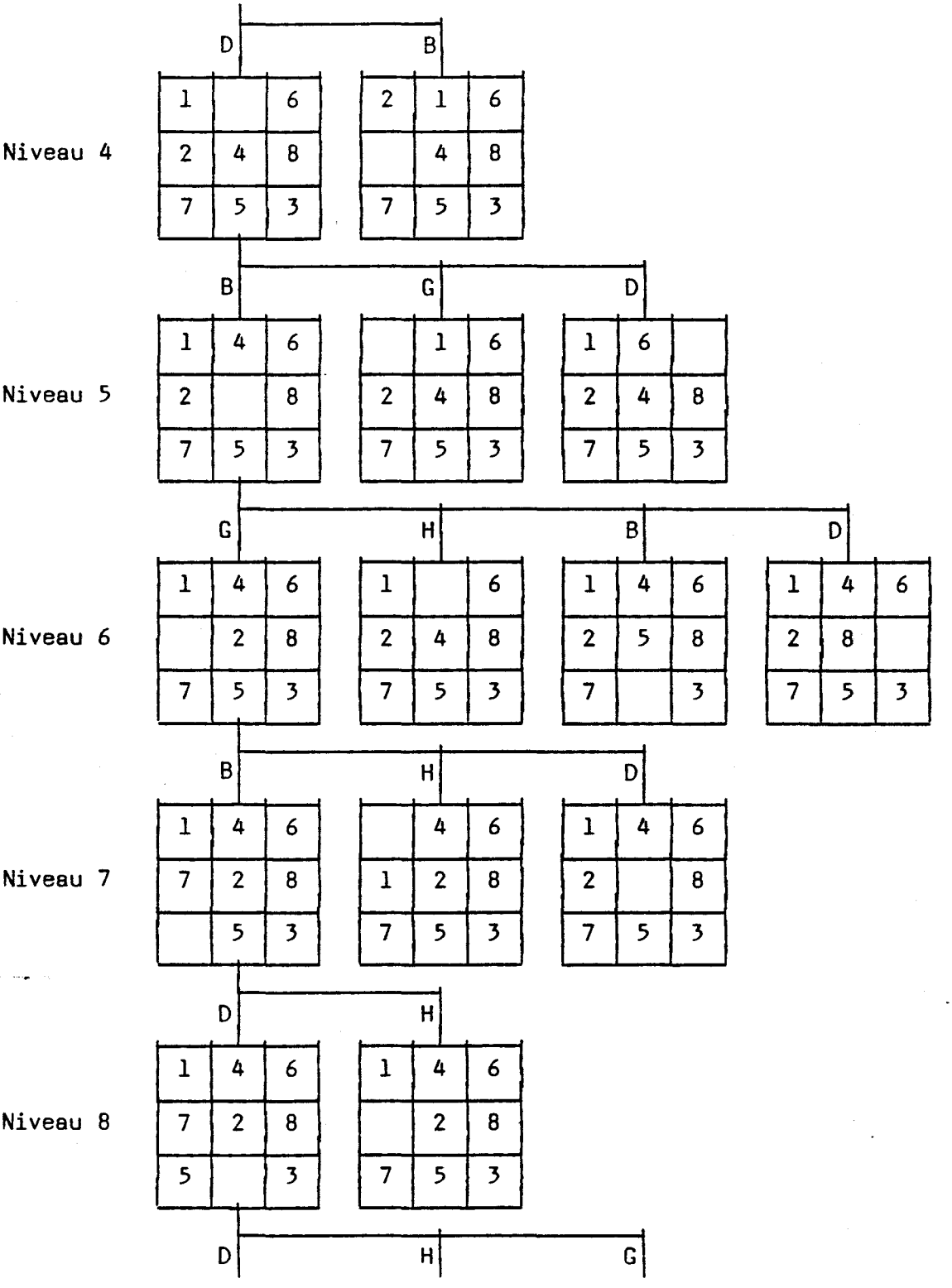


FIGURE 2.2.3 (suite)

EXEMPLE D'ESPACE D'ÉTAT DU CASSE-TÊTE DE LA TABLETTE À HUIT PLAQUETTES

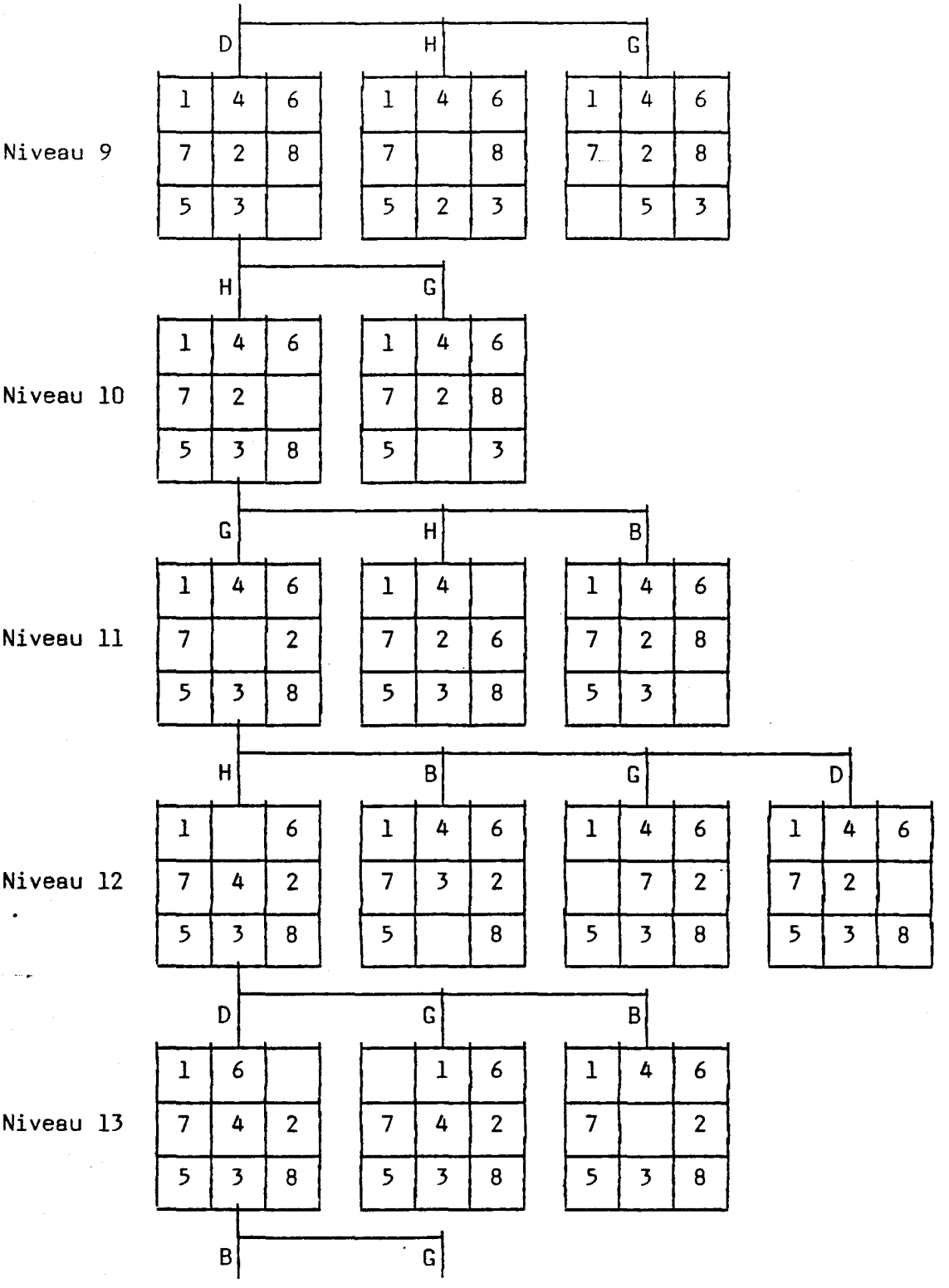


FIGURE 2.2.3 (suite)

EXEMPLE D'ESPACE D'ÉTAT DU CASSE-TÊTE DE LA TABLETTE À HUIT PLAQUETTES

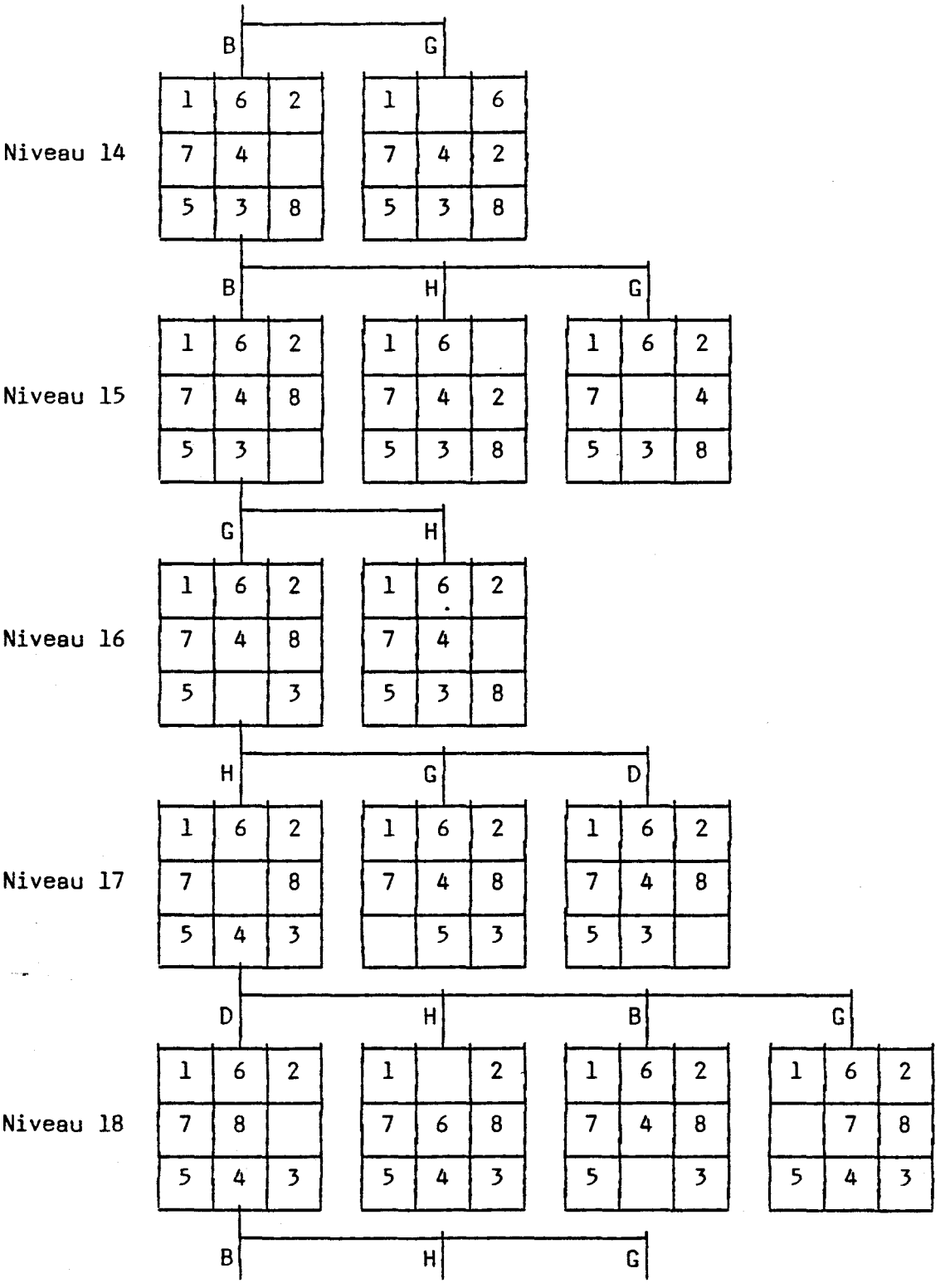


FIGURE 2.2.3 (suite)

EXEMPLE D'ESPACE D'ÉTAT DU CASSE-TÊTE DE LA TABLETTE À HUIT PLAQUETTES

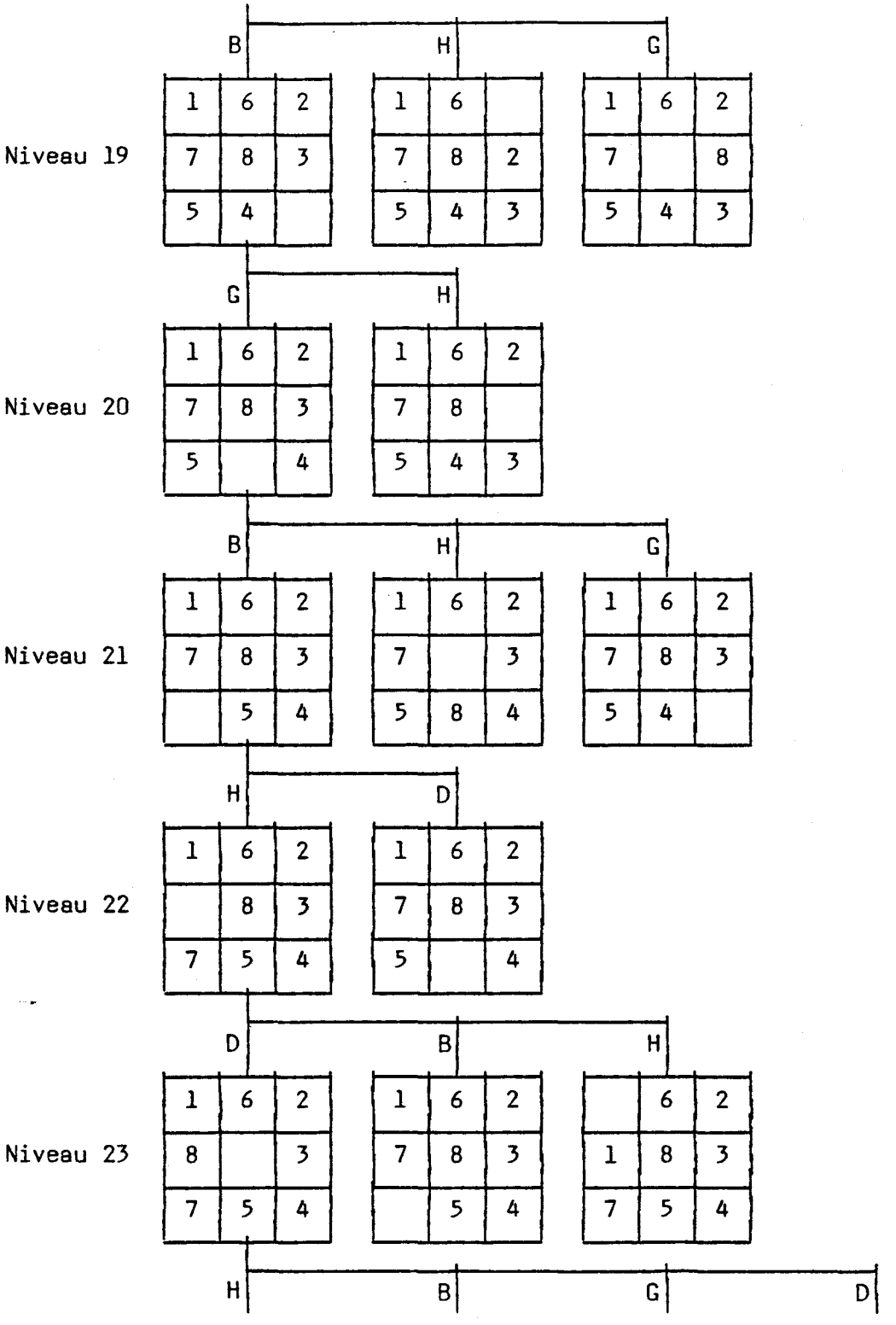


FIGURE 2.2.3 (suite)

EXEMPLE D'ESPACE D'ÉTAT DU CASSE-TÊTE DE LA TABLETTE À HUIT PLAQUETTES

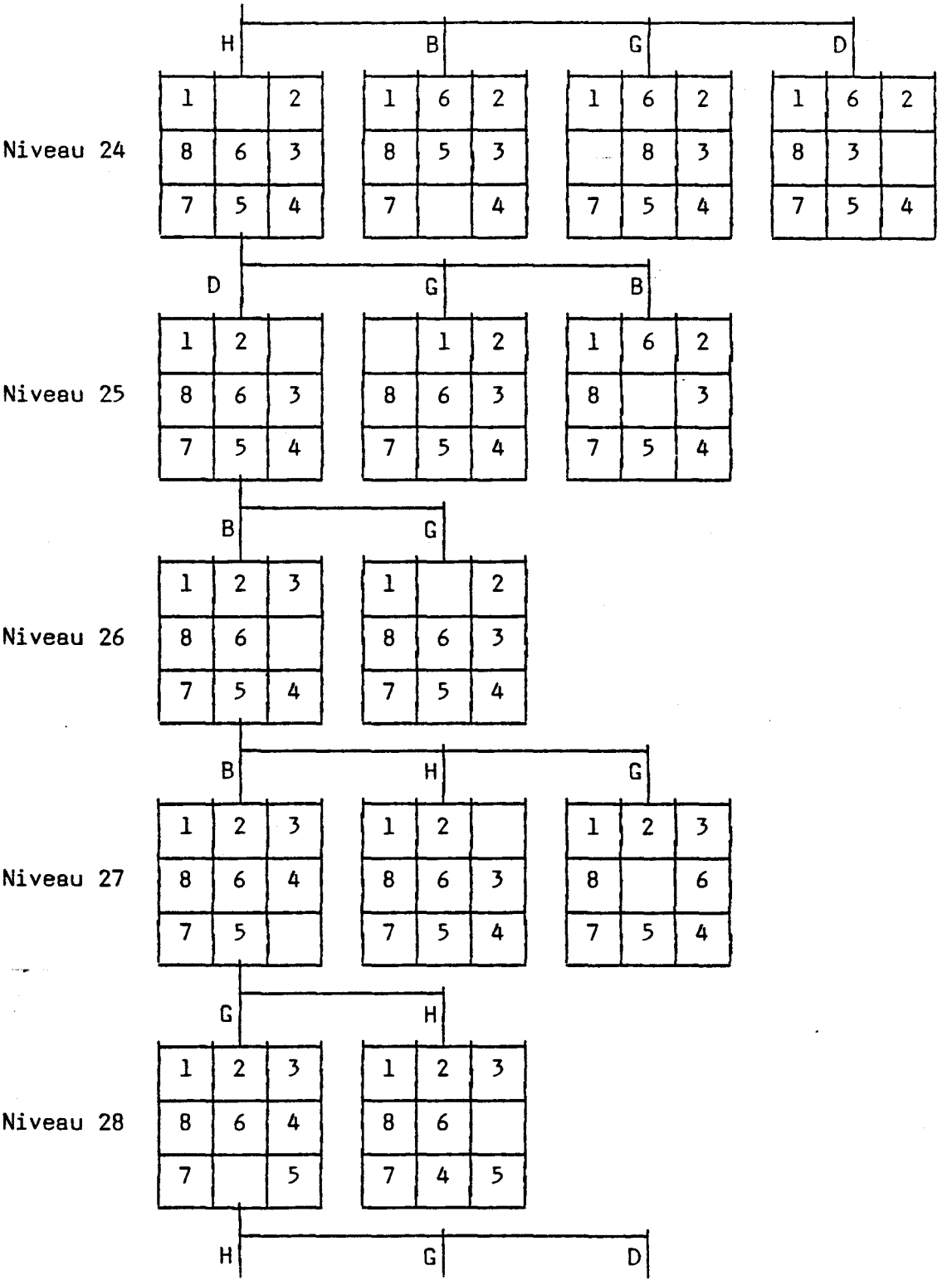


FIGURE 2.2.3 (suite)

EXEMPLE D'ESPACE D'ÉTAT DU CASSE-TÊTE DE LA TABLETTE À HUIT PLAQUETTES

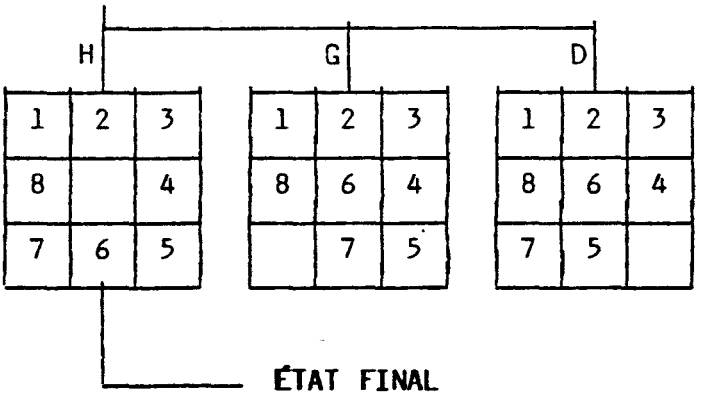


FIGURE 2.2.3 (suite)

EXEMPLE D'ESPACE D'ÉTAT DU CASSE-TÊTE DE LA TABLETTE À HUIT PLAQUETTES

2.2.2 La représentation par "réduction de problème"

Un système de résolution de problème qui utilise le raisonnement arrière et dont l'application d'un opérateur peut diviser le problème en un ensemble de sous-problèmes dont, nous espérons, que chacun est de difficulté moindre que l'original est dit qu'il emploie une représentation par "réduction de problème". Un système utilisant le raisonnement arrière est distingué par le fait que ces opérateurs peuvent changer un objectif simple en une conjonction d'objectifs.

Un problème dont la solution est immédiate est dit un problème primitif. Alors la représentation utilisant la réduction de problème est définie par le triplet suivant:

- la description du problème initial;
- un ensemble d'opérateurs transformant un problème en sous-problèmes;
- la description d'un ensemble de problèmes primitifs.

La réduction de problème peut aussi être représentée par la construction d'un graphe ET/OU, les graphes ET/OU donnent un moyen pour garder en mémoire les sous-buts qui ont été essayés et quelles combinaisons de sous-buts sont suffisantes pour atteindre le but original.

Correspondant à une formulation commune, un graphe ET/OU est construit selon les règles suivantes:

1- chaque noeud représente soit un problème simple, soit un ensemble de problèmes à résoudre. Le graphe a pour noeud initial le problème original;

2- un noeud représentant un problème primitif appelé un noeud terminal (n'a pas de descendants);

3- pour chaque application possible d'un opérateur à un problème P, transformant celui-ci en un ensemble de sous-problèmes, il existe un arc orienté du problème P au noeud représentant l'ensemble des sous-problèmes résultants.

   Considérons la réduction de P en trois ensembles de sous-problèmes distincts A, B et C. Dans la situation où P peut être résolu si au moins un des ensembles A, B et C peut être résolu, alors A, B et C sont appelés noeuds OU;

4- considérons A = (D, E), B = (B) un ensemble sous-problèmes à un élément, C = (F, G, H). En général, pour chaque noeud représentant un ensemble de deux sous-problèmes ou plus, il y a des arcs orientés du noeud vers chaque sous-problème individuel de l'ensemble. Si un ensemble de sous-problèmes peut être résolu seulement si chacun des membres doit être résolu, alors les noeuds sous-problèmes sont appelés noeuds ET.

La construction du graphe ET/OU dont on a discuté jusqu'à maintenant décrit tout l'espace de recherche du problème. Pour trouver une solution à un problème initial, il est seulement nécessaire de construire une partie du graphe qui peut démontrer que le noeud initial a une solution. Ce sous-graphe est appelé le graphe solution ou, dans des cas plus restreints, l'arbre solution. Les règles suivantes s'appliquent alors:

Un noeud a une solution si:

1- c'est un noeud terminal;

2- c'est un noeud non terminal, mais dont ses successeurs sont des noeuds ET, et peuvent être tous résolus;

3- c'est un noeud non terminal, mais dont ses successeurs sont des noeuds OU, et qu'au moins un peut être résolu.

Similairement, un noeud n'a pas de solution si:

1- c'est un noeud non terminal qui n'a pas de successeurs;

2- c'est un noeud non terminal dont ses successeurs sont des noeuds ET, et qu'au moins un n'a pas de solution;

3- c'est un noeud non terminal dont ses successeurs sont des noeuds OU, et tous n'ont pas de solution.

Il est pertinent de noter à ce stade-ci qu'un graphe de l'espace des états (dans le cas d'un raisonnement arrière) est un graphe ET/OU ne contenant seulement que des noeuds OU (un cas particulier d'un graphe ET/OU).

Un exemple qui se prête bien à la représentation par réduction de problème est celui du casse-tête de la "Tour de Hanoi". Une version de ce casse-tête est composée de trois disques de diamètre gradués troués au centre, et de trois piquets numérotés 1, 2 et 3. À l'état initial, les trois disques sont superposés au piquet n^{o} 1, le plus grand en-dessous, le plus petit au-dessus. Le problème est de transférer les trois disques au piquet n^{o} 3 dans le même ordre, mais seulement un disque à la fois peut être transféré. On a une représentation graphique du problème à la figure 2.2.4.

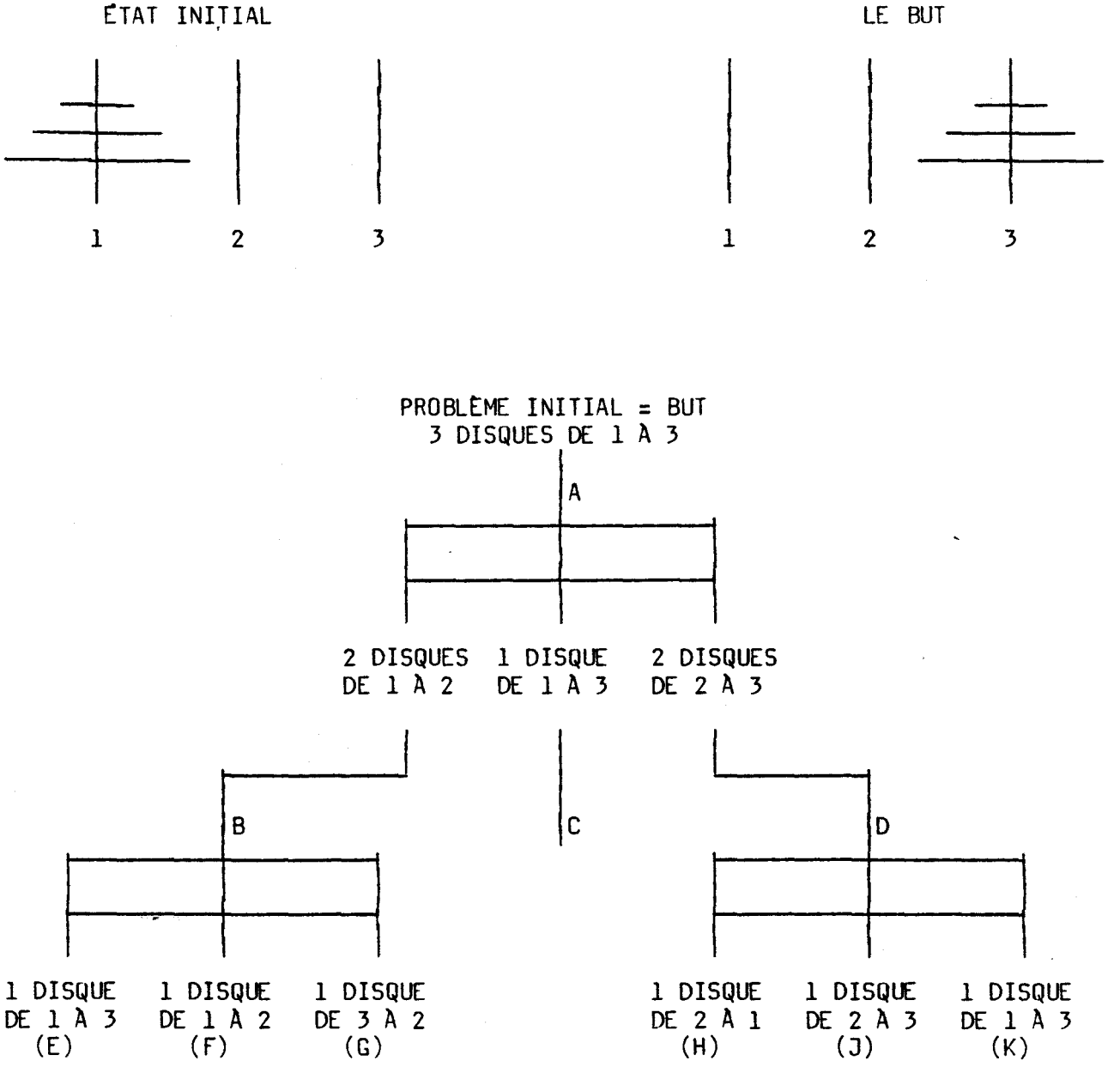


FIGURE 2.2.4

CASSE-TÊTE DE LA TOUR DE HANOÏ

EXEMPLE D'ESPACE PAR RÉDUCTION DE PROBLÈMES

Le problème initial est remplacé par trois sous-problèmes; chacun de ceux-ci est de difficulté moindre et la résolution conjointe résout le problème initial: le problème A est remplacé par les sous-problèmes B, C et D. Par la suite chacun de ces sous-problèmes est décomposé en sous-problèmes: la solution de B est obtenue par la résolution conjointe de B, F et G et ceux-ci sont des noeuds terminaux, C est résolu puisqu'il est terminal, la solution de D est obtenue par la résolution conjointe de H, J et K et ceux-ci sont des noeuds terminaux.

2.2.3 Les arbres de jeux

L'arbre d'un jeu est la représentation de tous les coups possibles d'un jeu tels que:

- il y a deux joueurs qui alternent pour faire leurs mouvements;

- à chaque tour, les règles définissent quels sont les mouvements qui sont légaux, et les effets que chaque mouvement possible aura sur l'ensemble du jeu;

- chaque joueur a l'information complète de la position de son opposant, incluant le choix des mouvements qu'il peut faire, et ceux qu'il a déjà faits;

- la partie commence d'un état bien défini;

- la partie se termine par un gain pour un joueur et une perte pour l'autre, dans certains cas par une égalité.

Le noeud racine est l'état initial, pour lequel c'est au tour du premier joueur à faire un mouvement. Les successeurs sont les états qu'il peut atteindre par ce mouvement. Leurs successeurs sont les états résultant de la réplique de l'autre joueur, et ainsi de suite.

Les états terminaux sont ceux représentant un gain, une perte ou une égalité. Chacun des chemins du noeud racine au noeud terminal représente une suite de mouvements possibles du jeu.

L'ensemble de toutes les suites de mouvements possibles du jeu est l'arbre du jeu.

Un exemple simple d'un arbre de jeux est celui du tic-tac-toe. Considérons les joueurs A et B, et A joue le premier. La figure 2.2.5 montre une partie de l'arbre.

2.3 LES MÉTHODES DE RECHERCHE

2.3.1 La recherche aveugle de l'espace des états

La recherche dans l'espace des états est définie par le triplet (S, O, G) où S est un ou plusieurs états initiaux, O est l'ensemble des opérateurs, G est l'ensemble des états buts. L'espace des états est communément identifié avec un graphe orienté dans lequel chaque noeud est un état et chaque arc représente l'application d'un opérateur transformant l'état en un successeur. Une solution est un chemin menant d'un état initial à un état but.

La recherche pour une solution est conduite en faisant explicitement un parcours de l'espace des états pouvant contenir un cheminement à une solution. Si l'ordre, dans lequel un cheminement à une solution potentielle considérée, est arbitraire c'est-à-dire qu'elle n'utilise pas d'information spécifique permettant de juger du domaine dans lequel une solution pourrait exister, la recherche est alors appelée "recherche aveugle".

Cependant dans bien des cas la recherche aveugle n'est pas praticable à cause de la trop grande dimension de l'espace des états. Dans ces cas on utilise des techniques de recherche à l'aide d'heuristique.

Les états terminaux sont ceux représentant un gain, une perte ou une égalité. Chacun des chemins du noeud racine au noeud terminal représente une suite de mouvements possibles du jeu.

L'ensemble de toutes les suites de mouvements possibles du jeu est l'arbre du jeu.

Un exemple simple d'un arbre de jeux est celui du tic-tac-toe. Considérons les joueurs A et B, et A joue le premier. La figure 2.2.5 montre une partie de l'arbre.

2.3 LES MÉTHODES DE RECHERCHE

2.3.1 La recherche aveugle de l'espace des états

La recherche dans l'espace des états est définie par le triplet (S, O, G) où S est un ou plusieurs états initiaux, O est l'ensemble des opérateurs, G est l'ensemble des états buts. L'espace des états est communément identifié avec un graphe orienté dans lequel chaque noeud est un état et chaque arc représente l'application d'un opérateur transformant l'état en un successeur. Une solution est un chemin menant d'un état initial à un état but.

La recherche pour une solution est conduite en faisant explicitement un parcours de l'espace des états pouvant contenir un cheminement à une solution. Si l'ordre, dans lequel un cheminement à une solution potentielle considérée, est arbitraire c'est-à-dire qu'elle n'utilise pas d'information spécifique permettant de juger du domaine dans lequel une solution pourrait exister, la recherche est alors appelée "recherche aveugle".

Cependant dans bien des cas la recherche aveugle n'est pas praticable à cause de la trop grande dimension de l'espace des états. Dans ces cas on utilise des techniques de recherche à l'aide d'heuristique.

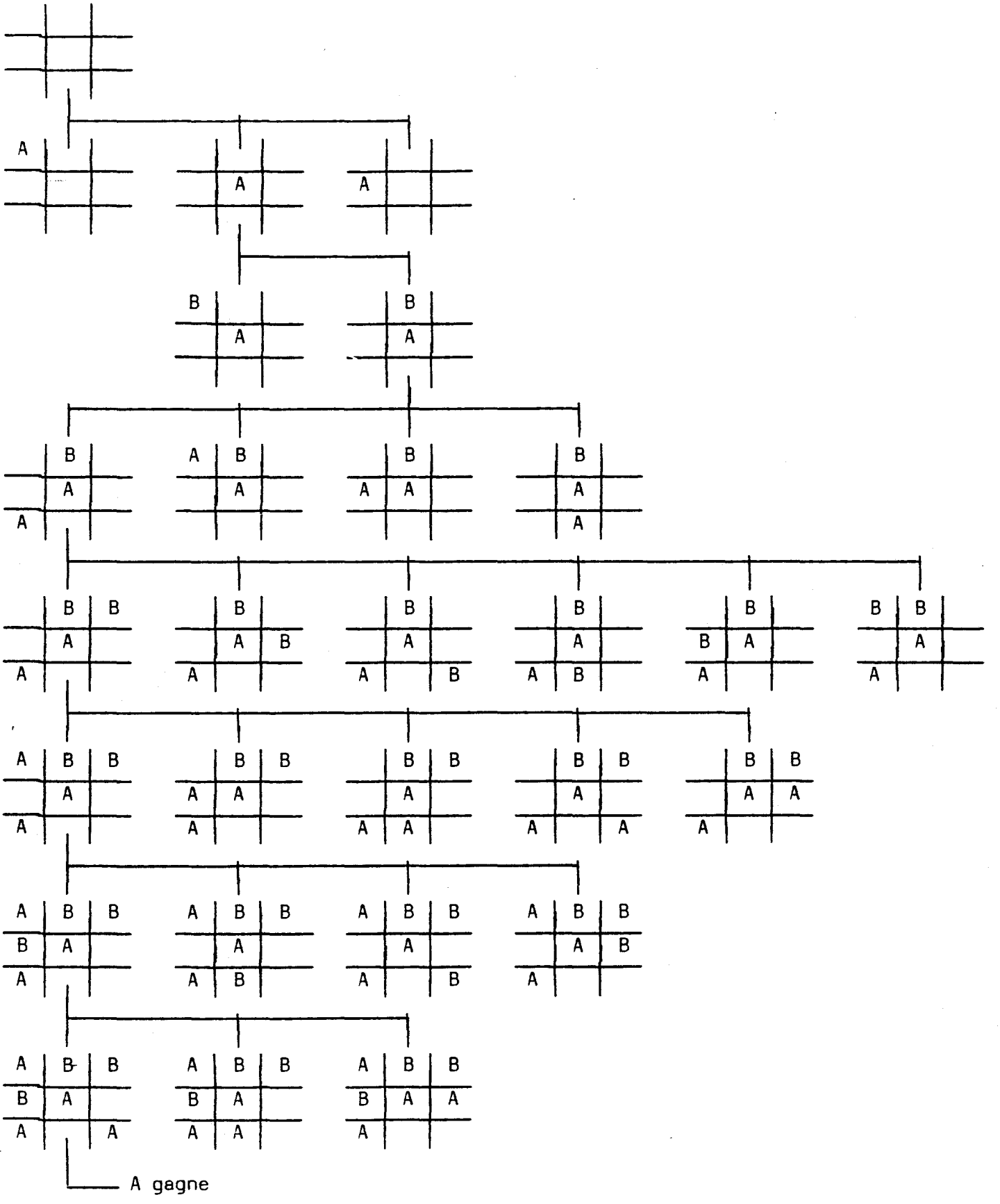


FIGURE 2.2.5

ARBRE DE JEUX POUR LE TIC-TAC-TOE

Trois méthodes de recherche aveugle sont examinées.

1- Méthode en largeur d'abord

La méthode en largeur d'abord déploie des noeuds dans l'ordre de leur proximité du noeud de départ, mesurant le nombre d'arcs les séparant. En d'autres mots, la méthode considère toutes les suites possibles d'opérateurs de longueur n avant toute suite de longueur n + 1. Une généralisation de la méthode peut être construite dans le cas où on veut trouver le cheminement le moins coûteux de l'état initial à l'état but. Une fonction coût non négative est associée à chaque arc joignant deux noeuds. Le coût de la solution est alors la somme des coûts des arcs le long du cheminement. Ce type de recherche est appelé "recherche à coût uniforme".

2- La recherche en profondeur d'abord

La recherche en profondeur d'abord est caractérisée par l'expansion du noeud généré le plus récent. La profondeur du noeud de départ est 0. La profondeur de tout autre noeud est un degré de plus que son prédécesseur. Une conséquence de déployer le noeud le plus profond en premier est que la recherche suit un seul cheminement du noeud de départ vers le bas à travers l'espace des états. Si elle rencontre un noeud sans successeur, elle considère un chemin alternatif. Dans beaucoup de problèmes, l'espace des états peut avoir une profondeur infinie ou peut être plus profond qu'une limite supérieure à une longueur acceptable d'une suite. Pour prévenir un chemin trop long, une limite est souvent définie sur la profondeur à être déployée.

3- La recherche bidirectionnelle

Les deux méthodes précédentes utilisent un raisonnement avant. C'est-à-dire d'un noeud de départ de l'espace d'état vers le but

et utilise un opérateur qui transforme le noeud i à un noeud successeur j. On peut tout aussi bien utiliser le raisonnement vers l'arrière (en profondeur d'abord ou en largeur d'abord). Les raisonnements avant et arrière peuvent être combinés dans une technique que l'on appelle la recherche bidirectionnelle. L'idée est de replacer un graphe de recherche, lequel peut croître exponentiellement par deux graphes plus petits, un commençant par l'état initial et l'autre par le but. La recherche se termine quand les deux graphes se croisent. Une version bidirectionnelle de la méthode du coût uniforme qui garantit de trouver le cheminement le plus court à travers un graphe général de l'espace des états est redevable à POHL (1969, 1971).

2.3.2 La recherche aveugle d'un graphe ET/OU

Un problème à résoudre utilisant la recherche d'un graphe ET/OU peut être défini en spécifiant un noeud de départ (représentant le but initial ou la description du problème), un ensemble de noeuds terminaux (décrivant les problèmes primitifs), et un ensemble d'opérateurs pour réduire les buts en sous-buts. Chaque application possible d'un opérateur à un noeud n est représentée par un arc dirigé du noeud n au noeud successeur. Tous les successeurs d'un noeud n sont appelés noeuds OU, puisque l'application d'un seul opérateur suffit à résoudre le problème au noeud n. Chaque noeud successeur au noeud n représente un ensemble de sous-problèmes. Si cet élément a plus d'un élément, alors l'ensemble des arcs dirigés de ce noeud OU correspond aux individus de l'ensemble. Les successeurs de ce noeud OU sont appelés noeud ET, puisque tous les éléments de l'ensemble doivent avoir une solution pour résoudre le sous-problème.

Pour distinguer visuellement les noeuds ET, des noeuds OU, les arcs dans le graphe sont reliés par une horizontale.

Par contraste la recherche dans un espace d'état utilise généralement le raisonnement avant, alors que la recherche dans un graphe ET/OU utilise exclusivement un raisonnement arrière. Comme la recherche dans l'espace des états, on peut avoir la technique de la méthode en largeur d'abord, ou la méthode en profondeur d'abord.

2.3.3 Les heuristiques de la recherche dans l'espace des états

Lorsque l'espace des états est trop grand et/ou pour empêcher l'explosion combinatoire, certaines informations pour la résolution d'un problème (dans un domaine particulier) peuvent être utilisées pour aider à réduire la recherche. Des informations de ce type sont appelées des informations heuristiques.

Les informations heuristiques sont utilisables dans les situations suivantes:

1- pour décider quel sera le prochain noeud à déployer, au lieu d'un déploiement aveugle, en largeur d'abord ou en profondeur d'abord;

2- lors du déploiement d'un noeud, pour décider lequel (ou lesquels) des successeurs sera généré, au lieu de générer aveuglement tous les successeurs possibles à un moment donné;

3- pour décider si certains noeuds devraient ne jamais être considérés du graphe de recherche.

Dans le premier cas on parle de recherche ordonnée, ou recherche des noeuds les plus prometteurs, dans le second cas on parle de noeud partiellement déployé, dans le troisième cas on parle de "pruning" (taillage, coupure) lorsqu'on décide si des noeuds ne doivent jamais être déployés, ceux-ci ne menant à aucune solution.

Dans tous les cas, la façon de définir, le noeud le plus prometteur, est estimée par une fonction d'évaluation. Le choix de la

fonction d'évaluation dépend du problème que l'on a à résoudre. Dans un cas l'espace des états peut contenir des cheminements multiples avec différents coûts, le problème est de trouver la solution optimale. Dans un second cas, un prolongement du premier cas, la recherche peut excéder les limites dont on s'est fixées, dans le temps et l'espace, avant de trouver une solution; alors il faut trouver une solution avec un effort raisonnable. Dans un troisième cas, cette fonction peut être celle qui minimise l'effort de la recherche.

L'algorithme A* (décrit par Hart, Nilsson et Raphael (1968)) définit le problème de trouver un cheminement avec un coût minimal, joignant le noeud de départ au noeud but dans le graphe de l'espace des états. Chaque arc (représentant l'application d'un opérateur) à un coût de 1; quoique le coût associé à un arc peut être arbitraire.

L'algorithme utilisé est la recherche ordonnée de l'espace des états. Sa caractéristique distinctive est la définition d'une fonction d'évaluation f*. Comme dans la plupart des cas, le noeud choisi pour déploiement est toujours celui dont f* est minimum. La fonction f* considère la valeur de chaque noeud comme ayant deux composantes: le coût minimum pour se rendre au noeud n défini par g*(n) et le coût minimum pour rejoindre un but du noeud n défini par h*(n). f*(n) est défini par:

$$f^*(n) = g^*(n) + h^*(n)$$

et représente le coût minimum pour passer au noeud n. Nous assumons que le coût des arcs est positif. On note par h, g et f les coûts actuels lesquels sont estimés par h*, g* et f*.

La fonction h* est la fonction porteuse de l'information heuristique et est définie d'une façon appropriée au domaine particulier du problème à résoudre.

Les conditions nécessaires pour l'application de l'algorithme A* sont: h* doit être non négatif et ne doit jamais surestimer le coût de rejoindre le but du noeud à évaluer c'est-à-dire que pour un noeud

n quelconque $h^*(n) \leqslant h(n)$; cette condition est appelée la condition d'admissibilité. On peut montrer que si h* est admissible (satisfait la condition d'admissibilité) et que le coût de chaque arc est positif et plus petit qu'une constante positive, alors A* garantie une solution optimale si celle-ci existe. Si h est identique à zéro, A* se réduit à la méthode de recherche du coût uniforme.

Une généralisation de l'algorithme A* est définie par une fonction d'évaluation de la forme suivante:

$$f^*(n) = (1 - w)\, g^*(n) + w h^*(n), \text{ où } w = [0, 1].$$

Dans le cas où w = 0, nous avons une recherche en largeur d'abord qui minimise le coût de la solution. Dans le cas où w = 1 nous avons une fonction qui minimise l'effort de la recherche.

Une forme encore plus générale de la fonction d'évaluation précédente est:

$$f^*(n) = g^*(n) + w(n)\, h^*(n), \text{ où } w(n) \geqslant 1.$$

Cette approche est appelée la pondération dynamique.

Finalement dans le cas où $f^*(n) = g^*(n) + h^*(n)$ il a été montré que si $h^*(n) \leqslant h(n) + e$ et $h^*(n) \geqslant h(n) + d$ c'est-à-dire que h* ne surestime jamais le coût minimum de la distance à parcourir par plus d'une valeur constante e, alors le coût trouvé par A* n'excèdera jamais par plus de e la solution optimale. Dans ce cas la fonction h* est dite e-admissible et la solution est e-optimal.

2.3.4 Les heuristiques dans les graphes ET/OU

Une solution d'un graphe ET/OU est un sous-graphe démontrant que le but est atteint. Le sous-graphe étant appelé arbre solution. Le coût d'un arbre solution peut être défini de deux façons:

1- la somme des coûts d'un arbre solution est la somme de tous les coûts des arcs dans l'arbre;

2- le coût maximum d'un arbre solution est la somme de tous les coûts des arcs concernant le cheminement le plus coûteux de la racine à un noeud terminal.

Par exemple, si chaque cas dans un arbre solution a un coût de 1, alors la somme des coûts est le nombre d'arcs dans l'arbre, et le coût maximum est la profondeur du noeud le plus profond.

Si l'espace de recherche a été exploré en entier, alors un arbre solution optimal peut être construit et son coût est mesuré comme suit: soit (n, m) le coût de l'arc du noeud n a son successeur m, on défini la fonction h(n) de la façon suivante:

1- si n est un noeud terminal alors h(n) = 0;

2- si n a des successeurs OU, alors h(n) est la valeur minimum des arcs de tous ses successeurs m_i, c'est-à-dire:

$$h(n) = \min_{m_i} C(n_i\ m_i) + h(m_i) = \min_{m_i} C(n, m_i) + h(m_i)$$

m_i étant un des successeurs de n;

3- si n a des successeurs ET, et la sommation des coûts est utilisée, la sommation de tous ses successeurs m_i c'est-à-dire:

$$h(n) = \sum_{m_i} (n_i\ m_i) + h(m_i) = \sum_{m_i} C(n, m_i) + h(m_i)$$

4- si n a des successeurs ET, et le coût maximum est utilisé, alors h(n) est le coût maximum parmi tous les successeurs m_i c'est-à-dire:

$$h(n) = \max_{m_i} (n_i\ m_i) + h(m_i) = \max_{m_i} C(n, n_i) + h(m_i)$$

5- si n est non terminal avec aucun successeur alors h(n) est infini.

Par cette définition h(n) est fini si et seulement si le problème représenté par le noeud n a une solution. Pour chaque noeud n qui peut être résolu, h(n) donne le coût de l'arbre solution optimal pour le problème représenté par le noeud n. Si s est le noeud initial, alors h(s) est le coût de la solution optimale du problème initial.

Il existe différentes approches pour la recherche ordonnée dans un graphe ET/OU. Une approche est celle prise par Nilsson (1969, 1971). L'approche utilise un processus à deux étapes pour le déploiement de noeuds individuels:

1e identifie l'arbre solution potentielle plus prometteur;
2e choisi un noeud de cet arbre pour le déployer.

Pour accomplir la première étape une fonction h* est définie à tous les noeuds n de l'arbre qui n'ont pas montré être insolubles. Cette fonction est un estimé de h(n), celle-ci estime le coût d'une solution optimale du problème au noeud n. Si n est connu comme étant un noeud terminal alors par définition h*(n) = h(n) = 0. Autrement, si n n'a pas encore été déployé alors l'estimé doit être basé sur n'importe laquelle information heuristique qui est disponible du domaine particulier du problème. La fonction h* est déduite de la définition précédente.

Finalement, l'arbre solution potentielle le plus prometteur T, est définie de h* de la façon suivante:

1- le noeud de départ s est dans T;

2- si l'arbre de recherche (la partie de l'arbre de recherche jusqu'à maintenant) contient un noeud n et des successeurs ET alors tous les successeurs sont dans T;

3- si l'arbre de recherche contient un noeud n et des successeurs OU de n, alors un successeur de n est dans T tel que: C(n, m) + h*(m) est minimal.

Le coût estimé de T est h*(s). Si tous les autres arbres solutions potentiels du même arbre de recherche étaient construits, on montrerait que T est un arbre pour lequel h*(s) est minimal. On montre de plus que cette recherche ordonnée est admissible (c'est-à-dire que l'on peut trouver un arbre solution de coût minimum, si celui-ci existe) si $h^*(n) \leqslant h(n)$ pour chaque noeud n déployé, et si tous les coûts des arcs sont plus grands qu'un nombre positif d.

2.3.5 La recherche dans les arbres de jeux

2.3.5.1 La procédure du Minimax

La procédure du Minimax est une technique de recherche dans les arbres des jeux. Posons A et B comme étant les noms des deux joueurs. Considérons d'abord que c'est au tour de A de jouer; le problème pour A est de trouver le mouvement qui maximisera son gain. Une fois que A aura joué, le problème pour B sera de choisir parmi tous les mouvements possibles qui s'offrent à lui, celui qui minimisera le gain de A. Considérons maintenant que les valeurs des positions terminales sont connues, alors les valeurs des positions non terminales que nous définirons comme étant F(n), sont calculées à partir des positions finales de la façon suivante:

1- la valeur donnée au joueur A a un noeud avec successeurs OU (lorsque c'est au tour de A de jouer) est la valeur maximum parmi les valeurs des successeurs;

2- la valeur donnée au joueur A a un noeud avec successeurs ET (lorsque c'est au tour de B de jouer) est la valeur minimum parmi les valeurs des successeurs.

Dans cette description de la procédure Minimax nous avons assumé que l'arbre complet du jeu a été généré. Cependant, l'arbre des possibilités est le plus souvent trop grand pour être généré dans son entier pour permettre un calcul de la fonction F(n) à partir des noeuds terminaux. Une alternative consiste à générer une portion raisonnable de l'arbre à partir de la position courante. Lorsque cette portion a été générée, une méthode doit nous permettre d'évaluer les valeurs des noeuds sub-terminaux. Une fonction de ce type est appelée la fonction évaluation statique.

2.3.5.2 La procédure Alpha-Beta

La procédure Alpha-Beta se résume comme suit: considérons deux joueurs A et B, et que c'est au tour de A à jouer. Parmi les coups possibles qui s'offrent à A, celui-ci évaluera ses chances à l'aide d'une fonction d'évaluation. Il choisit parmi les coups possibles celui dont la fonction a une valeur maximum. Toutes les évaluations et les développements de l'arbre pour lesquels la fonction d'évaluation n'est pas maximum sont éliminés.

3.0 LA REPRÉSENTATION DE LA CONNAISSANCE

Puisque la méthodologie de la recherche en I.A. est le design de programmes qui exhibe des caractéristiques de l'intelligence, les chercheurs en I.A. ont souvent adopté une approche pragmatique du sujet de la connaissance en concentrant leur attention à l'amélioration du comportement de leurs programmes. Une représentation de la connaissance est une combinaison de structures de données et de procédures d'interprétation qui si utilisée de bonne façon dans un programme mènera à un comportement intelligent.

Les travaux sur la représentation de la connaissance en I.A. ont comporté le design de plusieurs classes de structures de données pour la mise en mémoire de l'information, aussi bien que le développement de procédures qui permettent une manipulation de ces structures de données dans le but de faire des inférences.

On essaie dans le présent chapitre de donner une vue d'ensemble très générale des différents modes de représentation des connaissances que les chercheurs ont développés. La recherche en I.A. sur la représentation des connaissances est présentement l'un des domaines de recherche les plus actifs.

Quels sont les types de connaissances qui ont besoin d'être représentés d'une façon intelligente? Pour répondre à cette question considérons la liste suivante de types de connaissances qui ont besoin d'être représentés dans les programmes de I.A.: les objets, les événements, la performance et les métaconnaissances.

Les objets: typiquement ce sont des connaissances sur les faits, sur les objets qui nous entourent tels que: les oiseaux ont des ailes, la neige est blanche, les hirondelles sont des oiseaux, etc.

Les événements: ce sont des connaissances qui ont trait aux actions et événements dans le monde. En plus de pouvoir représenter les actions, un formalisme de représentation pourra être nécessaire pour une chronologie lors d'une suite d'événements, ainsi que les relations de cause à effet.

La performance: ce sont des connaissances qui ont trait au savoir-faire, comme par exemple se tenir en équilibre sur une bicyclette, comment composer des phrases, comment prouver des théorèmes.

Les métaconnaissances: ce sont des connaissances sur les connaissances. Par exemple nous savons la limite et l'origine de notre connaissance sur un sujet particulier, ou sur la fiabilité de certaines informations, ou l'importance relative d'un fait particulier du monde qui nous entoure, ou notre expertise dans un domaine particulier.

Les buts des programmes en I.A. peuvent être décrits en termes de capacité d'exécution de tâches cognitives, comme reconnaître les objets, répondre aux questions, les mouvements d'un robot, etc. Mais l'utilisation actuelle de la connaissance dans ces programmes comporte trois étapes:

1- acquisition de plus de connaissances;

2- réutilisation, autant de fois que nécessaire, de faits dans la base de connaissances qui sont pertinents pour la résolution d'un problème particulier;

3- le raisonnement à partir des faits pour en arriver à une solution.

Par acquisition on pense à apprentissage. L'apprentissage implique l'acquisition de nouveaux faits. Ceux-ci doivent passer par différents processus d'intégration. Dans certains programmes un nouveau fait est classifié avant d'être ajouté à la base de connaissances.

Dans d'autres, ils doivent être formulés de telle façon à ne pas interagir avec les autres.

La capacité de pouvoir réutiliser et distinguer une connaissance pertinente, lors de la résolution d'un problème, est une qualité de toute première importance dans un programme. Les humains sont particulièrement efficaces dans cette tâche, et plusieurs formalismes de représentation sont basés sur la mémoire des humains.

Concernant le raisonnement, quand nous demandons au programme d'exécuter une tâche qui n'est pas explicitement spécifiée dans sa base de connaissances il doit alors raisonner.

On peut distinguer différents types de raisonnement.

Le raisonnement formel: implique la manipulation syntactique de structures de données pour en déduire de nouvelles suivant des règles d'inférence très spécifiques.

Le raisonnement procédural: utilise la simulation pour répondre aux questions et résoudre les problèmes.

Le raisonnement par analogie: qui semble très naturel pour les humains mais qui est très difficile à accomplir par des programmes en I.A.

Le raisonnement par généralisation et abstraction: qui est un processus de raisonnement naturel pour les humains mais qui n'a pas trouvé d'application utile en I.A.

Le raisonnement de haut niveau: comporte l'utilisation de connaissances sur ce que nous savons, sur l'importance de certains faits, etc.

Les techniques de représentation de la connaissance font intervenir des routines pour la manipulation de structures de données spécialisées pour exécuter des inférences intelligentes. Cependant il est intéressant de noter qu'il n'existe pas de théories sur les formalismes de représentation.

Voici une description sommaire des différents formalismes de représentation de la connaissance qui sont utilisés en I.A. Ceux-ci sont illustrés par des exemples simples pour en permettre une meilleure compréhension.

3.1 LA REPRÉSENTATION LOGIQUE

Ce type de représentation concerne le traitement formel de la connaissance, et a été appliqué avec succès au développement de programmes informatiques qui peuvent raisonner. On peut considérer deux catégories de la représentation logique: la logique propositionnelle, et la logique des prédicats.

La logique propositionnelle: la notion la plus fondamentale en logique est celle de la vérité. Une proposition bien formulée a deux valeurs possibles de vérité: vrai ou faux. Par exemple: Jean est l'oncle de Marie, l'auto de Robert est bleue sont des propositions. En général des propositions pures et disjointes ne sont pas intéressantes. La plupart de nos paroles ou de nos pensées peuvent être représentées par une combinaison de propositions simples, soit par conjonction, soit par disjonction, soit par négation, soit par implication, ou bien par une équivalence.

D'une combinaison syntaxique de proposition nous pouvons construire des sentences de la logique propositionnelle comme des expressions mathématiques.

C'est dans la logique propositionnelle que nous avons rencontré en premier les règles d'inférence. Une règle d'inférence permet la déduction d'une nouvelle sentence à partir de sentences précédemment données, et en assure sa valeur vraie si les sentences originales sont vraies.

La règle d'inférence la plus connue est le Modus ponens. Cette règle déclare que si nous avons deux sentences de la forme X et, X implique Y sont vraies, alors on peut inférer que Y est vraie.

D'une façon formelle la règle de Modus ponens s'écrit ainsi:

$$X \wedge (X \rightarrow Y) \rightarrow Y$$

où $\wedge$ est une conjonction

et $\rightarrow$ est une implication.

La logique des prédicats: est une extension de la logique propositionnelle. Elle en garde tous les principes. Mais l'intérêt de la logique est changé. Plutôt que de garder seulement des sentences pour leur valeur de vérité, la logique des prédicats s'intéresse plus précisément à des déclarations sur les individus (ou objets) et de leurs relations entre eux. Les relations entre ceux-ci sont appelées prédicats. Par exemple considérons les individus suivants (objets, sentences): toi, cette feuille de papier, le numéro un, la dame de coeur, Socrate, etc. Un prédicat est appliqué à un nombre spécifique d'argument et à une valeur vraie ou fausse si quand les individus sont utilisés comme argument. Par exemple soit le prédicat "est rouge"; le prédicat "est rouge" a une valeur vraie ou fausse dans un contexte donné, et quand il est appliqué aux individus mentionnés plus haut:

est rouge (la dame de coeur) = vrai
est rouge (cette feuille de papier) = faux

Un prédicat peut avoir plus d'un argument. Un exemple de représentation de la logique des prédicats est le langage prolog.

Énumérons quelques avantages et désavantages de la représentation logique:

- Elle semble une façon naturelle d'exprimer un certain nombre de notions, elle correspond à une compréhension intuitive, elle est facile à formuler et à expérimenter.

- La logique est précise, il existe des méthodes standards pour déterminer le sens d'une expression dans un formalisme logique.

- La logique est flexible. Un fait particulier peut être représenté d'une façon simple sans avoir à considérer son utilisation possible.

- La logique est modulaire. Les assertions logiques peuvent être entrées dans la base de connaissances indépendamment les unes des autres.

- Le désavantage majeur de la logique vient de la séparation de la représentation et du processus d'inférence. La difficulté n'est pas de savoir comment mettre en mémoire les faits mais de savoir comment les utiliser. La difficulté réside dans la partie heuristique du programme.

3.2 LA REPRÉSENTATION PROCÉDURALE

La représentation logique (déclarative), s'intéresse particulièrement à l'aspect statique de la connaissance: élaboration de faits concernant les objets, les événements, et leur relation entre eux. La représentation procédurale a une ambition plus globale, elle s'intéresse en plus à savoir comment trouver des faits pertinents à être utilisés dans des inférences tout en gardant la puissance de la logique. Cet aspect du comportement intelligent a été le mieux capturé dans des représentations procédurales.

Avant l'avènement de la représentation procédurale, les chercheurs en I.A. se sont concentrés à déterminer quels types de connaissances pouvaient être représentés adéquatement dans un formalisme logique. Les questions concernant à savoir comment manipuler efficacement les faits dans la base de connaissances étaient considérées comme secondaires. Les procéduralistes par contre estimaient que la connaissance utile d'un domaine est intrinsèquement liée aux connaissances spécialisées ayant trait au savoir "comment utiliser la connaissance utile".

MACLISP est un exemple de langage utilisant la représentation procédurale.

Énumérons maintenant quelques avantages et désavantages de la représentation procédurale:

- facilite la représentation de connaissances heuristiques, lesquelles sont cruciales dans les gros systèmes;

- est capable de faire des raisonnements plausibles. Ce type de raisonnement semble nécessaire si on veut modéliser le raisonnement humain;

- est souvent inconsistante et incomplète en ce sens que les déductions ne sont pas toujours correctes, et que le système peut avoir tous les faits pour atteindre la conclusion mais est incapable de le faire;

- la modularité est sacrifiée, parce que les interactions entre les faits sont inévitables à cause de l'interdépendance des connaissances heuristiques.

3.3 LES RÉSEAUX SÉMANTIQUES

Les réseaux sémantiques furent développés comme modèle de la représentation des connaissances vers la fin des années 1960. Ceux-ci furent inventés comme un modèle explicite représentant la mémoire associative de l'humain. Un réseau consiste en des "noeuds" représentant les objets, les concepts, les événements, etc., et des arcs entre les noeuds représentant leurs liens et interrelations. Considérons le réseau simple suivant:

$$\text{oiseau} \xrightarrow{\text{une partie}} \text{ailes}$$

où "oiseau" et "ailes sont les noeuds et "une partie" est le lien spécifiant une relation. Une interprétation possible de ce fragment de réseau est l'énoncé suivant:

tous les oiseaux ont des ailes.

Comme illustré auparavant, des énoncés de cette sorte ont une représentation naturelle dans des modèles de représentation logique. Un aspect particulier cependant de la représentation par réseaux sémantiques est que des associations importantes peuvent être faites explicitement et succintement. Des faits pertinents d'un objet ou d'un concept peuvent être inférés d'un noeud auquel ils sont directement liés sans chercher à travers la base de connaissances.

La facilité de pointer vers des faits pertinents à cause des liens, établit une propriété hiérarchique dans le réseau. Par exemple le fragment de réseau suivant:

$$\text{hirondelle} \longrightarrow \text{oiseau} \longrightarrow \text{ailes}$$

peut être interprété comme ayant un sens, puisque l'hirondelle est un oiseau, et que les oiseaux ont des ailes, alors l'hirondelle a des ailes. L'interprétation sémantique de la structure du réseau, évidemment, dépend seulement du programme qui le manipule, il n'y a pas de conventions concernant le sens à donner aux liens.

3.4 LES CADRES SÉMANTIQUES (frames, scripts)

Les "cadres sémantiques" sont le formalisme de représentation le plus récent en I.A. Ceux-ci adaptent une structure plus complexe que les réseaux. Cette structure classifie la connaissance. L'objet est un ensemble structuré d'attribut. Les objets sont organisés selon une hiérarchie qui permet de traiter certains types d'implications que l'on appellent "héritage de propriété". Un objet est décrit par un ensemble de couples. Chaque couple est formé d'un nom attribut et d'une valeur correspondante. Par exemple: (couleur, nom, taille, etc.) (rouge, Pierre, 1,75 m, etc.) est un cadre.

Un aspect intéressant du comportement de l'inférence sur une structure de données basé sur les cadres, est sa capacité à déterminer si un cadre est applicable à certaines situations. L'idée est qu'un cadre est choisi pour aider au processus de compréhension d'une situation courante, et ce cadre essaie de s'accorder lui-même aux données qu'il découvre. S'il trouve qu'il n'est pas applicable, il peut transférer le contrôle à un autre cadre plus approprié.

Un autre aspect, qui mérite d'être mentionné, est qu'à chaque cadre on peut attacher des informations telles que:

- comment utiliser le cadre;
- quoi faire en cas de situation non prévue;
- quelle valeur par défaut il faut donner si l'information d'une partie du cadre est manquante;
- des procédures si nécessaires.

Ce mode de représentation organise la connaissance d'une façon qui permet d'y avoir accès directement. Ce formalisme a donné l'occasion aux chercheurs de tenter d'organiser et de structurer les grandes bases de connaissances.

KRL est un exemple de langage qui utilise ce formalisme de représentation.

3.5 LA REPRÉSENTATION ANALOGIQUE

La base de connaissances est une collection de représentations naturelles telles que cartes, images, modèles géométriques, etc. La modification de la base est par ajouts ou retraits de ces représentations.

L'approche est d'utiliser une représentation qui est analogue aux propriétés de la situation à représenter. La représentation est utilisée par une observation directe de l'information désirée telle que la forme, la grandeur, les angles, les contraintes, etc.

La représentation est naturelle et compréhensible, facile à modifier. D'importantes propriétés sont directement observables sans être inférées.

Cependant, cette représentation n'est appropriée que pour des tâches bien spécifiques.

4.0 LES LANGAGES

L'idée première, et la plus fondamentale, qui supporta le développement de langages de programmation presque exclusifs à l'I.A. était d'utiliser l'ordinateur pour manipuler de symboles arbitraires (pas seulement des nombres). Cette idée et celle de la technique du traitement des listes qui suivirent, furent introduites dans le langage 1 PL (1957), l'un des premiers langages de programmation.

Les éléments primaires du langage étaient des symboles contrairement à des nombres. L'association de ces symboles par des listes (le traitement des listes était alors introduit) rendait possible la construction de structures de données de forme et de grandeur non prévisibles. Le problème de la forme non prévisible des structures de données fut résolu par l'introduction du concept des "cellules". Chaque cellule était composée de deux mots. Chacun de ces mots pouvait contenir soit un symbole ou un "pointeur" à une autre cellule. Ce simple arrangement, appelé une liste structurée, permet la construction d'arbre à embranchement binaire. Le problème de grandeur non prévisible fut maîtrisé en allouant un nombre flottant de cellules requises correspondant à la structure de données.

Ce langage même s'il n'est plus guère utilisé, introduisit un certain nombre d'idées fondamentales qui maintenant sont incorporées dans tous les langages de programmation en I.A.

LISP

Ce langage utilise abondamment la notion de liste aussi bien pour la représentation des données que des programmes. Les éléments d'une liste sont énumérés entre des parenthèses, et peuvent représenter des listes simples et des listes structurées (des listes de listes).

Chaque élément est représenté en mémoire par une cellule composée de deux mots dont le second contient un pointeur sur la suite de la liste.

Un programme consiste en des fonctions définies dans un format plutôt mathématique. Chaque appel de fonction est représenté comme une liste. Le nom de la fonction est le premier élément et la valeur des autres éléments sont des arguments. La procédure d'exécution d'un programme en LISP se définit comme étant l'application d'une fonction sur ses arguments. Cette dernière caractéristique, et à cause du fait que la représentation des programmes est sous forme de listes, rend facile l'utilisation de la récursivité.

Les symboles en LISP sont appelés atomes, et chaque atome peut être associé à plusieurs propriétés.

C'est aux Etats-Unis que LISP est le plus utilisé pour la conception de systèmes. Plusieurs environnements commerciaux de programmation ont été créés sur des "machines LISP" et les commandes de programmation sont soit identiques ou similaires à l'usage du LISP.

Il existe plusieurs versions du langage LISP telles que:

- COMMON LISP,
- GOLDEN COMMON LISP,
- INTERLISP,
- XLISP,
- QLISP,
- MACLISP,
- MULTILISP,
- SI-LISP.

PROLOG

En Europe, en particulier en France (où il a été créé), en Angleterre et en Hongrie, ce langage connaît une certaine popularité.

Ce langage, tout en gardant l'idée du traitement des listes, s'appuie sur des résultats de la logique mathématique (logique du 1er ordre) appliqué au "principe de résolution". Ce principe est une généralisation du principe de Modus ponens. Cette combinaison permet, par l'introduction de faits et de règles, de trouver des enchaînements à l'établissement de nouveaux faits. PROLOG supporte déjà un mécanisme d'inférence (moteur d'inférence).

Le principe de résolution qui est l'unification, consiste en la mise en correspondance de deux ou plusieurs arbres (listes) susceptibles d'être à l'origine d'une solution.

Le mode de fonctionnement du moteur est par le raisonnement arrière (chaînage arrière).

Il existe plusieurs versions de PROLOG, on retrouve:

Nom	Compagnie
MICRO-PROLOG	LPA
PROLOG	Expert System International
PROLOG II	Prologia (France)
PROLOG-86	Solution Systems
TURBO-PROLOG	BORNLAND

AUTRES LANGAGES

Certains langages comme le FORTRAN, le PASCAL et le BASIC ont surtout servi à l'élaboration de systèmes experts. Certains outils de développement pour systèmes experts, ont été programmés en PASCAL.

Leurs moteurs d'inférence sont similaires à celui de PROLOG. Des exemples d'outils de développement en PASCAL sont: MICRO EXPERT, INSIGHT et EXPERT-EASE.

Un langage qui suscite de plus en plus d'intérêt est le langage C. La raison qu'on invoque est la grande rapidité d'exécution des programmes, une qualité de toute première importance lorsque la base de connaissances est grande. Il existe une tendance dirigée vers la programmation en langage C. Certains programmes commercialisés écrits en PROLOG et en LISP sont maintenant réécrits en langage C.

Les autres langages sont par exemple:

POP-2 : utilisé en Grande-Bretagne, a les propriétés de LISP et une syntaxe qui ressemble à l'ALGOL.

FUZZY : la création de ce langage fut motivée par la théorie des ensembles flous. C'est le langage le plus récent en I.A. La théorie des ensembles flous, qui se veut une généralisation de la théorie des ensembles Booleen, permet un degré d'appartenance d'un individu à un ensemble. Cette théorie introduit la notion de facteur de certitude.

SAIL : surtout appliqué aux systèmes de reconnaissance visuelle des formes, et aux systèmes servant à la compréhension de la langue parlée.

FOL : ce langage est similaire à PROLOG. Il est utilisé comme un vérificateur de preuve qui accepte des énoncés logiques et des commandes de preuve qui peuvent être exécutées et testées pour exactitude.

LM : langage surtout utilisé en robotique.

LOG LISP et LISLOG : ce sont des langages hybrides des langages LISP et PROLOG.

LOGO : ce langage est surtout utilisé pour l'enseignement. Celui-ci est très puissant pour la représentation visuelle de fonctions géométriques.

SMALLTALK et PLASMA: ce sont des langages de la catégorie des "langages objets" (dont le mode de représentation est similaire aux frames). Ces langages suscitent un intérêt croissant.

KRL : la programmation en langage KRL est basée sur une représentation de la connaissance pour cadres sémantiques (frames).

5.0 LES SYSTÈMES EXPERTS

Le terme "système expert" est utilisé pour décrire différents programmes informatiques basés sur une idée très bien répandue, celle de la notion des couples conditions-actions appelés "règles de production" ou tout simplement "règles".

Les systèmes experts suscitent un important engouement dans les divers milieux professionnels parce qu'ils permettent l'informatisation de certaines fonctions intellectuelles telles que l'identification ou le diagnostic de situations, la prévision d'événements, la conception d'objets, la planification d'actions. Celles-ci trouvent des applications concrètes dans les entreprises.

De telles fonctions sont souvent difficiles à modéliser sous forme d'algorithme. Dans certains cas, cette difficulté peut être contournée en utilisant la méthodologie des systèmes experts. La caractéristique essentielle de cette méthodologie, par rapport à d'autres techniques de l'I.A., c'est de tenter de reproduire les facultés de décision ou de jugement des experts humains en prenant acte que faute d'algorithmes complets parfaitement structurés, les experts sont capables de fournir des granules d'algórithmes (ou granules de connaissances), spécifiques d'un domaine d'application, plus ou moins indépendants les uns des autres, non définitifs et donc susceptibles de révision.

Dans ce chapitre, on illustre les principes des systèmes experts. On débute d'abord par une description des différentes composantes d'un système expert soit la base des connaissances et le moteur d'inférence. Nous aborderons ensuite les caractères de différenciation des systèmes experts. Ces différences ont trait surtout aux aptitudes du langage de programmation et aux modes de fonctionnement du moteur d'inférence.

5.1 RÔLE ET ORGANISATION DES SYSTÈMES EXPERTS

Il est difficile de donner une définition exacte des systèmes experts, mais trois points de vue méritent d'être retenus pour cerner la notion des systèmes informatisés dits systèmes experts:

- d'abord pour le rôle dans la prise en charge de certaines activités humaines;

- par l'organisation offerte pour accueillir et exploiter les connaissances d'experts humains sur un domaine particulier;

- par le mode d'invocation des connaissances prises en compte par les experts pour exprimer l'expertise à l'intention du système.

5.1.1 Rôle des systèmes experts

Lorsque le processus intellectuel, par lequel un humain évalue une situation et prend une décision, est précisément modélisé, il est relativement facile et direct de le programmer. Par contre dans beaucoup de domaines tels que le diagnostic médical, la prospection minière, l'organisation du travail, les connaissances dont font appel les experts sont souvent éparses, parcellaires, puisées de leurs expériences, ou très spécifiques du domaine d'application. Dans ce domaine le "savoir-faire" des spécialistes semble alors, pour une large part tout au moins, représentable comme un ensemble d'unités de savoir-faire ou "règles", chacune d'elles étant appropriée pour une classe de situations éventuelles. Ces règles étant recueillies auprès d'experts. Chacune d'elles décrivant une étape possible du raisonnement de l'expert. De plus chacune d'elle étant sujette à révision.

Les systèmes experts se sont développés comme une technique visant à atteindre les trois objectifs suivants: premièrement, capturer

aisément les unités de savoir-faire, c'est-à-dire faciliter l'expression la plus directe possible des règles, par rapport à leur forme d'émergence; deuxièmement, exploiter les unités de savoir-faire, c'est-à-dire pour combiner et/ou chaîner des groupes de règles pour inférer (ou dériver) des connaissances telles que jugements, plans, preuves, décisions, prédictions, nouvelles règles, ou pour rendre compte de la manière dont les nouvelles connaissances ont été inférées; finalement, supporter aisément la révision de l'ensemble des unités de savoir-faire, c'est-à-dire des facilités pour les ajouts et suppressions de règles.

Au travers d'expériences très diverses, les systèmes experts sont développés comme une technique informatique visant à satisfaire ces trois objectifs.

5.1.2 Organisation des systèmes experts

Un système expert possède les composantes suivantes:

- un "langage" d'expression des connaissances fournies par les experts;

- une "base de connaissances" qui représente les connaissances spécifiques du domaine d'application (3e objectif);

- un "moteur d'inférences" qui exploite les connaissances de la base de connaissances en les considérant comme des données (1ere partie du 2e objectif);

- et possiblement des "fonctions complémentaires" de dialogue et d'explication de son propre comportement pour:

  - faciliter l'acquisition et la révision des unités de savoir-faire;
  - faciliter la saisie des données de problèmes;

- assurer une interaction éventuelle avec l'utilisateur en cours de résolution;
- rendre compte de la manière dont a été conduite la résolution;

- finalement, quelquefois avec des possibilités d'apprentissage (2e partie du 2e objectif).

Le langage d'expression des connaissances: du point de vue de l'usager, généralement les langages naturels offrent la forme la plus confortable pour communiquer avec un système expert, soit pour instruire le système ou soit pour le consulter. En particulier, des énoncés décrivant des actions à exécuter si certaines conditions sont observées, sont naturellement codées dans des règles. De plus c'est ce type d'information qui est le plus fréquemment utilisé par des experts humains dans l'exécution de leurs travaux.

La base des connaissances: on distingue dans cette base deux types de connaissances: les "connaissances assertionnelles" ou "faits", les "connaissances opératoires" ou "règles".

Les faits sont des connaissances décrivant des situations (ou conditions) considérées soit comme établies, ou à établir.

Les règles représentent le savoir-faire; elles indiquent quelles conséquences tirer ou quelles actions à accomplir si telle situation (ou condition) est établie ou à établir.

Une règle, par exemple, peut être un énoncé de la forme "Si cette condition (ces conditions) est(sont) vérifiable(s) ALORS cette action est appropriée." La partie SI de la règle énonce les conditions qui doivent être présentes, parmi les faits établis ou à établir, pour que la règle soit applicable, et la partie ALORS est l'action appropriée à prendre. Durant l'exécution du programme, une règle dont la partie condition est satisfaite est déclenchée, c'est-à-dire qu'elle peut avoir sa partie action exécutée par le "déclencheur"

(nous reviendrons plus loin sur ces notions). L'action des règles peut changer la base des faits, ceci a pour effet que certaines règles qui n'étaient pas déclenchables avant, le deviennent parce qu'elles ont leurs parties conditions satisfaites.

Les règles sont habituellement des connaissances d'assez faible envergure pour être facilement compréhensibles par l'usager, tout en correspondant à des opérations distinctes significatives dans le domaine. On dit des règles qu'elles ont des "granules de connaissances".

Le moteur d'inférence: c'est un programme qui met en oeuvre des mécanismes généraux de combinaison des connaissances assertionnelles (faits) et connaissances opératoires (règles). Les stratégies utilisées peuvent être très différentes d'un moteur à un autre. Le moteur puise parmi les règles, les enchaîne, jusqu'à satisfaire une condition d'arrêt (qui dépend du moteur et de la base des connaissances disponibles). La résolution d'un problème passe par l'application d'un ensemble de règles qui se conditionne indirectement, les unes les autres par leurs effets sur la base de connaissances.

Le mode d'invocation des connaissances: les conventions proposées pour la formulation des règles sont très variables d'un système expert à un autre. Cependant un point commun réside, en matière de langages, c'est la "rédaction associative" des règles. Par rédaction associative on entend que l'on veut évoquer les règles par un mode d'accès associatif: c'est-à-dire que l'information contenue dans une règle n'est pas lue en donnant une adresse précise d'une cellule de la mémoire, mais en donnant une partie du contenu de la cellule qui contient l'information recherchée.

L'expression explicite des conditions de déclenchement dans la représentaton de la règle correspond souvent à la formulation naturelle par les experts de leur savoir-faire. On exploite l'information contenue dans les conditions de déclenchement; par comparaison

entre les conditions de déclenchement des règles et des faits à un instant donné permet de filtrer les règles pour en retenir un certain nombre.

La rédaction associative permet en principe de rédiger chaque règle en ignorant l'existence effective des autres. On présume seulement que chaque règle introduite dans la base de connaissance comporte, quel que soit son auteur, les éléments du filtre qui permettront de l'appeler. On présume aussi que les faits introduits comme effets d'une règle peuvent contribuer à appeler d'autres règles par leur contenu des éléments du filtre.

5.2 LES MOTEURS D'INFÉRENCES

Dans cette section on décrit principalement le "cycle de base" d'un moteur d'inférence. L'enchaînement du cycle de base, est un mécanisme nécessaire à toute résolution de problème.

5.2.1 Le cycle de base d'un moteur d'inférence

Disons d'abord qu'une règle est représentée par deux compsantes: le "déclencheur" et le "corps". Le déclencheur réunit les conditions de déclenchement. Le corps réunit les informations relatives aux effets résultant du déclenchement.

Le moteur d'inférence enchaîne des cycles de travail comportant chacun deux phases: une phase d'"évaluation" et une phase d'"exécution".

Avant le lancement du moteur, la base des connaissances contient les informations représentatives de l'énoncé du problème à traiter: les faits avérés et faits à établir (expressions de problèmes ou buts) constituent la "base des faits", les connaissances opératoires constituent la "base des règles".

En phase d'évaluation, le moteur détermine s'il existe des règles, dans la base des règles, à déclencher par rapport à l'état courant de la base des faits, si oui, ces règles seront retenues pour être déclenchées en phase d'exécution.

L'arrêt du moteur est commandé soit en phase d'évaluation ou en phase d'exécution. L'arrêt du moteur en phase évaluation proviendra de l'absence de règles à déclencher. L'arrêt du moteur en phase d'exécution proviendra d'un ordre donné par l'une des règles déclenchées.

5.2.2 La phase d'évaluation

Généralement la phase évaluation comprend normalement trois étapes (dépendant de la complexité du moteur d'inférence): la sélection, le filtrage, et la résolution_des_conflits.

La sélection (ou restriction) détermine par l'état présent de la base des faits et la base des règles (dans le cas d'un retour-arrière du moteur: par un état passé) un sous-ensemble de la base des faits et un sous-ensemble de la base des règles qui méritent d'être comparés lors de l'étape du filtrage. Une méthode courante consiste à répartir les faits et les règles en classes particulières en exploitant des connaissances particulières du domaine d'application.

Comme les systèmes, en général, sont devenus de plus en plus importants et de plus en plus complexes, les questions d'efficacité ont nécessité l'élaboration de structures sur la base des règles et sur la base des faits. Par exemple, pour permettre une accessibilité plus rapide aux règles qui sont applicables dans une situation donnée, sans avoir à vérifier si toutes les règles sont déclenchables, une façon de faire est d'indexer les règles ou les partitionner correspondant à des conditions qui les déclencheront. Une autre façon peut alors consister à privilégier les faits à établir

par rapport aux autres ou même le fait à établir (le problème, le but) le plus récemment apparu par rapport au plus ancien (stratégie en profondeur d'abord).

Leur moteur d'inférence à l'étape du filtrage (pattern-matching) compare les règles et les faits retenus de l'étape sélection. Une partie de ces règles seront rassemblées parce que les conditions de déclenchement ont été satisfaites. Les règles seront appelées "ensemble de conflits".

La résolution des conflits en pratique est le cas d'une situation où l'ensemble de conflits est d'un ensemble à plus d'un élément c'est-à-dire le cas où plus d'une règle est déclenchable dans un même cycle. C'est une situation typique des systèmes avec un très grand nombre de règles. Le moteur doit alors choisir une règle parmi cet ensemble. La phase de résolution des conflits est celle où on a apporté toute une attention comme par exemple, le contrôle de l'instabilité, l'établissement de l'ordre des actions, l'interruption. Plusieurs approches différentes peuvent être essayées, comme par exemple:

- la première qui est filtrée, ou la première est définie en termes d'ordre sur la base des règles;

- la règle la plus prioritaire où la priorité est définie par le programmeur correspondant aux caractéristiques de la demande de la tâche à accomplir;

- la règle la plus spécifique, qui est celle dont la partie condition (la partie déclencheur) détaille le mieux la base des faits;

- une nouvelle règle qui vient d'être déclenchée, et qui n'était pas apparue précédemment;

- par un ordre tout simplement arbitraire;

- par l'exploration de toutes les règles applicables en parallèle;

- la règle se référant à l'élément le plus récent ajouté à la base des faits;

- la règle qui est jugée de par sa forme, la moins complexe;

- la règle qui semble d'emploi la moins coûteuse;

- ou une combinaison des approches précédentes.

5.2.3 La phase exécution

Dans cette phase du cycle de base, le moteur d'inférence commande la mise en oeuvre des actions définies par les règles résultant de la phase évaluation. Les stratégies de mise en oeuvre peuvent être très différentes d'un moteur à l'autre.

Lorsque l'ensemble des règles est vide certains moteurs très simples s'arrêtent; on dit que ces moteurs ont un "régime de contrôle irrévocable". Par contre, on parle d'un moteur à "régime de contrôle par tentative" lorsqu'il y a remplacement de déclenchements de règles par d'autres. Pour signifier qu'un moteur revient sur une résolution de conflits antérieurs en remettant en cause des déclenchements de règles on dit qu'il opère un "retour-arrière" backstrack). En résumé un retour en arrière s'opère lorsqu'une impasse est rencontrée au cycle n, un retour-arrière est réalisé au cycle n+1 par un retour au contexte du cycle n-1, cependant l'ensemble des règles résultant de la résolution des conflits au cycle n-1 est automatiquement écartée de l'ensemble de conflits du cycle n+1. Ce type de retour-arrière est appelé "retour-arrière chronologique"; d'autres types de retour-arrière déterminent différemment le contexte de retour.

5.3 AVANTAGES ET DÉSAVANTAGES DES SYSTÈMES EXPERTS

Les systèmes experts ont le plus souvent été utilisés pour représenter la connaissance des experts à savoir comment un expert exécute une tâche particulière, comme par exemple, la compréhension du langage parlé, le diagnostic médical, l'exploration minière, ... En psychologie aussi, les systèmes experts sont des outils très populaires pour la modélisation du comportement humain, sans doute par le caractère stimuli-réponse des règles.

Des attributs importants des systèmes experts sont la "modularité" et la "granularité". Les règles de la base de connaissances peuvent être ajoutées, modifiées et changées d'une façon indépendante. Le changement d'une règle peut être exécuté sans qu'on soit dérangé par des effets directs sur les autres règles (toutefois, ce changement peut avoir un impact sur la performance du système) puisque les règles communiquent seulement par le moyen de la base des faits (ne communiquent pas directement entre elles). Une règle n'est déclenchée que lorsque certains aspects sont reconnus par le moteur. La modularité relative est une qualité importante lorsque l'on construit des systèmes avec une base de connaissances très élaborée. Elle rend plus facile la création de la base. Cependant la modularité est plus difficile à maintenir lorsque les systèmes sont dé plus en plus complexes. Dans ceux-ci même si la modularité peut être conservée, le manque d'interaction entre les règles conduit à une inefficacité qui peut devenir un problème important.

Un autre attribut est l'"uniformité" lors de la construction des règles de la base de connaissances. Puisque toute l'information doit être codée suivant une structure rigide, la compréhension des règles est plus facile. Il devient facile de mettre au point et de réviser la base de connaissances.

Pour certains cas de petits changements de la connaissance factuelle peuvent influencer rapidement le comportement global du système. Dans ces cas on parle de "sensibilité" ou de "réactivité" du système.

L'étape du filtrage du cycle de base est une opération a priori coûteuse en temps. Cependant des améliorations peuvent être recherchées tant au niveau de l'organisation des données mémorisées dans la base factuelle qu'au niveau de l'implémentation des mécanismes de filtrage.

La redistribution continuelle du contrôle pose un problème plus fondamental et difficile: celui de la résolution des conflits. En rédigeant la partie déclencheur des règles l'expert exprime les circonstances relatives à la base factuelle, qui justifient ou interdisent que telle règle déjà retenue par la sélection, soit intégrée à l'ensemble de conflits soumis ensuite à la résolution des conflits. L'algorithme de résolution des conflits choisit une règle parmi celle qui sont dans l'ensemble des conflits, qui obtiendra le contrôle. Le plus souvent, il s'agit d'un algorithme purement intrinsèque au moteur, c'est-à-dire indépendant de l'application. Dans certains systèmes, l'utilisateur dispose de primitives (quelquefois impliquées dans des règles spéciales appelées "Métarègles") pour intervenir explicitement sur l'algorithme de résolution des conflits ou sur l'algorithme de sélection qui ne sont alors que des algorithmes "par défaut".

5.4 CARACTÈRES DE DIFFÉRENCIATION ENTRE SYSTÈMES EXPERTS

Les systèmes experts se différencient les uns des autres plus ou moins profondément, soit d'un point de vue externe: selon le type de langage qu'ils proposent pour capturer la connaissance expert, soit d'un point de vue interne: selon les particularités et les modes de fonctionnement des moteurs d'inférences.

5.4.1 Différenciation selon les aptitudes du langage

Au-delà de la simple distinction, traditionnellement mise en avant, entre faits et règles, les langages associés aux systèmes experts

proposent fréquemment et parfois imposent d'autres classes de connaissances.

Faits: dans certains systèmes on peut distinguer des catégories de faits, chacune dotée de moyens de manipulations spécifiques.

Règles: les déclencheurs jouent un rôle essentiel pour permettre la "rédaction associative" des règles et d'en faciliter un style de "programmation déclarative".

Chaque système respecte un modèle de compatibilité entre les faits et les éléments appelés "filtres", qui composent les déclencheurs. Les modèles de compatibilité utilisés peuvent être:

- l'identitité,
- la semi-unification,
- l'unification.

Cependant sans être limité à ces modèles de compatibilité certains systèmes peuvent offrir une gamme d'opérateurs de filtrage, de fonction de filtrage de règles spécifiques au filtrage.

Un modèle de compatibilité qui est l'identité est un filtre qui ne peut être compatible qu'avec une expression de faits parfaitement identique, ou une expression de faits faisant intervenir des mots déclarés comme synonymes des mots présents dans le filtre et des mots déclarés comme non significatifs. Par exemple le filtre "CIEL BLEU" est compatible avec le fait "LE CIEL AZUR" si AZUR est synonyme de BLEU et LE est un mot non significatif.

Dans la plupart des moteurs d'inférences le modèle de compatibilité est la semi-unification (c'est-à-dire un cas particulier de l'unification utilisée en logique des prédicats). Les règles mais non les faits peuvent comporter des variables. Par exemple le filtre CIEL (COULEUR) est compatible avec le fait CIEL (BLEU). L'identificateur

de variable COULEUR est compatible avec n'importe quel identificateur de constante, qui est BLEU dans ce cas-ci; ou plus généralement avec n'importe quelle expression sans variable.

En Prolog, le mode de compatibilité est l'unification plénière, des variables qui sont admises aussi bien dans les faits que dans les règles. Par exemple un filtre "P (x, f(y), y(a));" est compatible avec un fait tel que: "P (h(u), v, g(n))"; où "u", "v", "n" sont des variables.

Dans certains systèmes la compatibilité entre le déclencheur et la base des faits n'est pas réduite à la compatibilité où l'incompatibilité absolue, mais plutôt vers un degré de compatibilité.

On peut définir les effets de déclenchements de règles comme des transformations de la base des faits. Certains auteurs trouvent plus approprié de distinguer trois sortes d'effets, en faisant un parallèle avec les trois classes d'instructions de base des langages de programmation:

- instructions de traitement: les langages proposent ou possèdent intrinsèquement des primitives pour élaborer et insérer dans la base des faits de nouveaux faits, voire même pour supprimer certains faits de cette base. Parfois des primitives sont présentes et permettent de modifier la base des règles. Il est usuel de classer toute règle capable de modifier l'état de la base des règles dans la catégorie des "métarègles";

- instructions d'entrée-sortie: certains langages proposent des primitives pour expédier des messages, consulter l'environnement: soit l'utilisateur, soit les capteurs, etc., ou plus généralement lancer des procédures externes au moteur d'inférence et à la base des connaissances;

- instructions de contrôle: certains langages disposent de primitives qui agissent directement sur le comportement du moteur d'inférence.

Objets assertionnels/opératoires: dans certains systèmes on offre des possibilités à l'utilisateur de structures les connaissances (faits, règles, procédures) en classes significatives du domaine d'application. Dans d'autres on offre des variantes plus ou moins complexes telles que les frames (cadres, modèles), les objets, les scripts, les prototypes. Une structure simple et courante combine en une seule entité un nombre variable de triplet de la forme "identificateur-propriété-valeur" ou "Objet-Attribut-Valeur" (O-A-V). Les valeurs de propriétés peuvent être de natures diverses: constantes, messages, procédures à activer, etc.

Dans certains systèmes pour le spécialiser dans un domaine on doit fournir de telles classes, c'est-à-dire déterminer les objets, les propriétés et les valeurs afférentes au domaine. Certains O-A-V peuvent être élaborés en cours d'exploitation du système: pour des cas particuliers objets plus généraux.

Procédures: ce sont des connaissances opératoires que les experts ont jugé naturel de désigner explicitement et non par rédaction associative. Les procédures sont appelées par leurs noms et non par leurs conditions d'application. Si on sait pertinemment que telle connaissance, précisément déterminée et assurément présente dans le système, doit être invoquée sans possibilité d'alternative, alors la rédaction associative n'a effectivement pas lieu d'être.

Certains langages permettent l'introduction des appels de procédures dans les règles et les métarègles (dans la partie corps, ou la partie déclencheur; en partie déclencheur ce sont des procédures mises en oeuvre au cours du filtrage). Comme on l'a vu précédemment les procédures peuvent être utilisées dans les O-A-V. Cette technique, d'attacher des procédures O-A-V, est appelée "attachement procédural".

Démons: ce sont des connaissances opératoires généralement écrites en rédaction associative; mais ne sont pas assujetties aux mêmes cycles de base. Celles-ci sont exploitées dès que surviennent certains changements de la base des faits, et plus rarement la base des règles; tels que ajouts, retraits, consultations, etc. L'usage des démons tend à accroître la réactivité des systèmes experts. L'utilisateur contrôle les démons par des directions spécialisées dans le corps des règles. On dit qu'on arme ou désarme un démon. Lorsque se produit un changement d'un type auquel on a associé un démon celui-ci s'il est armé exécute un travail de type spécifié à l'avance.

Usage de variables: l'usage de variables permet de condenser plusieurs expressions de connaissances en une seule. On dit qu'on factorise les connaissances. Les moteurs sont parfois qualifiés de "moteurs 1" ou "moteurs 0" selon qu'ils admettent un langage avec variable ou non. Parmi les langages, PROLOG (qui est un moteur d'inférence) est l'un des plus original. Il permet l'utilisation de variables aussi bien dans les règles que dans les faits. Toutes les variables présentes dans PROLOG sont quantifiées universellement. Par exemple une expression de la forme A(u) ⟵ B(u), C(u) est une règle interprétable comme "quel que soit x pour établir A(u) il suffit d'établir B(u) et C(u).

Capacité de représenter des connaissances incertaines et imprécises: dans beaucoup de domaines où l'on se préoccupe de développer des systèmes experts, les connaissances (règles, faits) traitées ou inférées sont incertaines ou imprécises. Un fait est incertain s'il contient une assertion dont on ne peut affirmer qu'elle est vraie ou qu'elle est fausse bien qu'elle doive effectivement avoir l'une ou l'autre valeur de vérité; une règle est incertaine lorsqu'elle produit des conclusions incertaines même à partir de prémisses certains. Un fait est imprécis s'il implique des objets incomplètement identifiés; une règle est imprécise si elle contient des faits imprécis en prémisses et en conclusion.

Il faut noter qu'à côté de la théorie des probabilités qui sert dans certains systèmes comme modèle de l'incertitude et de l'imprécision on dispose d'autres modèles plus ou moins empiriques tels que la théorie des possibilités elles-mêmes issues de la théorie des sous-ensembles flous. Celle-ci donne à chaque individu d'un ensemble un degré d'appartenance de l'individu à cet ensemble.

En pratique, le langage d'un système expert traitant l'incertitude et l'imprécision inhérentes aux faits ou règles, doit prévoir des conventions et primitives pour que l'utilisateur puisse affecter aux connaissances qu'il transmet un degré d'incertitude ou un domaine d'imprécision. Par exemple dans certains systèmes experts (comme MYCIN) le "coefficient d'atténuation" d'une règle de la forme "P implique Q" représente la certitude que "Q est vrai" lorsque "P est vrai". Dans PROSPECTOR on a introduit le "facteur de suffisance" et le "facteur de nécessité". Le facteur de suffisance défini à quel point il suffit que "P est vrai" soit certain pour que "Q est vrai" soit certain, tandis que le facteur de nécessité défini à quel point il est nécessaire que "P est vrai" soit certain pour que "Q est vrai" soit certain.

5.4.2 Différenciations selon les aptitudes du moteur d'inférences

5.4.2.1 Les modes fondamentaux de raisonnement

Les recherches sur le raisonnement ont permis de découvrir chez les humains deux modes fondamentalement différents de raisonnement. Parfois on travaille avec l'information disponible la prenant comme elle vient, et essayant de tirer des conclusions qui sont appropriées à nos buts. Dans l'autre cas nous travaillons en commençant par le but, (ou hypothèse, ou problème à résoudre) et regardant pour les évidences qui appuient notre hypothèse. On qualifie respectivement ces modes de raisonnement "raisonnement avant" et "raisonnement

arrière". En recherche sur les systèmes experts, par similitude, on qualifie les modes d'invocation des règles de "chaînage avant" et de "chaînage arrière".

Le chaînage avant: un moteur d'inférences fonctionne en "chaînage avant" (forward chaining) pur lorsque les faits de la base des faits, sur lesquels partent les déclencheurs des règles, représentent des informations dont la valeur de vérité est déjà établie. Les règles sont alors dites "règle-en-avant" (forward rules). Couramment, il est convenu que les déclencheurs sont leurs membres gauches.

Le chaînage arrière: un moteur d'inférences fonctionne en chaînage arrière pur (backward chaining) lorsque:

- certains faits de la base des faits sont considérés comme étant à établir ou évaluer; on les appelle problèmes, ou hypothèses, ou buts;

- les déclencheurs des règles se réfèrent uniquement aux problèmes en suspens;

- lorsqu'une règle est déclenchée, de nouveaux problèmes, définis par le corps de la règle, peuvent être introduits dans la base des faits, et sont considérés comme de nouveaux problèmes à résoudre;

- les problèmes auxquels réfère le déclencheur sont considérés comme résolus lorsque les problèmes introduits par le déclenchement de la règle, soit qu'ils s'avèrent être des problèmes déclarés a priori comme "primitifs" (des problèmes considérés comme résolus), soit qu'ils ont été résolus au préalable.

Les règles sont dites "règles-en-arrière" (backward rules) si l'exécution de l'inférence (tirer la conclusion) n'est éventuellement réalisée qu'après le déclenchement.

Le chaînage mixte ou chaînage bidirectionnel: certains systèmes fonctionnent en "chaînage mixte": une partie des faits de la base des faits sont considérés comme étant à établir, d'autres sont considérés comme établis; les conditions des règles peuvent porter simultanément sur des faits de l'une ou l'autre sorte.

Pour les systèmes qui invoquent, pour un même problème, à un instant donné des règles-en-avant, et d'autres moments des règles-en-arrière, on parlera de chaînage bidirectionnel.

Dans certains cas on a alternance entre un cycle en chaînage avant et un cycle en chaînage arrière.

5.4.2.2 Les modes d'inférences

Le Modus ponens: le Modus ponens peut être exprimé d'un point de vue pratique par si "P implique Q" est vrai et si P est vrai alors Q est vrai. Le chaînage avant correspond étroitement à ce mode d'inférence.

La spécialisation universelle: la spécialisation universelle peut s'exprimer ainsi: si "P(x) implique Q(x)" est vrai quel que soit x, alors "P(o) implique Q(o)" est vrai. Certains systèmes autorisent des variables quantifiées universellement dans les règles seulement. Dans ce cas le mécanisme de base du chaînage avant s'appuie sur l'existence d'une règle dont le déclencheur est de la forme P(x) et le corps de la forme Q(x) où x est une variable, et sur l'existence d'un fait établi P(o) pour introduire Q(o) dans la base des faits établis. Il s'agit d'une combinaison du "Modus ponens" et de la "spécialisation universelle": si "P(x) implique Q(x) est vrai, pour tout x alors "P(o) implique Q(o)" est vrai, si P(o) est vrai, alors Q(o) est vrai.

Les systèmes qui fonctionnent par chaînage arrière, admettant des variables dans les règles utilisent la spécialisation universelle pour invoquer une règle dont le déclencheur est de la forme Q(x) (dont le corps est P(x)), lorsqu'il existe un problème Q(o). Dès qu'un problème P(o), antérieurement introduit par chaînage arrière via une règle R (dont P(o) est le corps, et Q(o) le déclencheur), est reconnu résolu on peut appliquer la spécialisation et le Modus ponens pour conclure que Q(o) est résolu.

Le principe de résolution: ce mode intègre et généralise la spécialisation universelle, le Modus ponens, et le Modus tollens. PROLOG qui admet des variables aussi bien dans les règles que dans les faits, utilise le principe de résolution. Par exemple si nous considérons la règle "P(u, f(s), f(u)) implique Q(b, g(u), u)", et le problème Q(y, g(b), c). En PROLOG ils seraient représentés par: Q(b, g(u), u) $\longrightarrow$P(u, f(s), f(u)); $>$Q(y, g(b), c).*

L'application du principe de résolution permet d'inférer le problème $>$P(b, f(a), f(c));.

Extension du Modus ponens et de la spécialisation universelle: dans les systèmes experts traitant l'information imprécise ou incertaine le Modus ponens et la spécialisation universelle font l'objet d'extension. Par exemple dans un système, manipulant des faits et règles incertains, si on une règle "P implique Q" qui n'est certaine qu'à un degré I_1, et si P n'est certain qu'à un degré I_2 alors par Modus ponens extensionné Q ne sera certain qu'à un degré $I_1 * I_2$.

On pourrait aussi avoir que "P implique Q" est certain mais que P est imprécis, alors d'un fait P_o proche de P on pourra inférer un fait Q_o proche de Q.

Dans les systèmes experts traitant des connaissances incertaines ou imprécises, le processus d'inférence ne peut se limiter à la mise en oeuvre de chaque règle, isolément des résultats antérieurs. On doit

---

* Syntaxe de Prolog II.

faire appel à d'autres modes d'inférence pour agréger les résultats partiels issus de l'exploitation de règles expertes. Par exemple (comme dans MYCIN) si l'une des règles affecte à F un degré de certitude I_1, et une autre un degré de certitude I_2, alors F sera affecté du degré de certitude $I_1+I_2-I_1*I_2$.

De plus il est nécessaire dans ces systèmes d'assouplir le modèle de compatibilité entre déclencheurs et faits. Il est naturel de tenir compte dans l'étape de résolution des conflits de la plus ou moins grande compatibilité entre règles et faits. Si la règle est déclenchée, il faut tenir compte dans l'ajustement de la conclusion du degré de compatibilité qui a conditionné le déclenchement.

5.4.2.3 Stratégies de développement de la recherche

Aux modes d'invocation des règles on superpose une stratégie de développement de la recherche. Ces stratégies sont soit un développement "en profondeur d'abord", soit "en largeur d'abord".

La stratégie de développement de la recherche joue un rôle-clé dans la définition de l'étape restriction des moteurs d'inférence. En chaînage arrière seuls les faits à établir sont à comparer aux règles lors de l'étape de filtrage, c'est-à-dire que l'ensemble des faits issus de l'étape de restriction est réduit aux seuls faits à établir. Si en outre on suit une stratégie de développement en profondeur l'ensemble des faits à établir sera réduit aux faits à établir issus du dernier déclenchement de règle. De plus, si on considère les faits à établir, issus du dernier déclenchement, strictement dans l'ordre où ils sont écrits dans la règle, l'ensemble des faits à établir est réduit à un singleton: le premier fait à établir issu du dernier déclenchement de la règle.

De plus, la recherche en profondeur d'abord induit un type particulier de résolution des conflits: l'ensemble des conflits de l'étape

filtrage n'est pas explicitement formée; on déclenche la première règle retenue de l'étape du filtrage, elle-même issue de l'étape restriction qui dépend de la stratégie de développement de la recherche. Éventuellement celle-ci peut engendrer un retour-arrière donc la recherche d'une autre règle déclenchable. Dans certains cas on forme l'ensemble des conflits, et on choisit arbitrairement la première règle à déclencher, et la suivante dans le cas d'un retour-arrière.

Dans certains systèmes on utilise une fonction heuristique intrinsèque au moteur attachant une valeur au déclenchement de chaque règle. Dans d'autres systèmes, on fait usage de "métarègles" pour résoudre l'ensemble de conflits; celles-ci doivent être définies par l'usager.

Finalement, disons qu'une stratégie peut être à la fois en profondeur ou en largeur et irrévocable (irrévocable: les règles choisies lors de la résolution des conflits ne sont jamais remises en cause). Une stratégie peut être en profondeur ou en largeur et par tentatives. Les stratégies par tentatives peuvent essayer l'application d'une règle ou d'un ensemble de règles puis éventuellement remettre en cause, ultérieurement, cette application. On dit alors qu'on opère un retour-arrière. Certains moteurs disposent de primitives utilisables dans le corps des règles pour commander des retours-arrières (PROLOG, ARGOS). Généralement les moteurs à régime par tentatives peuvent fonctionner en régime irrévocable.

Enfin disons que le développement en profondeur est plus souvent utilisé parce que plus facile à programmer.

5.4.2.4 Monotonie et non-monotonie

D'une manière pratique on dit qu'un système est monotone à partir de faits et règles, par application des modes d'inférence autorisés on

infère de nouvelles connaissances (faits ou règles) qu'on adjoint aux précédentes, sans qu'il n'y ait lieu d'en éliminer. En d'autres mots lorsque la valeur de vérité d'une connaissance a été établie une fois, elle ne change plus. Les systèmes de diagnostic construits jusqu'à présent sont le plus souvent monotones.

Néanmoins de nombreux systèmes fonctionnent d'une façon non monotone. Ces systèmes ont, le plus souvent, un langage qui offre des primitives qui permettent de supprimer, ou inhiber provisoirement des connaissances. Les situations rencontrées sont de types suivants:

- les situations de diagnostic interactif qui peuvent conduire à modifier la valeur de vérité d'un fait;
- les situations ou l'on essaie plusieurs lignes de raisonnement en échafaudant des hypothèses;
- les situations où l'on exploite des règles expertes simplement plausibles dites "règle par défaut".

5.4.2.5 Structuration de la base des connaissances

L'étape du filtrage a pour but de déterminer l'ensemble de conflits. Dans le but de diminuer la part du temps d'exécution du programme lors du filtrage, certains langages offrent des techniques permettant de limiter le nombre de règles et faits à examiner.

Dans ARGOS la base des faits est structurellement partitionnée en une "base des problèmes" et une "base de faits établis". En outre à chaque règle est associé un indicateur signifiant que la règle est "active" ou "inhibée", seules les règles actives sont sujettes au filtrage. Le passage d'une règle de l'état actif ou inhibé peut être commandé par des primitives du langage de rédaction des connaissances expertes.

Dans CENTAUR on partitionne les règles. On distingue les classes suivantes: les "règles de déclenchement", les "règles d'inférences", les "règles d'affinement", et les "règles de synthèse".

Dans SPHINX on distingue: les "règles d'évocation", les "règles de connaissances diagnostiques ou thérapeutiques", les "règles d'interprétation d'examens", les "règles de résolution de conflits".

Le moteur de "MYCIN" peut faire correspondre à chaque fait à établir l'ensemble des règles susceptibles d'être déclenchées pour apporter un élément d'information le concernant.

Il existe des méthodes plus sophistiquées que celles énumérées précédemment, qui structurent la base de connaissances. Les méthodes sont:

- la méthode des filtres,
- la méthode des réseaux,
- la méthode implantée dans PSC.

Ces méthodes très brièvement reposent sur l'hypothèse suivante: dans beaucoup de cas, les modifications de la base des règles sont très peu nombreuses par rapport au nombre de cycle de base des moteurs; entre deux de ces modifications, il peut alors être intéressant d'effectuer un prétraitement des règles. Ce prétraitement consiste en des tests qui déterminent les règles déclenchables, pour ensuite regrouper les règles dont les tests sont communs. Chaque fait de la base originelle a été utilisé aux tests du réseau. Les instances complètes sont susceptibles de former l'ensemble de conflits. Ce processus est répété à chaque règle en tenant compte des modifications sur la base des faits. Ces dernières méthodes cependant risquent de compenser le gain en temps de filtrage par une consommation prohibitive de l'espace-mémoire.

5.5 EXEMPLE DE FONCTIONNEMENT D'UN MINI-SYSTÈME EXPERT

Considérons la base de connaissances de la figure 5.5.1. La base des faits comporte quatre faits: trois faits sont établis, et un fait reste à établir. On suppose qu'il n'existe pas dans la base des faits un fait établi C, en même temps qu'un fait à établir C. De plus, on suppose que dans une règle, les conditions de déclenchement sont à gauche, et les effets sur la base des faits sont à droite.

On considère un moteur d'inférence ayant un cycle de base avec les caractéristiques suivantes:

Phase d'évaluation

1- Étape de restriction: aucune restriction ne sera opérée c'est-à-dire qu'au premier cycle les faits et les règles utilisés seront respectivement ceux de la base des faits et de la base des règles. Pour les cycles suivants les faits et les règles seront respectivement issus de la base des faits et de la base des règles du cycle antérieur.

2- Étape de filtrage: les conditions de déclenchement sont satisfaites dès que les symboles présents dans le membre gauche de la règle est présent dans la base des faits.

3- Étape de résolution de conflits: les règles qui sont applicables après l'étape de filtrage constituent l'ensemble de conflits, on convient que l'ordre des règles dans l'ensemble de conflits seront soumises au même ordre que les règles dans la base des règles; la première règle de l'ensemble de conflits sera systématiquement choisie pour le contrôle.

Phase exécution

La phase exécution consistera à introduire dans la base des faits les symboles figurant dans le membre droit de la règle (la règle issue de

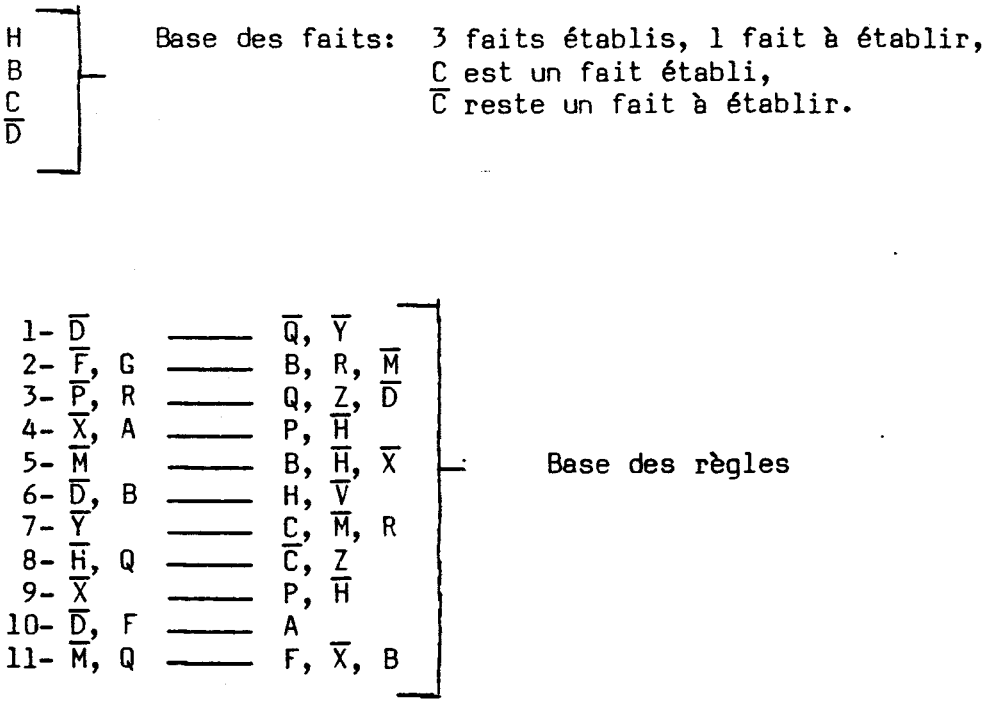


FIGURE 5.5.1

BASE DES CONNAISSANCES DU MINI-SYSTÈME EXPERT

la résolution de conflits), et à éliminer de la base des faits les symboles représentatifs des faits à établir. Plus précisément on a:

- si le membre droit comporte un symbole $\overline{S}$ ou S alors que S est déjà présent dans la base des faits, dans ce cas on n'ajoute rien dans la base des faits;

- si le membre droit comporte un symbole S alors que $\overline{S}$ est déjà présent dans la base des faits, dans ce cas on ne rajoute rien dans la base des faits;

- si le membre droit comporte un symbole S alors que $\overline{S}$ est présent dans la base des faits alors on ajoute S, et on élimine $\overline{S}$.

Retour-arrière

Le moteur adoptera une stratégie de retour-arrière tant que la base des faits comportera des faits à établir. Si au cycle n' l'ensemble de conflits est vide alors au cycle n+1 on affectera la base des faits et la base des règles du dernier cycle antérieur ayant donné lieu à l'exécution d'une règle.

Arrêt du moteur

Le moteur s'arrêtera si la base des faits ne comporte que des faits établis.

On donne à la figure 5.5.2 sous forme d'un graphe l'enchaînement des cycles du moteur à partir de la base des faits et la base des règles telles que données en figure 5.5.1.

1er cycle:
- les règles 1 et 6 sont applicables;
- la base des faits devient $\overline{H}$, B, C, D en appliquant la règle 1;
- après application de la règle 1, l'ensemble de conflits de ce cycle est la règle 6.

2e cycle:
- les règles 7 et 8 sont applicables;
- la base des faits devient $\overline{H}$, B, C, $\overline{Q}$, $\overline{Y}$ en appliquant la règle 7;
- après application de la règle 7, l'ensemble de conflits de ce cycle est la règle 8.

.
.
.
.
.

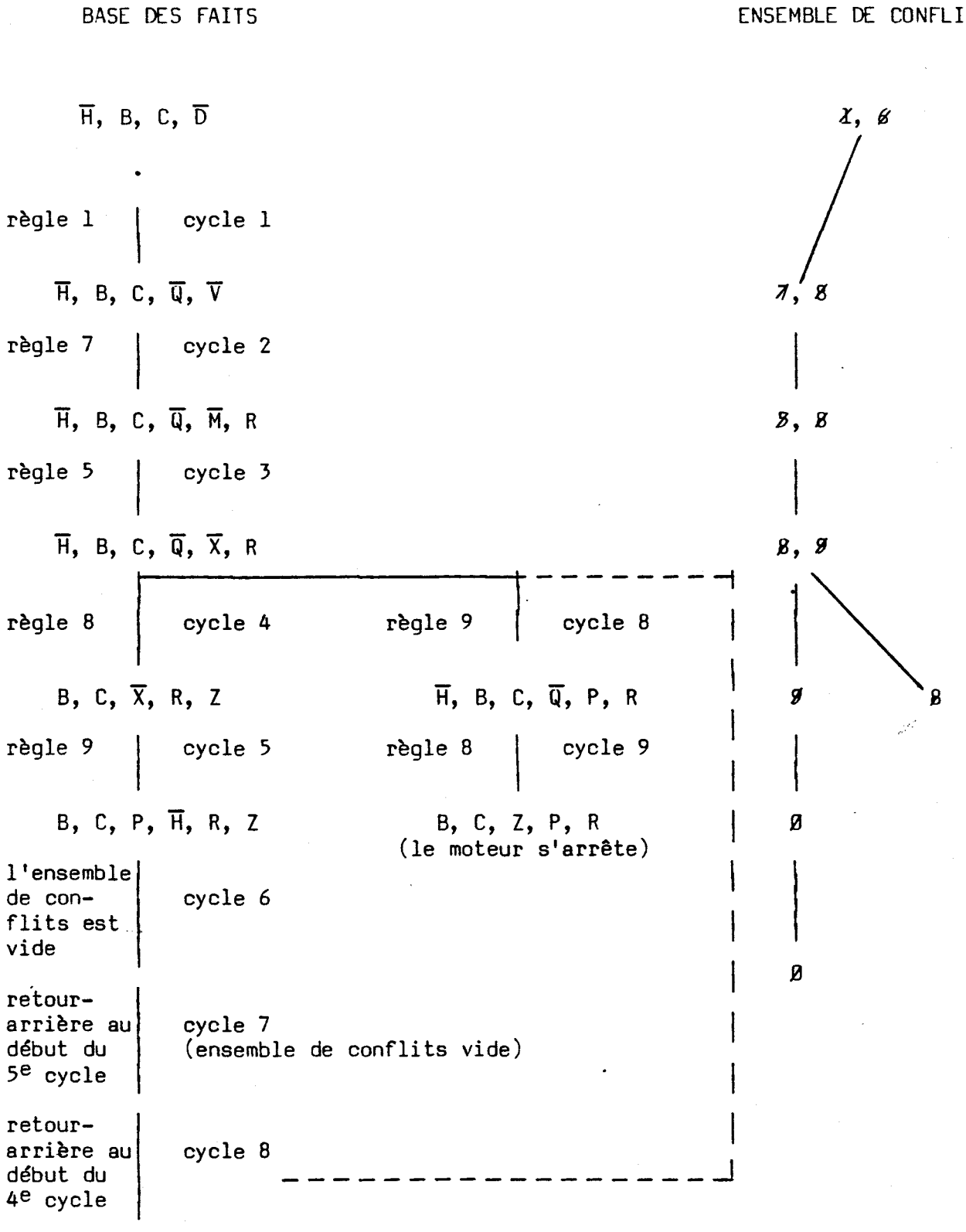


FIGURE 5.5.2

ENCHAÎNEMENT DES CYCLES DU MOTEUR DU MINI-SYSTÈME EXPERT

5e cycle:
- la règle 9 est applicable;
- la base des faits devient B, C, P, $\bar{H}$, R, Z;
- après application de la règle 9, l'ensemble de conflits de ce cycle est vide.

6e cycle:
- aucune règle est applicable, alors on a un retour-arrière au cycle suivant.

7e cycle:
- on retrouve la base des faits du début du plus récent cycle ayant donné lieu à une phase exécution;
- la base des faits devient B, C, X, R, Z;
- l'ensemble de conflits de ce cycle est vide alors nouveau retour-arrière au cycle suivant.

8e cycle:
- on restaure la base des faits du début du cycle antérieur au cycle 5;
- la base des faits devient $\bar{H}$, B, C, $\bar{Q}$, $\bar{X}$, R;
- l'ensemble de conflits est la règle 9;
- on applique la règle 9;
- la base devient H, B, C, Q, P, R;
- l'ensemble de conflits devient vide.

9e cycle:
- la règle 8 est applicable;
- la base des faits devient B, C, Z, P, R;
- il ne reste plus de fait à établir;
- le moteur s'arrête.

Le moteur que nous venons de décrire utilise un chaînage mixte; il se sert de faits établis et de faits à établir pour provoquer l'enchaînement des règles.

6.0 LES AUTRES APPLICATIONS DE L'INTELLIGENCE ARTIFICIELLE

6.1 LA RECONNAISSANCE DE L'ÉCRITURE

À ce jour, de nombreux systèmes de lecture de textes imprimés ou dactylographiés ont déjà été commercialisés. Mais, ils sont encore coûteux et ne peuvent traiter n'importe lequel type de texte sans préavis. Il faudra prendre en compte de nombreuses connaissances linguistiques pour progresser notablement. Le but ultime de la recherche actuellement est de mettre au point des machines capables de reconnaître le langage écrit et de traiter les informations qu'on leur fournit sous cette forme. Ces recherches s'inscrivent dans le cadre général de la vision artificielle et de la reconnaissance des formes et nous retrouvons un certain nombre de points communs entre la reconnaissance de caractères imprimés et les systèmes de vision artificielle des robots.

Les premiers systèmes de reconnaissance de l'écriture, mis au point dans les années '50, étaient des systèmes de lecture optique. Il existe aujourd'hui de nombreuses variantes commercialisées de ces systèmes. De nombreuses administrations et imprimeries utilisent les lecteurs optiques pour enregistrer et éditer leurs textes dactylographiés. Ces systèmes imposent cependant de nombreuses contraintes: la forme des caractères doit être précise et ne doit pas varier dans le temps.

La reconnaissance de l'écriture manuscrite est un problème beaucoup plus complexe, du fait de son infinie variabilité. La mise au point de systèmes de reconnaissance nécessite de collecter et d'étudier très finement de très grosses banques de données contenant des centaines de milliers de caractères, de façon à déterminer les paramètres vraiment représentatifs de l'écriture.

6.1.1 La reconnaissance en temps réel

Comme tout système de vision artificielle, un système de reconnaissance de l'écriture est constitué de capteurs qui enregistrent les données, lettres, chiffres ou dessins, et d'un ordinateur qui analyse et interprète ces données. Dans les systèmes de reconnaissance de l'écriture en temps réel, les capteurs doivent être capables de transmettre les données en temps réel à l'ordinateur; on utilise le plus souvent des tablettes graphiques sensibles sur lesquelles on écrit à l'aide d'un stylo très spécial. Les mouvements du stylo sont enregistrés magnétiquement ou électrostatiquement et transmis à l'ordinateur sous forme d'une succession de points. Les applications de ces systèmes correspondent aux domaines de la cartographie, l'enseignement de l'écriture, dessin assisté par ordinateur, ou bien à l'authentification des signatures. Après avoir été saisie par les capteurs, les lettres sont en mémoire de l'ordinateur sous forme de tableaux de points ou de vecteurs, ceux-ci subissent un premier traitement qui est une simplification de chaque caractère destiné à alléger les calculs dans les étapes ultérieures de la reconnaissance. Le système, doit ensuite identifier les caractères qu'il enregistre en se référant aux caractères qui ont été introduits dans sa mémoire au cours d'une phase d'apprentissage.

Cette phase d'apprentissage est une étape intermédiaire d'analyse et de paramétrisation des caractères. Elle consiste en une description résumée de chaque lettre. C'est une description mathématique simplifiée. Chaque lettre sera par exemple caractérisée par le nombre et l'orientation des segments de droite, et des arcs de cercle qui la compare, le nombre et l'emplacement de rebroussement qu'elle comporte, etc. Le choix de ces paramètres doit être un compromis entre la qualité de la reconnaissance, et la limitation du temps de calcul.

6.1.2 La reconnaissance des caractères d'imprimerie

Le processus d'analyse est assez simple lorsque les caractères sont imprimés car ils sont normalisés. La technique la plus courante, est l'utilisation des masques. Le masque est une matrice de points de petite taille représentant un motif qui peut éventuellement se trouver dans la lettre; on déplace le masque sur toute la surface de la lettre où l'on recherche la présence de ce motif. Ainsi, l'une des manières d'identifier la lettre A est de balayer la surface de la lettre à l'aide d'un masque représentant un segment incliné à 45° vers la droite, qui correspond à la partie gauche du triangle supérieur de la lettre. Il est également possible d'utiliser une technique de balayage du caractère par des lignes droites. Le programme calcule alors le nombre d'intersections entre la droite et le caractère enregistré, détermine si la concavité des arcs de cercle est orientée vers la droite ou vers la gauche. D'autres techniques consistent à analyser la structure globale de la lettre en lui faisant subir la transformée de Fourier pour en ressortir certains détails.

6.1.3 La reconnaissance des caractères manuscrits

Dans ce cas, le texte est d'abord segmenté en une série de tracés continus, segments de droite ou arcs de cercle que l'on appelle "primitives de base". On caractérise ensuite les relations qui existent entre ces éléments de base (point de rebroussement, changement de carbure) par une seconde série de primitives. Enfin un troisième niveau de primitives traduit la position relative et les relations qui existent entre les différentes parties du tracé. Finalement, un système de classement des primitives qui classifie chaque fait (droite, courbe, arc de cercles) en classe de longueur et en classe de courbure.

6.1.4 Étape finale du processus de la reconnaissance des caractères d'imprimerie et manuscrits

L'étape finale du processus de la reconnaissance est l'identification des caractères. Cette identification est effectuée par comparaison avec les caractères de référence. Les caractères de référence sont introduits par une phase préalable d'apprentissage.

Au cours de la phase d'apprentissage, le système détermine les caractéristiques des différentes classes de formes qu'il sera amené à identifier. Il stocke ces caractéristiques en mémoire. Il existe deux méthodes d'apprentissage: la méthode de type statistique, et la méthode de type structurel.

Les méthodes de reconnaissance statistique considèrent chaque forme à reconnaître comme un ensemble de paramètres statistiques. Chaque paramètre est caractérisé par la position le long d'un axe dans un espace de mesure qui comporte autant de dimensions que de paramètres. Chaque caractère est donc représenté par un point dans cet espace. Au cours de la phase d'apprentissage le système définit les régions de l'espace (en termes de distribution de probabilités) qui correspondent aux différentes représentations possibles d'un même caractère. Au cours de la phase de reconnaissance le système calcule les paramètres des caractères qu'il doit identifier, en déduisant la classe à laquelle le caractère appartient et donc la probabilité qu'il s'agisse de tel ou tel caractère. En pratique on préfère diviser l'espace en sous-espaces indépendants correspondant à chacune des classes de caractères. Ce type de méthode est surtout utilisé pour la reconnaissance de caractères imprimés ou pour la reconnaissance globale de mots manuscrits.

Les méthodes structurelles de reconnaissance de formes s'apparentent plutôt au processus d'analyse syntaxique d'une phrase. Chaque caractère est considéré comme un assemblage de formes élémentaires ou

primitives. Au cours de la phase d'apprentissage, le système détermine ces formes élémentaires par segmentation du caractère et se constitue ainsi une sorte de grammaire de description des caractères. Au cours de la phase reconnaissance il se réfère à cette grammaire pour identifier les caractères inconnus.

Il est possible d'éliminer un certain nombre d'erreurs pour l'identification des mots en utilisant des informations statistiques sur la langue considérée: en français par exemple il est très rare qu'un mot contienne deux u successifs ou des suites de consonnes telles que bpcl, ou des consonnes triplées. La chaîne de caractères représentant le mot est reconnu par identification avec un lexique.

6.2 LA RECONNAISSANCE DE LA PAROLE

Les applications qui ont déjà vu le jour ou qui sont en cours de développement, reposent sur des méthodes limitées de reconnaissance, utilisables seulement dans le cadre de simplification considérable: vocabulaire réduit et généralement spécialisé, prononciation séparée des mots, souvent par une seule personne. Le langage parlé naturellement présente un certain nombre de caractéristiques qui en rendent le traitement très complexe.

Les caractéristiques sont les suivantes:

- l'absence de silence entre les mots pose le problème de la segmentation d'un flat continu de mots enchaînés;

- la très grande variabilité que peut présenter un même discours selon la façon de parler du locuteur;

- le locuteur lui-même (enfant, femme, homme);

- les accents régionaux;

- les perturbations apportées par l'environnement;

- le fait que la parole résulte d'une combinaison d'informations de caractères très différents, se situant aux niveaux phonétique, phonologique, syntaxique, sémantique et pragmatique, multiplie les critères de décisions;

- l'absence de règles précises permettant de reconnaître sans ambiguité les informations se situant aux différents niveaux, et même d'en séparer certains qui sont fortement liés;

- la nécessité de faire appel à des connaissances relevant des domaines très différents (physiologique, psychologique, linguistique, acoustique, informatique, traitement du signal, etc.) et de traiter un grand volume de données pour dégager les informations pertinentes à la reconnaissance de la parole.

Il existe deux approches pour réaliser des systèmes de reconnaissance de la parole:

- la première fait appel aux hypothèses simplificatrices les plus importantes; elle consiste à reconnaître globalement des mots isolés ou enchaînés appartenant à des vocabulaires réduits et prononcés par une seule personne;

- la seconde permet d'aborder le problème de la reconnaissance de la parole continue éventuellement par plusieurs locuteurs en procédant en deux étapes:

  - en premier lieu, la reconnaissance des sons élémentaires qui composent la langue;
  - en second lieu, l'exploitation d'informations de niveau supérieur (les mots possibles, les structures de phrase énonçables) pour reconnaître la phrase énoncée en dépit des erreurs commises au niveau de la reconnaissance des sons.

La seconde approche a été étudiée dans le cadre d'un projet baptisé ARPA-SUR finance de 1971-1976 par la DARPA (Defence Advanced Research Project Agency) organisme du département de la défense aux Etats-Unis. L'un des cinq laboratoires participants a réalisé les objectifs du projet (Université Carnegie Mellon de Pittsburgh avec l'objectif de 1 000 mots différents prononcés par cinq personnes différentes) avec le système HARPY. Malgré ce succès aucun système n'est encore commercialisé.

Par contre, les systèmes utilisant la procédure de reconnaissance globale sont de plus en plus nombreux à être en service dans le monde.

Quelle que soit l'approche, le processus de reconnaissance de la parole commence par une analyse du signal vocal, préalablement convertit en signal électrique. Le signal varie en fonction du temps. Ce dernier est convertit en signal numérique du temps c'est-à-dire qu'une suite de nombres mesurant l'amplitude du signal à des intervalles de temps rapprochées (1/10 000 de seconde). Ce signal digitalisé est dans certains systèmes directement utilisé par le dispositif de reconnaissance. Le plus souvent le signal est représenté sous une forme spectrale obtenue en mesurant le signal pendant des intervalles de temps plus larges (10 millisecondes), les éléments des différentes fréquences, et leur amplitude. On utilise la transformée de Fourier du signal numérique pour la représentation spectrale. Cette paramétrisation permet de réduire considérablement le flux d'information représenté par le signal vocal, et d'éliminer les redondances présentes dans celui-ci.

6.2.1 L'approche de la reconnaissance globale

Les systèmes de reconnaissance globale sont caractérisés par le fait qu'une phase d'apprentissage est requise. Pendant cette phase,

l'utilisateur prononce les mots de son application. Pour chaque mot, une analyse acoustique (paramétrisation acoustique) est effectuée, et les résultats sont stockés en mémoire.

Lors de la phase de reconnaissance, pour chaque mot prononcé par l'utilisateur, le système effectue la même analyse que précédemment. Les paramètres des mots sont comparés à tous les paramètres des mots de référence. Le mot le plus ressemblant est alors affiché. Par "programmation dynamique" on détermine la façon optimale de mettre en correspondance les deux mots à comparer.

Les avantages de la reconnaissance globale sont:

- excellente capacité de reconnaissance;
- son indépendance vis-à-vis des particularités de la langue à reconnaître.

Les inconvénients:

- son vocabulaire est encore limité à cause de la mémoire nécessaire pour stocker les caractéristiques des mots de référence;
- le temps de calcul demandé par le processus de reconnaissance;
- la prononciation en mots séparés est peu naturelle, et le système ne peut être utilisé que par un seul locuteur.

6.2.2 L'approche de la reconnaissance analytique

Les systèmes de reconnaissance analytique se distinguent d'abord des systèmes globaux par un décodage acoustique du signal vocal beaucoup plus élaboré, comprenant quatre phases et réalisé par un module acoustique:

1ere phase: l'analyse acoustique qui fournit une paramétrisation numérique de la phrase prononcée constitue la première phase (paramétrisation par transformée de Fourier).

2e phase: la segmentation qui vise à décomposer le signal en une suite de sons élémentaires différents (plusieurs méthodes peuvent être utilisées mais il s'agit généralement de détecter les zones de stabilité (phonèmes) et d'instabilité (transition entre phonèmes)).

3e phase: l'extraction des "indices pertinents" c'est-à-dire la sélection des caractéristiques qui serviront à la reconnaissance des différents segments détectés. Dans les cas les plus courants il s'agit d'une description condensée du spectre (pour chaque type de segment il s'agit d'identifier les caractéristiques des paramètres de la transformée de Fourier du signal) et ce aux instants de stabilité, ou bien aux instants d'instabilité.

4e phase: (en parallèle avec la 3e phase) l'extraction des informations "prosodiques" c'est-à-dire l'information ayant trait à des particularités de la prononciation. Ces informations sont de trois types:

- la variation de la fréquence fondamentale de la voix,
- l'intensité,
- son rythme (durée de syllabes).

Ces données sont transmises à un module prosodique spécialisé capable éventuellement d'y détecter des informations qui viendront s'ajouter au résultat de l'analyse des phonèmes.

En dehors du module acoustique, les systèmes acoustiques font appel à une succession d'autres modules: le module phonétique, le module phonologique, le module lexical, le module syntaxique, le module sémantique, le module pragmatique.

Le module phonétique joue un rôle important, il a pour fonction de traduire la liste des indices pertinents en une suite de phonèmes (la langue française comporte 30 phonèmes).

En général pour chaque segment on ne parvient pas à une liste unique de phonèmes, mais plutôt à une probabilité de présence de chacun des phonèmes de la liste. Il faudra choisir parmi la liste en utilisant des informations des niveaux supérieurs au niveau phonétique.

Le module phonologique porte sur les phénomènes de la langue qui se traduisent par le fait que le contenu phonétique théorique des mots est modifié dans une articulation rapide ou bien en fonction du contexte. Les règles correspondantes sont souvent induites soit par la difficulté de prononciation de certaines suites de phonèmes ou soit par leur inesthétisme. On a les catégories de règles suivantes:

- les règles de liaisons,
- les règles d'élisions,
- les règles de la dénasalisation.

C'est à ce niveau que retraite les cas des accents en fonction de l'appartenance géographique au socio-linguistique de chaque locuteur.

Le module lexical fait intervenir les informations sur les mots qui composent la langue. Il faut y inclure les mots communs, les verbes et leurs conjugaisons, les noms propres, les mots rares, les mots techniques, les mots qui se prononcent différemment dépendant des règles phonologiques. Chacun doit être accompagné d'informations sur sa catégorie grammaticale (son genre, son nombre, etc.).

Le module syntaxique est supposé renfermer les règles de la grammaire qui permettent d'écrire et d'analyser le langage, en termes grammaticaux et fonctionnels.

Le module sémantique traite du ou des sens des mots tels qu'on peut les trouver dans un dictionnaire et des relations entre eux.

Le module pragmatique a pour tâche de déterminer le sens d'une phrase dans le contexte de l'application.

On distingue deux catégories de modules, les modules de niveau supérieur, et les modules de niveau inférieur. Les modules de niveau inférieur correspondent aux modules acoustique et prosodique. Les autres sont de niveau supérieur, et relèvent de la problématique de la "compréhension du langage naturel".

6.2.3 Les progrès de la recherche en reconnaissance globale

Une évolution intéressante est la réalisation des systèmes dits "multilocuteurs multiréférences" capables de reconnaître (dans le cas d'un vocabulaire restreint d'une application) des mots prononcés par une personne quelconque. Dans ces systèmes l'apprentissage de chacun des mots du lexique demande la prononciation de ce mot par un grand nombre de locuteurs, de sexe, de timbre et de variétés dialectales différents. Des algorithmes de classification automatique permettent de déterminer pour chacun des mots des classes de prononciation et de ne retenir qu'un représentant pour chacune de ces classes.

Un autre développement important porte sur l'application des méthodes globales à la reconnaissance de mots prononcés sans silence intermédiaire. Une question jusqu'alors abordée par les techniques analytiques. Certaines difficultés proviennent de la co-articulation et de la segmentation du continum sonore. Pour résoudre ces difficultés, on utilise un algorithme de programmation dynamique à deux niveaux permettant de mettre en correspondance dans un premier temps les spectres des mots qui semblent être prononcés et ceux des références du lexique; dans un deuxième temps, l'ensemble de la phase à reconnaître avec les combinaisons possibles de mots autorisés par la syntaxe.

D'autres progrès sont liés à des résultats obtenus en psycholinguistique. Ces résultats ont montré que nous sommes capables d'identifier un mot avant la fin de sa prononciation. De plus elles ont établi qu'une erreur phonétique affectant le début d'un mot rend la reconnaissance beaucoup plus difficile qu'une erreur commise sur un phonème en fin d'un mot. Enfin que la détection d'un phonème cible est réalisée plus lentement en début de mot. Ces résultats ont conduit à la proposition d'un modèle basé sur l'identification des mots à partir de la reconnaissance de la zone correspondant aux premiers phonèmes du mot, selon une stratégie montante, et lorsque le nombre de mots est suffisamment réduit on vérifie l'identité des phonèmes suivants des mots retenus avec ceux de la phrase effectivement prononcée.

6.2.4 Les progrès de la recherche en reconnaissance analytique

L'une des approches est par l'utilisation des systèmes experts pour la lecture des sonogrammes. La connaissance qu'ont les phonéticiens de la lecture et du décryptage visuel des sonogrammes est difficile à traduire en une séquence de calculs déterminée. Le concept des systèmes experts est en revanche capable de prendre en compte ce type de savoir-faire. Le problème est qu'un phonéticien utilise ses connaissances en phonétique jusqu'au niveau de la perception visuelle des éléments pertinents, qui lui permettent de décoder le sonogramme. Il faut faire intervenir ce savoir jusque dans la conception du capteur chargé de lire le sonogramme.

Une autre approche consiste à utiliser des algorithmes de reconnaissance globale de mots enchaînés, bien maîtrisés aujourd'hui, mais en remplaçant l'entité du mot par des unités d'identification plus fines en tenant compte de l'information contextuelle fournie par les phonèmes environnants, et en particulier les transitions entre les phonèmes.

Enfin, une approche inspirée de la neurologie, est l'utilisation d'architectures parallèles. Celle-ci s'inspire des connaissances que l'on a des réseaux de neurones dans le cerveau humain. Elle apporterait une solution élégante à des problèmes entiers comme par exemple l'apprentissage de structures de la langue à partir d'exemples, l'accès lexical, le traitement d'informations incomplètes ou bruitées.

6.3 LA COMPRÉHENSION DU LANGAGE

Dans certains domaines comme la traduction automatique, ou toute application qui implique une action postérieure à la lecture, le système doit déterminer la structure puis le sens de la phrase après avoir identifié les mots.

Il n'existe encore aucun système capable de traduire automatiquement un texte manuscrit quelconque, et le nombre de systèmes capables de traduire des textes imprimés est très faible. Les machines existant à l'heure actuelle sont plutôt spécialisées dans des domaines techniques précis où il est possible de restreindre le vocabulaire et la taille du contexte.

Généralement, un système de traduction automatique procède selon plusieurs niveaux.

- Niveau morphologique: permet d'identifier les mots.

- Niveau lexical: les mots sont recherchés dans le dictionnaire.

- Niveau syntaxique: permet d'identifie les séquences de mots permises dans la langue considérée. Au cours de cette phase de l'analyse, le système peut à la fois prédire le ou les

mots susceptibles d'apparaître dans la phrase en fonction des mots déjà reconnus et vérifier que l'ordre des mots satisfait bien aux contraintes grammaticales de la langue. L'analyse syntaxique est parfois couplée à l'analyse sémantique, liée au sens des mots et aux concepts qu'elles sous-tendent.

- Niveau sémantique: qui traduit la structure de mots en une formule censée exprimer le sens du texte initial.

- Niveau pragmatique: permet de préciser ou de restreindre le sens d'un mot en fonction du contexte de la phrase et du domaine d'application considéré.

- Niveau d'exécution: qui confronte cette formule aux données stockées dans la machine afin d'élaborer une réponse.

La séquence de ces quatre niveaux est souvent appelée "phase de compréhension". Nous traitons dans l'ordre chacun des sous-problèmes qu'il contient.

Le niveau morphologique ne présente pas de grandes difficultés si le langage est écrit au contraire du langage parlé. Les techniques utilisées font partie de la reconnaissance de l'écriture et de la reconnaissance de la parole, et ont été décrites aux sections précédentes.

Le niveau lexical a pour mission d'établir si les mots identifiés au niveau morphologique existent vraiment. Le niveau lexical ne se limite pas à confirmer les hypothèses du niveau précédent, celui-ci prépare également les étapes suivantes, ce qui impose de faire des

choix concernant les informations à stocker dans le lexique. L'approche adoptée à ce niveau est d'établir une liste de catégories grammaticales plus ou moins exhaustives dépendant des applications. Par la suite on ajoute dans le dictionnaire, en face de chaque mot une liste de traits sémantiques que devront posséder les mots adjacents auxquels ils sont susceptibles de se rapporter.

Le niveau syntaxique a pour but d'identifier les séquences admissibles dans la langue considérée. Deux approches sont possibles, soit le modèle de la grammaire à contexte libre, si le niveau pragmatique est inexistant, soit le modèle de la grammaire dépend du contexte. La grammaire dite à contexte libre se présente sous la forme: 1- d'un ensemble de symboles terminaux (les mots) que nous écrivons en minuscules; 2- d'un ensemble de symboles auxiliaires (les catégories) écrits en majuscules (un de ces symboles, conventionnellement note s, sert de point de départ); 3- de règles de réécriture, de la forme $A \rightarrow \alpha 1, \alpha 2..., \alpha n$, dans lesquelles A est un symbole auxiliaire et les α des symboles terminaux. Par exemple $A \rightarrow ab$ signifie que l'une des règles admise par la grammaire consiste à substituer n'importe quelle apparition de A par la séquence ab. Par contre, dans la grammaire dépendante du contexte, seules les apparitions de A respectant certaines conditions du contexte peuvent donner lieu à réécriture.

Le niveau sémantique, est le plus délicat, il concerne le sens de la phrase. Lors de la résolution de sous-problèmes, le système doit fabriquer la formule censée exprimer le sens du texte. Différentes approches ont été proposées: l'"espace des sens", la "structure des sens", la "grammaire des sens".

L'espace des sens: a pour but de rendre compte des phénomènes de nuance ou de proximité sémantique qui se traduisent par un voisinage métrique. Mais la difficulté d'une telle approche est souvent de ne pouvoir trouver un outil concret de mesure de distance entre deux notions.

La structure des sens: a pour but de construire une structure à partir de sens primitifs. Comme par exemple "boire" c'est "ingérer un liquide".

La grammaire des sens: mène à concevoir un langage ayant à la fois une syntaxe rigoureuse et des règles d'interprétation univoques, elle aboutit surtout à postuler que tout texte d'une langue naturelle peut être traduit dans une expression de cette langue. On exploite la logique mathématique. Pour y parvenir, il faut traduire la structure construite par l'analyse syntaxique en une formule logique considérée comme exprimant la signification de la question posée par l'utilisateur. On utilise la technique suivante: à chaque règle de grammaire est associée une procédure qui détaille la construction d'une partie de la formule logique. Ces procédures mises bout à bout dans l'ordre où les règles de grammaire ont été utilisées, constituent un programme dont l'exécution donne pour résultat la formule souhaitée.

Le niveau pragmatique: cherche à modifier le résultat de l'analyse sémantique par la prise en considération d'éléments au texte analysé. L'approche la plus en vogue est le système des "scripts". Les scripts sont des séquences d'événements correspondant au déroulement typique d'une action parfaitement banale; déroulement que l'auteur d'un texte n'a aucune raison d'expliciter. Plus récemment la notion "MOP" a généralisé celle du script en permettant de mettre en commun des éléments appartenant à des scripts différents.

6.4 LA VISION

Dans de nombreux domaines la faculté de voir permet un apport d'informations irremplaçables. Cet apport est d'autant plus important en robotique. Elle sert à construire un modèle de l'environnement. Ce modèle est alors utilisé pour prendre une décision en fonction de la tâche à accomplir: détermination de la nature, et la position d'un objet, compréhension de la disposition des objets dans une pièce.

Mais, l'une des principales difficultés lors de la mise au point de systèmes avec vision, est la formidable puissance de calcul qu'il faut mettre en oeuvre.

Un système de vision artificielle est une combinaison de capteurs et d'un système informatique. Les capteurs mesurent les intensités lumineuses de la scène observée et le système informatique traite et analyse ces données: il en extrait une description symbolique de son environnement. La plupart du temps, l'acquisition d'images est réalisée à l'aide de capteurs de type caméra de télévision.

Deux grandes classes de capteurs tridimensionnels ont été mis au point à ce jour:

- les premiers donnent des informations sur le relief d'un objet grâce à des mesures de distances relatives entre les points de cet objet;

- les secondes, que l'on préfère, permettent d'accéder à la distance absolue entre chaque point de l'objet et la capturer. Quatre démarches ont été envisagées et appliquées avec succès:

  1- l'utilisation d'un télémètre laser;
  2- la seconde technique, avec une source laser active, permet de suréclairer un certain nombre de points de la scène et d'en déterminer la distance absolue ou au capteur par triangulation.

  Les deux autres méthodes sont passives:

  3- la méthode par stéréovisionnement en oeuvre la stéréocorrespondance de deux images, par la détermination des couples de points issus du même point réel;
  4- celle-ci recueille une série d'images qui se succèdent dans le temps à l'aide d'une caméra qui effectue un mouvement continu. Cette méthode devrait prendre une grande importance dans la vision des robots mobiles.

Après l'acquisition des données par les capteurs, l'image se retrouve en mémoire dans l'ordinateur sous forme d'un tableau de nombre indiquant la luminosité ou bien sa répartition en trois couleurs fondamentales (bleu, vert et rouge). Le système de vision artificielle aborde maintenant l'étape du traitement de l'image. Le problème consiste à extraire de l'image un certain nombre de caractéristiques pertinentes qui vont permettre de façonner une représentation condensée et structurée synthétisant l'information présente dans la scène observée. Cette description intermédiaire fait intervenir des paramètres physiques ou géométriques bien définis. Après extraction des paramètres, ceux-ci seront ensuite analysés pour interprétation. Un certain nombre d'outils ont été mis au point jusqu'à maintenant.

6.4.1 Identification globale d'objets

L'identification d'objets en plans isolés et parfaitement contrastés est parfaitement maîtrisée et trouve aujourd'hui des applications industrielles.

Les procédures d'identification visent à caractériser l'objet par sa forme et suppose que celui-ci présente une réflectivité homogène. L'image de l'objet est alors analysée en regroupant des points de luminosité comparables à des ensembles homogènes appelés région. Cette segmentation est effectuée par seuillage. Cette analyse est suivie de l'étape identification. L'identification est effectuée par une méthode paramétrique: le système compare les caractéristiques des régions qu'il a déterminées sur l'image observée aux caractéristiques de représentations de références. Les représentations de références sont préalablement effectuées par un utilisateur au cours d'une séance d'apprentissage.

6.4.2 Utilisation de la théorie de la morphologie

Cette théorie permet de caractériser une forme par ses interactions avec des formes géométriques simples appelées "éléments

structurants". Une transformation issue de cette théorie permet de faire apparaître sur l'image que les zones sombres et étroites et squelettiser ces zones c'est-à-dire ne garder que leurs lignes médianes. Ces traitements spécifiques jouent donc le rôle de filtres de l'information et réduisent la quantité de données à interpréter au minimum.

6.4.3 L'identification des objets partiellement observés

L'identification des objets qui ne sont que partiellement observés ne peut plus s'appuyer sur un modèle global de l'objet. Cet objet doit donc être considéré comme un ensemble structuré de caractéristiques locales. On appelle ces méthodes d'identification des "méthodes structurelles". Plusieurs problèmes se posent lors de l'établissement de ces méthodes. Il faut d'abord choisir un vocabulaire adéquat des caractéristiques locales pour définir l'objet (segment de droite, arc de cercle, angle aigü, etc.) et d'autre part modéliser les relations géométriques existant entre les différents éléments de cet objet. Enfin, il faut mettre au point des procédures d'identification effectuant une recherche de caractéristiques locales obéissant à une structure géométrique attendue.

6.4.4 La problématique de la vision chez les robots mobiles

La conception d'un robot mobile qui évolue dans un univers non structuré pose de sérieux problèmes. Il s'agit d'appréhender des environnements complexes sur lesquels la connaissance a priori est faible (champs de perception non limité, présence d'objets inconnus...). Il doit pouvoir circuler intelligemment et sans collision avec son environnement: il doit s'en construire une représentation géométrique. Il doit acquérir un certain niveau de compréhension sémantique sur cet univers, comme par exemple identifier les objets sur lesquels il doit agir. Il doit analyser la structure de l'espace (portes, fenêtres, cadres ...).

Les difficultés sont d'une part, le passage de l'identification d'objets correspondant à un modèle précis connu du robot, à un niveau beaucoup plus général de la compréhension de l'espace, d'autre part la construction progressive des représentations de l'environnement au fur et à mesure des déplacements du robot, ce qui suppose que celui-ci soit capable d'en reconnaître les parties déjà vues, de s'y situer et de prendre en compte tous les problèmes d'incertitudes dans sa perception du monde.

6.5 L'APPRENTISSAGE

Le terme "apprentissage" fait référence ici à la conception de techniques d'acquisition automatique des connaissances symboliques. Les systèmes experts apprendront peut-être un jour leur métier eux-mêmes.

L'approche moderne de l'apprentissage, née de l'Intelligence Artificielle, est une discipline scientifique dont les premiers travaux de recherche ne remontent qu'à une dizaine d'années. Ces recherches visent à la simulation de toutes les formes d'apprentissage naturel.

Nous décrivons brièvement quelques formes d'apprentissage pour lesquelles des modèles ont été proposés.

Une forme d'apprentissage est l'apprentissage de procédure de classification. C'est la forme d'apprentissage de connaissances dans laquelle il s'agit d'intégrer un concept déjà défini et présenté à l'aide d'exemples et de contre-exemples. Il existe deux catégories de méthodes: les méthodes numériques et les méthodes symboliques. Ces deux catégories de méthodes consistent d'abord à prendre un ensemble d'exemples représentatifs des concepts définis que nous voulons étudier (ou une collection d'objets représentatifs des concepts). Chaque objet est décrit selon un certain nombre de paramètres.

Un certain nombre de méthodes numériques ont été proposées pour déterminer la liste ordonnée des critères (paramètres ou attribut) caractérisant le mieux un concept. Ces méthodes font généralement appel à une notion d'ordre au sens de la théorie de l'information, et à une mesure de cet ordre (l'entropie). On entend par ordre la séparation plus ou moins complète qu'introduit l'application d'un critère entre les exemples relevant de concepts différents. Le but est de parvenir à des sous-groupes parfaitement triés et homogènes. On évalue alors l'efficacité d'un critère par la distance entre les mesures du désordre initial et de celui qui reste après application du critère. Ce processus est répété sur les sous-groupes et en choisissant la succession de critères qui conduisent à la meilleure valeur du désordre. Cette méthode est appliquée dans EXPERT-EASE, un système expert commercialisé capable d'apprentissage.

Par contre l'approche symbolique ne se pose plus la question de savoir ce qui est le plus efficace, mais celle de savoir ce qui est le plus signifiant. Il s'agit donc de savoir si certains attributs ont plus de signification que d'autres. La mesure utilisée n'est plus le nombre relatif d'exemples triés par un critère mais la similitude entre les exemples.

Une autre forme d'apprentissage est la "généralisation". Cette forme a été utilisée dans le système expert LEX. Ce système apprend à partir d'exemples, à intégrer des fonctions mathématiques. Ce système est arrivé à un niveau de performance comparable à celui d'un homme entraîné.

LEX apprend à intégrer en générant automatiquement des problèmes c'est-à-dire des fonctions qu'il essaie d'intégrer. Il analyse ensuite sa démarche et détermine quelles heuristiques l'ont aidé à trouver un résultat et quelles autres l'en ont au contraire éloigné. Il utilise ces exemples et contre-exemples d'application des heuristiques pour affiner ces dernières, plus précisément la partie qui défini leur champ d'application. Un deuxième mode d'apprentissage de

LEX repose sur l'utilisation des contre-exemples. Ayant découvert que une règle n'était pas applicable, LEX modifie l'expression de la forme la plus générale acceptable de façon à interdire ce type d'expression tout en continuant d'accepter les formes un peu moins générales englobant les cas pour lesquels il est prouvé que l'heuristique est valable. LEX généralise en se référant à une hiérarchie de généralisation prédéfinie qui indique les relations de parenté entre tous les types de fonctions qu'il est susceptible de rencontrer.

La dernière forme que nous citons est l'"apprentissage inventif". Cette forme tente de concevoir des mécanismes de génération automatique de fonctions répondant à certains critères. Cette approche fait l'objet de recherches très actives en Intelligence Artificielle sous le nom de "programmation automatique" puisqu'il s'agit d'écrire automatiquement un programme réalisant certaines fonctions données. Cette discipline encore nouvelle fait appel à des outils mathématiques originaux. Elle fait un usage abondant de la notion de fonction récursivement définie c'est-à-dire dont la définition fait référence à elle-même.

7.0 INVENTAIRE DES SYSTÈMES EXPERTS COMMERCIALISÉS OU EN VOIE DE DÉVELOPPEMENT

Le présent chapitre se veut un inventaire assez complet des systèmes experts et généraux existant principalement au Canada et aux États-Unis et en Europe. Des systèmes appartenant à d'autres catégories et d'autres provenances ont été répertoriés. Ces derniers ont été dénommés systèmes de l'Intelligence Artificielle.

On retrouve en annexe 1 le présent inventaire. Avant de consulter celui-ci il est préférable que le lecteur consulte les sections suivantes de ce chapitre.

7.1 LES SOURCES D'INFORMATION SUR L'INTELLIGENCE ARTIFICIELLE

La popularité de l'Intelligence Artificielle augmente sans cesse et plus particulièrement pour les systèmes experts. Des équipes de chercheurs s'organisent et celles déjà existantes travaillent davantage afin de concrétiser un système innovateur et compétitif. Actuellement on compte au-delà de mille systèmes de l'Intelligence Artificielle à l'échelle mondiale. Les États-Unis ont déjà plusieurs longueurs d'avance dans ce domaine. Les institutions telles le Massachusset Institute of Technology (MIT) à Cambridge, l'Université de Stanford en Californie, l'Université du Texas et l'Université de Carnegie-Mellon ne sont que quelques-unes des forteresses des États-Unis en matière d'Intelligence Artificielle. La France fait de gros efforts afin de combler l'écart qui la sépare des États-Unis. D'excellents systèmes y ont d'ailleurs été produits ainsi que des langages propres à la réalisation de ces systèmes. Bien que des recherches soient menées dans des institutions universitaires de Montréal, Ottawa, Toronto, Vancouver, etc., le Canada reste loin derrière ces deux géants.

Beaucoup de conférences sur l'Intelligence Artificielle ont vu le jour. La plus célèbre, à l'échelle mondiale, est l'International Joint Conference on Artificial Intelligence (IJCAI) débutée en 1969 et se tient toutes les années impaires. C'est lors de ces conférences que les systèmes les plus populaires et les plus prometteurs sont habituellement présentés. Le Canada a aussi ses conférences dont la Conférence canadienne sur l'Intelligence Artificielle qui se tient toutes les années paires. La dernière conférence a eu lieu à Montréal, à l'École Polytechnique, au mois de mai dernier. Il existe encore beaucoup d'autres conférences qui se tiennent à intervalles réguliers, à travers le monde.

Outre les conférences, les adeptes de l'Intelligence Artificielle peuvent devenir membre de sociétés, comme par exemple la Société canadienne pour l'étude de l'intelligence par ordinateur (SCEIO), au Canada ou s'abonner à diverses revues traitant de l'Intelligence Artificielle. Depuis plusieurs années, un périodique l'Artificial Intelligence, se veut la voie officielle de l'Intelligence Artificielle. Ceci nous amène donc à parler des volumes, en nombre impressionnant, traitant de l'Intelligence Artificielle. On retrouve maintenant le Hand Book de l'Intelligence Artificielle, de A. Barr et E.A. Feigenbaum; c'est d'ailleurs un des principaux volumes à consulter. Outre les rapports de conférences, plusieurs livres sont intéressants à consulter. En français on peut feuilleter les ouvrages de Alain Bonet [5], Henri Farreny [1], Marius Fieschi [4] et la cinquième génération [2] de E.A. Feigenbaum et P. McCorduck alors qu'en anglais on peut consulter ceux de E.A. Feigenbaum, P. McCorduck et E.H. Shortliffe pour ne nommer que ceux-là. Les ouvrages sont trop nombreux et les auteurs trop diversifiés pour qu'on puisse tous les mentionner. Cependant les auteurs mentionnés précédemment sont d'excellentes personnes ressources à consulter.

Plusieurs grandes revues francophones telles La Recherche, Pour la science et Micro-systèmes traitent fréquemment de l'Intelligence Artificielle et de ses nombreuses applications.

L'intelligence Artificielle est un sujet très en vogue pour les groupes de recherche. C'est pourquoi dans toutes les universités du Québec, ou du moins la grande majorité, on retrouve de la documentation sur cette science. À notre connaissance, l'endroit offrant la documentation la plus complète dans ce domaine, dans la région de Montréal, est l'Université McGill.

7.2 LES PARTICULARITÉS DES TABLEAUX

7.2.1 Présentation

Lors de l'évaluation des systèmes experts plusieurs critères doivent être considérés. Suite aux nombreuses lectures, les critères "essentiels" à l'évaluation des systèmes ont été mis sous forme de tableaux. Voici donc ces critères:

- NOM;
- AUTEUR(S);
- RÉFÉRENCE;
- ANNÉE DE PRÉSENTATION;
- ORIGINE (PAYS ET INSTITUT);
- APPLICATION;
- NOMBRE DE RÈGLES;
- MODE DE REPRÉSENTATION DES CONNAISSANCES: les connaissances peuvent être introduites de plusieurs façons dans les systèmes:

a) règles de production: couple situation-action;
b) réseaux sémantiques: ensembles de noeuds (concepts) reliés par des arcs (relations entre les concepts);
c) frames (schémas): généralisation des réseaux dans le cas où les éléments, objets sont plus complexes;
d) base de données: assertions qui ne sont pas exprimées sous forme d'implication;
e) logique des prédicats;

- LANGAGE;

- PHASE ACTUELLE: sert à déterminer si le système est en phase de recherche, de production ou s'il est fonctionnel;

- FONCTIONNEMENT: mode de chaînage utilisé:
  a) chaînage avant,
  b) chaînage arrière,
  c) chaînage mixte,
  d) chaînage bidirectionnel;

- PERFORMANCES;

- MOTEURS D'INFÉRENCE DÉJÀ EXISTANTS: certains systèmes experts utilisent des systèmes généraux, des dérivés d'autres systèmes experts;

- COMMENTAIRES GÉNÉRAUX: regroupe les principaux défauts, principales qualités et originalités, s'il y a lieu. Cette colonne sert aussi à décrire les parties du système, détailler davantage son application, donner la catégorie de moteurs*, etc.

---

* Représentée par deux états, 0 et 1.
0: moteur n'utilisant pas de variables.
1: moteur utilisant des variables (dans les règles seulement et non la base de faits).

7.2.2 Description des catégories

Dans le présent inventaire les systèmes sont classés par domaine d'application plutôt que par ordre alphabétique. Le but est de permettre une consultation plus rapide du document. On y retrouve de nombreuses catégories. Une de ces catégories (systèmes généraux) est sans domaine d'application. Ces systèmes, connus aussi sous le nom de systèmes-experts vides, comprennent des langages qui, de par leur conception, sont considérés comme des moteurs. On y retrouve quelques systèmes pour l'apprentissage des connaissances et des systèmes présentant des affinités pour un domaine particulier.

L'inventaire comprend entre autres des systèmes pour les interfaces en langue naturelle. Une liste de systèmes incomplets a été ajoutée. Cette liste renferme des systèmes qui ne pouvaient être classés dans une catégorie précise vu le manque d'information.

7.2.3 Obtention de l'information

Après avoir établi les critères pour la classification des systèmes, ce fut le début de l'inventaire. Quelque 300 lettres ont été expédiées un peu partout dans le monde afin d'avoir de plus amples informations sur les systèmes et obtenir de l'information sur les travaux les plus récents des auteurs. Jusqu'à maintenant au-delà de 400 systèmes ont été répertoriés (excluant le domaine de la robotique) au Canada, aux Etats-Unis et en Europe. Une quinzaine de systèmes répertoriés proviennent du Japon, de l'Australie, de la République Fédérale Allemande.

L'inventaire et l'analyse des systèmes experts existants demeurent assez complètes. Cependant une mise à jour fréquente s'impose.

7.3 ORIENTATION DES RECHERCHES

Les grands centres de recherche américains tels MIT et Stanford n'ont pas de domaine d'intérêt particulier. Stanford autant que MIT font des recherches dans des domaines très variés. Le Québec tente de développer des interfaces en langue naturelle française efficaces puisque les performances des systèmes existants dans ce domaine sont plutôt modestes; on en est à essayer de rattraper le retard pris sur les américains. La France elle non plus n'a pas d'orientation particulière. Dans tous les cas on essaie de trouver un domaine d'application qui pourrait s'avérer lucratif. Un seul domaine est fortement délaissé, celui du jeu. Une centaine de chercheurs seulement démontrent de l'intérêt pour cette partie qui a pratiquement donné naissance à l'Intelligence Artificielle.

De plus en plus les compagnies, peu importe leur origine, veulent acquérir des systèmes experts. C'est pourquoi il existe une grande compétition dans ce domaine. Une grande lacune des systèmes experts actuels est le temps de réponse, excessivement lent dans certains cas. C'est pour combler ce manque que les Japonais ont instauré leur projet d'ordinateurs de cinquième génération. Au début des années 1990 les Japonais croient pouvoir révolutionner le marché de l'Intelligence Artificielle.

7.4 ANALYSE GÉNÉRALE

La plupart des systèmes inventoriés proviennent des États-Unis et de la France, le Canada étant actif dans ce domaine depuis peu. Les systèmes développés actuellement sont de plus en plus compétitifs et emploient des techniques modernes. Les systèmes de vision sont aussi de plus en plus présents; ils remplaceront, bientôt, les entrées traditionnelles sur clavier et feront avancer d'un grand pas cette science relativement nouvelle (environ 25 ans). Quoi qu'il en soit, comme dans tout domaine, il reste beaucoup de place pour l'amélioration.

8.0 SYNTHÈSE, CONCLUSIONS ET RECOMMANDATIONS

8.1 CE QU'EST UN SYSTÈME EXPERT

Nous avons, dans la présente étude, exposé les différents principes de l'intelligence artificielle, et plus particulièrement nous avons mis une emphase sur une technique d'acquisition et de diffusion des connaissances résultant de la recherche en intelligence artificielle: les systèmes experts dénommés aussi système à base de connaissances.

Bien que les techniques issues de l'intelligence artificielle aient été utilisées depuis la fin des années 60 à des fins militaires, et de l'exploration de l'espace, celles-ci ne se sont répandues dans tous les secteurs d'activités économiques que depuis le début des années 80 (systèmes experts, interface en langue naturelle, etc.).

L'informatique des années 80, met à la disposition des utilisateurs d'informatique des technologies dites de "cinquième génération" (ce terme est emprunté du grand projet de recherche, en intelligence artificielle, lancé au Japon au début des années 80) issues des recherches en intelligence artificielle. Ces recherches ont pour but de créer une nouvelle génération d'ordinateur basé sur le traitement symbolique de l'information, de créer des environnements de programmation qui rendront facile l'acquisition et la diffusion de la connaissance, de créer des interfaces homme-machines avancés (langue naturelle, graphisme, etc.). Sans aucun doute le traitement, la création et la diffusion de la connaissance va jouer un rôle de première importance en cette fin du 20^{e} siècle.

L'accès aux données internes et externes à l'organisation constitue un enjeu économique très important pour les entreprises. Des investissements importants sont consentis pour la réalisation de systèmes d'informations intégrés, mettant en oeuvre la technologie des bases de données. Maintenant en quoi les systèmes à base de connaissance diffèrent-ils des systèmes d'information traditionnels? Les systèmes

traditionnels sont d'énormes mémoires corporatives enregistrant les faits pertinents pour l'organisation et accessible aux experts humains spécialistes, ingénieurs, gestionnaires, employés) qui manipulent enregistrent et créent des connaissances à l'extérieur des systèmes informatiques. Par contre nous définissons un système à base de connaissances comme un système permettant non seulement de gérer le contenu d'une base de connaissances (appelée base des faits), mais aussi de manipuler, d'enregistrer et d'utiliser des structures de connaissances. On réserve l'appellation de "système expert" pour la catégorie des systèmes à base de connaissances qui exploitent des connaissances relatives "rare" détenues par quelques experts humains très sollicités.

Les structures de connaissances sont essentiellement des représentations symboliques qui permettent de décrire les structures de données du système expert, les structures de comportement et les structures de connaissances elles-mêmes. Dans le cas le plus simple, ces structures de connaissances sont décrites sous la forme de règle de production c'est-à-dire des règles de la forme Si ..., ALORS ... Dans des systèmes plus sophistiqués les connaissances pourront être structurées et manipulées sous des formes plus complexes, telles que les cadres conceptuels ("frames") qui permettent d'invoquer globalement des structures de données et de comportement.

Un système expert est constitué principalement d'une base de connaissances rassemblant les structures qui représentent l'expertise relative au domaine traité, d'une base de faits et d'un moteur d'inférence.

Le moteur d'inférence est un programme qui construit les raisonnements du système à partir de la base de connaissances. Face à un problème donné, le moteur sélectionne les structures de connaissances pertinentes, bâtit un plan de résolution et l'applique au problème.

A ce programme essentiel s'ajoutent des modules d'interfaces qui facilitent la communication entre les personnes et la machine.

Pour en savoir davantage sur les principes concernant les systèmes experts le lecteur pourra consulter le chapitre 5 de la présente étude.

Suivant la nature du projet, le concepteur pourra réaliser un système à base de connaissances sur du matériel conventionnel ou utilisera des machines spécialisées.

En ce qui concerne les outils de développement, il pourra soit programmer entièrement le système à base de connaissances dans un langage de programmation symbolique tel que LISP ou PROLOG, soit utiliser un environnement de conception de systèmes experts. Un tel environnement fournit généralement un langage de représentation et de manipulation des connaissances de haut niveau (règles de production, frame, objects), des mécanismes d'inférences ou d'évaluation prédéfinis, ainsi que des facilités d'édition.

Le langage de programmation PROLOG a été choisi par les japonais pour leur projet de cinquième génération. PROLOG est plus qu'un langage de programmation, il est doté aussi d'un moteur d'inférence. Celui-ci permet de dériver de nouveaux faits à partir des anciens basés sur le principe de la démonstration de théorèmes. Cependant certains experts estiment que le choix de PROLOG a été loin d'être le meilleur et que ce choix a été prématuré: c'est-à-dire que les chercheurs japonais sont contraints d'utiliser une logique formelle alors que l'intelligence artificielle doit manipuler des connaissances incertaines et ambigues. De plus, les algorithmes basés sur la démonstration de théorème doivent rechercher à travers la base de connaissances dans une explosion de possibilités. Cependant PROLOG à donner lieu à des progrès tangibles concernant l'architecture des systèmes. (Pour en savoir davantage sur PROLOG, le lecteur pourra consulter l'annexe 4 de la présente étude.)

L'ambiguité et l'incertitude des connaissances semblent être des caractéristiques communes dans la plupart des domaines d'applications

en intelligence artificielle. Les problèmes reliés à l'environnement n'échapperont pas à cette caractéristique.

En ce qui concerne les environnements de conception des systèmes experts, les progrès permettent maintenant non seulement de développer plus rapidement et plus efficacement des systèmes à base de connaissances, mais aussi de rendre cette technique disponible sur des machines moins coûteuses.

En général, les systèmes experts complexes (de plusieurs milliers de règles) sont développés sur des ordinateurs spécialement conçus pour le traitement symbolique: ces machines ont l'avantage d'avoir une grande capacité de calcul, une forte capacité de mémoire, et un environnement de travail perfectionné. De plus ces machines permettent de disposer d'environnement de conception de systèmes experts "shells" très élaborés tels que ART, KEE, KES. Ces environnements permettent d'utiliser différents modes de représentation et de manipulation des connaissances (systèmes à base de règles, manipulation de frames ou d'objets). Généralement ces environnements sont coûteux (de l'ordre de cent deux cent mille dollars pour une machine LISP et un environnement de conception).

On peut noter aussi que des compagnies vendent actuellement pour un coût correspondant à un tiers ou la moitié de celui d'une machine LISP des postes de travail avancés qui offrent des possibilités équivalentes à celle d'une machine LISP achetée deux ou trois ans plus tôt.

On peut développer aussi de petits systèmes à base de connaissances (de quelques centaines de règles) sur des micro-ordinateurs professionnels. On peut alors disposer d'une puissance équivalente à celle d'une machine LISP achetée cinq ou six ans plus tôt et d'environnements de conception en général adaptés à la manipulation de connaissances sous la forme de règle de production. On peut disposer d'un tel psote de travail pour dix à quinze mille dollars.

Il est possible aussi d'envisager la possibilité que l'entreprise décide de créer son propre environnement de conception de systèmes experts. Le coût de la conception de cet outil sera sans doute plus élevé que l'achat d'un système existant, mais celui-ci aura l'avantage d'être bien adapté au problème à traiter dans l'entreprise. De plus les chercheurs ayant participé à la conception seraient en tout temps disponible dans l'éventualité de nouveaux développements.

Le choix du type de matériel dépend de l'envergure du projet. Les systèmes sur micro-ordinateurs seront avantageusement utilisées pour le développement de maquette ou de petits systèmes experts, tandis que les systèmes experts à grande envergure seront construits sur des machines spécialisés et des environnements de conception sophistiqués.

Pour la conception de systèmes experts complexes, dans l'éventualité ou les machines et les environnements spécialisés ne sont pas disponibles, il est recommandé de faire une étude préalable en développant une maquette sur micro-ordinateur. Cette étude permettra de clarifier le problème, de valider la démarche, et de justifier le choix de matériels et de logiciels coûteux pour le développement du système expert.

Pour en savoir davantage sur les outils de conception de système expert le lecteur pourra consulter l'annexe 1 de la présente étude.

8.2 DÉMARCHE DE CONSTRUCTION D'UN SYSTÈME EXPERT

Le développement d'un système à base de connaissance fait intervenir plusieurs spécialistes ou usagers. L'ingénieur du savoir connaît les modèles, les méthodes et les outils utilisés en ingénierie du savoir. Il met à profit son expérience d'analyser pour interagir avec l'expert du domaine. L'ingénieur doit découvrir un expert du domaine (ou plusieurs experts si nécessaire pour l'élaboration d'une

base de connaissances qui reflète l'expertise plus largement acceptée). Il doit comprendre les raisonnements et les comportements de l'expert pour pouvoir les formaliser et les implanter en utilisant l'environnement de conception approprié. La base de connaissances est élaborée petit à petit par l'ingénieur du savoir et testée par l'expert du domaine sur différents cas.

Il ne faut pas négliger non plus l'importance des futures usagers qui correspondent à la troisième catégorie des intervenants. On devra tenir compte des comportements des usagers pour concevoir des interfaces efficaces et adaptées à l'usage prévisible du système expert. Ainsi le système sera utilisé s'il est plus facile pour des usagers d'obtenir une connaissance experte, plutôt que d'appeler par téléphone le spécialiste pour obtenir une réponse. Il est donc indispensable d'impliquer les usagers très tôt dans le projet afin de s'assurer que les décisions prises au cours de la conception permettront de livrer un produit satisfaisant leurs exigences d'utilisation.

On notera que dans un projet pour le développement d'un système expert, les experts et les usagers sont des intervenants distincts, alors que dans un projet de conception de systèmes traditionnels, le système est développé avec les usagers pour les usagers eux-mêmes.

Comme on a pu le constater, la nature des problèmes traités conduit à développer un système à base de connnaissances suivant une approche par prototypage. Les comportements liés à la manipulation de connaissances sont en général mal connus et très peu formalisés: l'expert humain agit le plus souvent par intuition et n'a que très rarement conscience des mécanismes de raisonnement de déduction qu'il met en oeuvre. La conception d'un système expert doit donc être fait de façon intensive et conduit à la mise au point d'un prototype par raffinements sucessifs. On peut ainsi préciser petit à petit les connaissances mises en oeuvre par les spécialistes, les formaliser et les emmaganiser dans le prototype du système expert.

On peut distinguer cinq phases principales au cours du développement du système expert.

La première phase consiste en une étude d'orientation. Celle-ci permet de clarifier le problème, étudier la faisabilité du projet en fonction de la nature du problème et des ressources nécessaires et proposer un plan de développement. Au cours de cette phase on doit identifier les principales caractéristiques des tâches qui devront être simulées par le système expert: problème à résoudre, comportement de l'expert et des usagers, nature des connaissances. Il est nécessaire d'impliquer dès le début du projet tous les intervenants et leur faire comprendre les conséquences de cette implication.

La deuxième phase consiste à l'élaboration d'une première maquette du système. Celle-ci permettra de valider l'approche de conception proposée et éventuellement de justifier les investissements à consentir pour l'acquisition du matériel et d'un environnement de conception adaptés au problème. Au cours de cette phase l'ingénieur du savoir interagit avec l'expert afin de déterminer les modèles appropriés pour la représentation des faits, pour la structuration des connaissances et pour l'identification des stratégies de résolutions utilisées pour résoudre les problèmes.

Il est important de réaliser la maquette dans des délais assez courts afin de démontrer la faisabilité du système envisagé et de permettre de décider de l'opportunité d'investir dans le projet. Le développement de la maquette permet aussi de conserver l'intérêt des différents intervenants tout en leur permettant d'évaluer objectivement l'ampleur du projet.

La troisième phase consiste en l'élaboration du système expert comme tel par une approche de prototypage. En fonction de l'expérience acquise au cours de la phase de modélisation conceptuelle, l'ingénieur du savoir construit le noyau de la base de connaissances avec le matériel et l'environnement de conception retenus pour supporter

le système expert. Par raffinements successifs, la base de connaissances est enrichie avec la collaboration de l'expert, et l'interface-usager est ajusté aux exigences d'utilisation du système expert.

La quatrième phase consiste en l'évaluation du système. L'évaluation du système permet de déterminer si le système rencontre les critères de performance retenus au cours de la phase de modélisation conceptuelle. C'est aussi le moment où le système est essayé par d'autres experts du domaine.

La cinquième et dernière phase consiste en l'intégration du système dans le milieu de travail des usagers en tenant compte de leur méthode de travail. On peut encore avoir à ajuster l'interface et à développer quelques outils complémentaires.

8.3 APPLICATION DE LA TECHNOLOGIE DES SYSTÈMES EXPERTS À LA DIRECTION ENVIRONNEMENT D'HYDRO-QUÉBEC

Nous avons développé en annexe 4, une maquette de système expert sur le problème des déversements accidentels. Cette maquette est écrite en langage turbo-prolog, une nouvelle version de prolog mise sur le marché à l'été 1986. Cet exemple montre qu'il est possible d'utiliser la technologie des systèmes experts aux problèmes environnementaux.

Ce petit système expert donne les stratégies possibles de confinement et de récupération des hydrocarbures en fonction de la situation qui prévaut sur le site du déversement. Cette maquette fut développée à partir du document intitulé "Données de base pour le développement d'un système expert concernant les techniques d'intervention lors d'un déversement accidentel de contaminants". Ce document fut préparé par André Bériault et Jean-Claude Tessier de la direction Environnement d'Hydro-Québec.

Comme le lecteur pourra le constater en consultant l'annexe 1 de la présente étude, il n'y eu jusqu'à maintenant que peu de chercheurs qui ont consacré des efforts pour le développement de systèmes experts appliqués à des problèmes environnementaux. Les domaines d'application furent surtout en médecine, en agriculture, en recherche pétrolière, en contrôle de processus, etc. Il ne fait pas de doute à notre avis que l'on pourrait tirer avantage à appliquer cette technologie à des problèmes environnementaux.

S'il y a un secteur d'activité où les domaines d'expertises se chevauchent c'est bien celui de l'environnement. Dans beaucoup de cas lorsqu'un problème apparaît, tous ces domaines d'expertise sont nécessaires dans le but d'en arriver à sa solution.

Lors d'une problématique, les experts concernés sont contactés, après discussions et concensus, une série de mesures correctives est proposée pour la résoudre. C'est un peu la démarche d'un système expert; ou plutôt le système expert essaie d'imiter la démarche de l'expert lui-même à l'aide de ses granules de connaissances qui lui ont été fournies préalablement.

À l'aide de systèmes experts, les connaissances de chacun des domaines d'expertise peuvent être formalisées, vulgarisées et programmées sous forme de règle de production. Ces connaissances ainsi programmées sont directement accessibles à tous les niveaux de l'entreprise, et dans toutes les régions si l'entreprise est étendue sur un vaste territoire comme Hydro-Québec.

Pour Hydro-Québec, la construction de systèmes experts pourrait à moyen terme permettre l'exportation d'une forme d'expertise au-delà de son siège social.

Dans la majorité des cas on pourrait consulter le système expert au lieu de l'expert lui-même. On pourra donc optimiser l'utilisation de l'expertise et de l'expert lui-même. Ce dernier aura plus de temps à

consacrer à son perfectionnement, et moins à résoudre des problèmes. Son perfectionnement pourra par ricochet, améliorer la connaissance experte du système lui-même.

En résumé, voici les avantages que l'on pourra en retirer:

1) meilleure communication entre experts;

2) meilleure communication entre expert et usager;

3) on pourra obtenir un transfert de la connaissance, par le système expert, de l'expert vers l'usager;

4) on pourra améliorer l'utilisation des experts: permettre aux experts d'être utilisés à d'autres fonctions sans nuire à l'expertise qu'ils doivent donner quotidiennement;

5) permet à l'expert de réviser et de compléter sa logique dans son domaine d'expertise.

De plus en plus ces technologies de diffusion de la connaissance sont utilisées dans de nombreux domaines. Il serait avantageux qu'elles le soient à la direction Environnement d'Hydro-Québec.

Comme nous avons pu le constater en faisant l'inventaire des systèmes experts existants, peu ont un lien direct à la résolution de problèmes environnementaux. Nous avons remarqué, cependant, que beaucoup d'efforts ont été déployés aux États-Unis pour le développement de systèmes experts qui servent à la surveillance et au contrôle des centrales électriques qui puisent leur énergie du nucléaire. Principalement, les recherches se sont concentrées après l'accident survenue à la centrale de Three Miles Island.

Notre point de vue est qu'il ne faudrait pas attendre que des systèmes experts soient développées par d'autres pour ensuite les

utiliser. Ces systèmes importés de l'étranger seront en général des outils mal adaptés à nos besoins. Dans l'éventualité où on décidera de les utiliser, une bonne partie de l'énergie intellectuelle de nos experts servirait à adapter ces systèmes pour nos propres besoins, et sans jamais atteindre une complète adaptation.

Nous croyons qu'il ne faut plus être de bons utilisateurs de technologies qui nous viennent de l'étranger. Il nous faut concentrer nos efforts intellectuels à la conception d'outils qui sont parfaitement adaptés à nos besoins. L'enjeu économique, pour ce qui est de la technologie des systèmes experts, est trop important. Il ne faut pas oublier que nous avons au Québec une masse très importante d'experts qui ne demandent pas mieux que d'être utilisés, et de donner tout leur potentiel de connaissance.

Ce processus de conceptualisation serait avantageux à tous les secteurs de l'utilisation de la ressource humaine à Hydro-Québec. Comme nous l'avons déjà souligné, tous les niveaux d'expertise doivent participer à l'élaboration de systèmes experts: gestionnaires, experts, informaticiens, ingénieurs de la connaissance et usagers. Ces systèmes pourront, par conséquent, très bien s'intégrer à nos méthodes de travail tout en rendant accessible la connaissance à tous les niveaux décisionnels de l'entreprise.

8.4 RECOMMANDATIONS

Au cours des prochaines années, les systèmes experts seront de plus en plus utilisés dans les entreprises, leur permettant de gérer les connaissances comme une véritable ressource. L'enjeu de cette évolution est très grand et son succès dépendra beaucoup des résultats des premiers essais d'implantation de ces nouvelles technologies. "Aussi les projets en ingénierie du savoir devront être abordés avec enthousiasme, mais aussi avec toute la prudence qui est recommandée

lorsqu'on introduit des méthodes et des outils qui risquent de marquer en profondeur la culture organisationnelle."[1]

L'intégration des systèmes experts à la direction Environnement demandera un effort important de la part des gestionnaires et des experts. Selon les besoins de la Direction, cette dernière doit se préparer à compléter un programme de travail qui comprendra trois phases de réalisation. La première phase consiste à se familiariser avec les techniques des systèmes experts en réalisant quelques projets (déversements accidentels, gestion des réservoirs à buts multiples, diagnostic environnemental, etc.) à partir d'un système général existant. L'annexe 1 présente une analyse critique de quelques systèmes généraux.

La deuxième phase consiste à définir une liste de besoins en système expert et à établir une liste de critères auxquels le système général devra satisfaire. Un critère important est sans aucun doute la notion d'évaluation de la qualité du diagnostic par analyse probabiliste, suite au caractère hypothétique et qualitatif des études de la direction Environnement. Le développement d'un système général sera alors entrepris par la direction Environnement en collaboration avec des organismes de recherche.

La dernière phase consiste à développer un système général et des systèmes experts pouvant interagir en temps réel. L'interaction en temps réel est indispensable dans la gestion courante des réservoirs. Pour permettre d'atteindre une capacité acceptable, on doit procéder à la conceptualisation et à la fabrication d'un équipement spécialisé permettant le transfert du software en hardware. Le développement d'un équipement spécialisé se fera en collaboration avec des organismes de recherche.

MOULIN, B. "Les systèmes à base de connaissances dans les organisations", la revue L'ingénieur, septembre-octobre 1986, pages 21 à 27.

Le Centre de recherche informatique de Montréal (CRIM) est très intéressé à prendre en charge le développement d'un système général et d'un équipement spécialisé en collaboration avec la direction Environnement d'Hydro-Québec. Un programme de travail sera établi avec le CRIM au début de l'année 1987 et sera proposé à la direction Environnement pour le début mars 1987.

RÉFÉRENCES

A.-U., Volume contenant ces documents.

Revue ARTIFICIAL INTELLIGENCE, volume 2, 1971.

Revue ARTIFICIAL INTELLIGENCE, volume 3, 1972.

Revue ARTIFICIAL INTELLIGENCE, volume 4, 1973.

Revue ARTIFICIAL INTELLIGENCE, volume 5, 1974.

Revue ARTIFICIAL INTELLIGENCE, volume 6, 1975.

Revue ARTIFICIAL INTELLIGENCE, volume 8, 1977.

Revue ARTIFICIAL INTELLIGENCE, volumes 10-11, 1978.

Revue ARTIFICIAL INTELLIGENCE, "Proceedings of the International Joint Conference on Artificial Intelligence, 1981.

Revue ARTIFICIAL INTELLIGENCE, volumes 20-21, 1983.

Revue ARTIFICIAL INTELLIGENCE, volume 22, 1984.

Revue ARTIFICIAL INTELLIGENCE, volumes 23-24, 1984.

Revue ARTIFICIAL INTELLIGENCE, volume 25, 1985.

Revue ARTIFICIAL INTELLIGENCE, volume 26, 1985.

AURON, B. et E.A. FEIGENBAUM, "The Handbook of Artificial Intelligence", Volumes I, II, III; Heuristech Press, Stanford, California; William Kaufman Inc., Los Altos, California; United States, 1981, Volume I: 409 p., Volume II: 640 p., Volume III; 428 p.

BONNET, A., L'intelligence artificielle (promesses et réalités), Interéditions, Paris, 1984, 271 p.

BRIOT, M. et DE SAINT-VINCENT, A.R., "La vision des robots", revue La Recherche, Mensuel n° 170, octobre 1985, pp. 1264-1275.

BUCHANAN, B.G., Expert Systems: Working Systems and the Research Literature, Knowledge Systems Laboratory, Stanford University, December 1985, 55 p.

CAPPUCIO, A., "Multilog 2: un outil professionnel", revue Micro-systèmes, n° 55, juillet-août 1985, pp. 126-129.

CAPPUCIO, A., "XPer: Gestion de bases de connaissances", revue Micro-systèmes, n° 53, mai 1985, pp. 158-160.

CHATILLA, R., LAUMOND, J.-P. et R. PRAJOUX, "Les robots mobiles autonomes", revue La Recherche, Mensuel n° 170, octobre 1985, pp. 1276-1289.

COHEN, P.R. et FEIGENBAUM, E.A., The Handbook of Artificial Intelligence (volume 1), Heuristech Press, Stanford, 1981, 409 p.

COHEN, P.R. et FEIGENBAUM, E.A., The Handbook of Artificial Intelligence (volume 2), Heuristech Press, Stanford, 1982, 428 p.

COHEN, P.R. et FEIGENBAUM, E.A., The Handbook of Artificial Intelligence (volume 3), Heuristech Press, Stanford, 1982, 640 p.

COLMERAUER, A., "Prolog langage de l'intelligence artificielle", revue La Recherche, n° 158, septembre 1984, pp.

COLMERAUER, A., KANOUI, H. et M. VAN CANEGHEM, "Prolog, bases théoriques et développements actuels", Technologie et Science Informatiques, Vol. 2, nº 4, 1983, Faculté des sciences de Marseille-Vurniny, 198?.

D.R., "La théorie des graphes", revue Micro-systèmes, nº 38, janvier 1984, pp. 138-142.

DeMORI, R., "Introduction à l'Intelligence Artificielle", conférence, Journée Augustin Frigon: Intelligence Artificielle Systèmes-Experts, 9 novembre 1984, pp. 1-0, 1-35.

ÉCOLE POLYTECHNIQUE DE MONTRÉAL, Journée Augustin Frigon (Intelligence Artificielle et Systèmes Experts), École Polytechnique de Montréal, 1984.

ÉCOLE POLYTECHNIQUE DE MONTRÉAL, Proceeding of the International Joint Conference on Artificial Intelligence, École Polytechnique de Montréal, 1977.

FAGOT-LARGEAULT, A., "La simulation du raisonnement médical", revue La Recherche, mensuel nº 170, octobre 1985, pp 1176-1187.

FALLER, B., "Intelligence artificielle et jeux de stratégie", revue Micro-systèmes, nº 54, juin 1984, pp. 148-152.

FALLER, B., "L'ordinateur et les jeux de l'esprit", revue La Recherche, mensuel nº 170, octobre 1985, pp. 1064-1075.
Remarque: les différentes techniques de l'Intelligence Artificielle.

FARRENY, H., Les systèmes experts (principes et exemples), Collection Techniques Avancées de l'Informatique, Cepadues Éditions, Toulouse, France, -1985, 254 p.

FEIGENBAUM, E.A. et P. McCORDUCK, La 5e génération: le pari de l'intelligence artificielle à l'aube de 21e siècle, Intereditions, Paris, 1984, p.

FERBER, J., "Le filtrage: une technique de base de l'intelligence artificielle", revue Micro-systèmes, n° 40, mars 1984, pp. 108-113.

FERBER, J., "Les systèmes-experts (I): raisonner dans un univers réel", revue Micro-systèmes, n° 41, avril 1984, pp. 118-122.

FERBER, J., "Les systèmes-experts (II): un moteur d'inférences PASCAL", revue Micro-systèmes, n° 42, mai 1984, pp. 132-137.

FERBER, J., "Les systèmes-experts (III): des noyaux qui ont la pêche", revue Micro-systèmes, n° 43, juin 1984, pp. 68-73.

FERBER, J., "Les systèmes-experts (IV): du mythe à la réalité", revue Micro-systèmes, n° 44, juillet-août 1984, pp. 112-114.

FERBER, J., "La compréhension automatique du texte", revue Micro-systèmes, n° 45, septembre 1984, pp. 238-244.

FERBER, J., "La compréhension automatique du texte", revue Micro-systèmes, n° 46, octobre 1984, pp. 188-191.

FERBER, J., "La compréhension automatique du texte", revue Micro-systèmes, n° 47, novembre 1984, pp. 170-174.

FERBER, J., "LISP: le langage de l'intelligence artificielle", revue Micro-systèmes, n° 48, décembre 1984, pp. 148-154.

FERBER, J., "LISP: le langage de l'intelligence artificielle", revue Micro-systèmes, n° 49, janvier 1985, pp. 126-131.

FERBER, J., "LISP: le langage de l'intelligence artificielle", revue Micro-systèmes, n° 50, février 1985, pp. 174-178.

FERBER, J., "LISP: le langage de l'intelligence artificielle", revue Micro-systèmes, n° 51, mars 1985, pp. 180-187.

FERBER, J., "Les langages objets: une affaire de messages", revue Micro-systèmes, n° 52, avril 1985, pp. 152-158.

FERBER, J., "Systèmes-experts: un dynamisme croissant", revue Micro-systèmes, n° 56, septembre 1985, pp. 206-211.

FERBER, J., "La programmation par acteurs (II): langages et méthode", revue Micro-systèmes, n° 63, avril 1986, pp. 164-171.

FERBER, J., "Systèmes-experts: du moteur à la connaissance", revue Micro-systèmes, n° 64, mai 1986, pp. 164-168.

FIESCHI, M., Intelligence artificielle en médecine (des systèmes experts), Éditions Masson, Paris, 1984, 207 p.

FOX, M.S., Industrial Applications of Expert Systems, Centre de cours intensifs de l'École Polytechnique de Montréal, printemps 1986, 45 p.

GALLAIRE, H., "La représentation des connaissances", revue La Recherche, mensuel n° 170, octobre 1985, pp. 1240-1251.

GANASCIA, J.-G., "La conception des systèmes experts", revue La Recherche, mensuel n° 170, octobre 1985, pp. 1042-1051.
Remarque: historique et conception des systèmes experts.

GANASCIA, J.-G., "L'apprentissage dans les systèmes experts", revue Micro-systèmes, n° 58, novembre 1985, pp. 146-151.

GIANNESINI, F., KANOUI, H. et V.C.M. PASEROR, "Prolog", Intereditions, Paris, France, 1985, 318 p.

GILMORE, J., "Les armes intelligentes", revue La Recherche, mensuel n° 170, octobre 1985, pp. 1052-1063.

HARDY, J.-L., "L'eden de SMALLTALK", revue Micro-systèmes, nº 39, février 1984, pp. 116-121.

HATON, J.-P. et A. BELAÏD, "La reconnaissance de l'écriture", revue La Recherche, mensuel nº 170, octobre 1985, pp. 1188-1197.

JACKSON, P.C. Jr., "Introduction to Artificial Intelligence", published by Mason and Lipscomb, London, England, printed in United States by Petrocelly Books, New York, 1974, 453 p.

KASTNER, J.R. et S.J. Hong, "A Review of Expert Systems", European Journal of Operational Research, 18, 1984, North-Holland, pp. 285-292.

KAYSER, D., "Des machines qui comprennent notre langue", revue La Recherche, mensuel nº 170, octobre 1985, pp. 1198-1213.

KNOWLEDGE SYSTEMS LABORATORY, Knowledge Systems Laboratory 1985, Stanford University, 1985, 94 p.

KODRALOFF, Y., "Quand l'ordinateur apprend", revue La Recherche, mensuel nº 170, octobre 1985, pp. 1252-1263.

LA RECHERCHE, mensuel nº 170, octobre 1985, Paris.

LARVET, P., "Un moteur d'inférence d'ordre zéro: en BASIC", revue Micro-systèmes, nº 50, février 1985, pp. 191-197.

LEGEARD, B., "Le langage PROLOG", revue Micro-systèmes, nº 44, juillet-août 1984, pp. 100-111.

MARIANI, J., "La reconnaissance de la parole", revue La Recherche, mensuel nº 170, octobre 1985, pp. 1214-1227.

MYLOPOULOS, J. et H. LEVESQUE, "An Overview of Knowledge Representation", conférence, Journée Augustin Frigon: Intelligence Artificielle et Systèmes-experts, 9 novembre 1984, pp. 3-0, 3-14.

MYLOPOULOS, J. TSOTSOS, J. et T. SHIBAHARA, "Research of Knowledge-based systems: PSN, ALVEN and CAA", conférence, Journée Augustin Frigon: Intelligence Artificielle et Systèmes-experts, 9 novembre 1984, pp. 3-15, 3-17.

NEGOITA, C. V., "Expert Systems and Fuzzy Systems", The Benjamin/Cummings Publishing Company, Menlo Park, California, United States, 1985, 190 p.

NILSSON, N. J., "Principles of Artificial Intelligence", Tioga Publishing Company, Palo-Alto, California, United States, 1980, 476 p.

PIC, A., "Jeux et intelligence artificielle", revue Micro-systèmes, nº 57, octobre 1985, pp. 148-156.

PITRAT, J., "La naissance de l'intelligence artificielle", revue La Recherche, mensuel nº 170, octobre 1985, pp. 1130-1041.
Remarque: une revue générale.

ROUSSEAU, M., "Futursys: la cinquième génération en marche", revue Micro-systèmes, nº 59, décembre 1985, pp. 148-153.

SANSONET, J.-P., "Les machines de l'intelligence artificielle", revue La Recherche, mensuel nº 170, octobre 1985, pp. 1228-1239.

SATHI, A., "AI Based Systems in Project Management Conference", Journée Augustin Frigon: Intelligence Artificielle et Systèmes-experts, 9 novembre 1984, pp. 2-0, 2-1.

SOCIÉTÉ CANADIENNE POUR L'ÉTUDE DE L'INTELLIGENCE PAR ORDINATEUR, Actes de la sixième conférence canadienne sur l'intelligence artificielle, École Polytechnique de Montréal, 1986, 268 p.

UNIVERSITY OF CALIFORNIA AT LOS ANGELES, Proceedings of the Ninth International Joint Conference on Artificial Intelligence (Volume 1), Los Angeles, 1985.

UNIVERSITY OF CALIFORNIA AT LOS ANGELES, Proceedings of the Ninth International Joint Conference on Artificial Intelligence (Volume 2), Los Angeles, 1985.

VEMIAN, R., "Mon système de reconnaissance vocale", revue Micro-systèmes, n° 49, janvier 1985, pp. 107-113.

WARNIER, J.D., "L'homme face à l'intelligence artificielle", Les Éditions d'Organisations, Paris, 1984, 140 p.

YAZDANI, M. et A. NARAYANAN, Artificial Intelligence Human Effects, Ellis Horwood Series, Université de Exeter, Angleterre, 1984, 318 p.

ANNEXE 1

INVENTAIRE DES SYSTÈMES EXPERTS ET DES SYSTÈMES GÉNÉRAUX

DISCIPLINES INVENTORIÉES

- Accidents et déversements
- Acquisition de connaissances
- Administration
- Agriculture
- Agronomie
- Archéologie
- Biologie
- Cartographie
- Chimie
- Conception et fabrication assistée par ordinateur (CFAO)
- Contrôle de processus
- Droit
- Enseignement assisté par ordinateur (EAO)
- Générateur de systèmes-experts
- Génie
- Géologie
- Informatique
- Jeux
- Langue naturelle
- Mathématiques
- Médecine

DISCIPLINES INVENTORIÉES (suite)

- Militaire
- Multi-experts
- Mycologie
- Physique
- Production manufacturière
- Programmation automatique
- Raisonnement
- Reconnaissance de la parole
- Réparation
- Risques
- Robotique
- Systèmes-experts
- Systèmes généraux
- Systèmes d'opération
- Traduction automatique
- Zoologie

ACCIDENTS ET DÉVERSEMENTS

Deecription des systèmes experts

REACTOR

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
REACTOR	-	-	-	-	Diagnostic et traitements d'accidents dans des réacteurs nucléaires.	-	-	-	-	-	-	KES	

ACQUISITION DE CONNAISSANCES

Description des systèmes experts

CHI

KLAUS

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
CHI	-	(7)	-	- (Kestrel Institute)	Acquisition des connaissances, vérification de la consistance des entrées, "débugging", édition, maintenance.	-	-	-	-	-	-	-	
KLAUS	N. Haas, G.G. Hendrix.	(7)	1980	États-Unis	Acquisition et représentation des connaissances.	-	-	-	-	-	-	-	

ADMINISTRATION

Description des systèmes experts

ACE

APEX system

BLAH

EDDAS

EDF

FINEX

KH-1

MANAGER

PEGASE

PLANET

PRAS

RABBIT

SERAFIN

SMC

TAXADVISOR

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
ACE	-	(1)	1983	États-Unis (Bell Labo-ratories)	Destiné à établir des rapports synthétiques sur l'état de câbles téléphoniques à par-tir des enregistre-ments d'incidents.	100	-	-	-	-	-	-	
APEX system	R. Davis	(31)	-	États-Unis (MIT)	Aid professional financial planners, manage clients' accounts	-	-	-	-	-	-	-	
BLAH	J.L. Weiner	(A)	1980	États-Unis (Univ. du New Hamp-shire)	Service d'impôt américain.	-	Règles de pro-ductions + dé-monstrations.	AMORD	Fonc-tionnel	Chaînage arrière	-	AMORD	
EDDAS	J.L. Feinstein, F. Selma.	(31)	1985	États-Unis	Conseils sur la révé-lation d'informations administratives confidentielles.	-	-	-	-	-	-	-	
EDF	-	(1)	-	-	Organisation de plans de relevé de compteurs.	Environ 50	-	-	-	-	-	SNARK	
FINEX	-	(13)	-	États-Unis (Univ. Caroline du Sud)	Analyse financière	-	-	PROLOG	-	-	-	-	1) Capable d'expliquer son raisonnement.

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
KM-1	-	(2)	-	- (System Development Corpora-tion)	Système de gestion de la connaissance qui tente d'intégrer les possibilités du système de gestion de données et du système de base de connais-sances.	-	-	-	Re-cherche	-	-	-	
MANAGER	C. Ernst	(1)	1985	France (C.E.R.F. I.A. + Univ. P. Sabatier)	Application à la gestion en temps réel des personnels para-médicaux d'un grand hôpital.	-	-	METALOG	-	-	-	METALOG	
PEGASE	-	(1)	1984	France (Univ. et École Poly-tech. de Lausanne	Destiné à assister des analystes finan-ciers devant juger des demandes de crédits.	-	-	-	-	-	-	-	
PLANET	V. Dhar	(11)	1984	États-Unis (Univ. de New York)	IJCAI-85, vol. 1, p. 188.	-	-	-	-	-	-	-	1) Capable de peser le pour et le contre des choix offerts.
PRAS	-	(2)	-	Japon (Hitachi)	Évaluation des risques de grands projets de cons-truction.	-	-	-	Dévelop-pement	-	-	-	
RABBIT	-	(2)	-	États-Unis (Xerox)	Aide l'utilisateur à interroger une base de données.	-	-	-	-	-	-	-	

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
SERAFIN	R. Pasero, R. David.	-	1985	France (Société Prologia	Domaine adminis-tratif.	-	-	-	-	-	-	-	
SMC	R. Pasero, R. David.	-	1986	France (Société Prologia)	Domaine bancaire.	-	-	-	-	-	-	-	
TAXADVISOR	-	(13)	-	États-Unis (Univ. d'Illinois)	Analyse financière.	-	-	-	-	-	-	-	
-	-	(13)	1985	Grande-Bretagne	Analyse financière.	210	-	-	Recher-che	-	-	-	
-	P. Hart	(31)	-	États-Unis	Estime une variété de risques commer-ciaux dans le domaine de l'assurance	-	-	-	-	-	-	-	
-	P. Hart	(31)	-	États-Unis (American Intern-ational Group)	Advise and support commercial insurance underwriters.	-	-	-	-	-	-	-	
-	C. Lévesque	(1), 3	1983	France (G.R.22, Univ. Paris VI)	Destiné à la gestion de personnel et à la confection de paye.	400	-	BASIC	-	-	-	-	
-	O. Pallea et al.	(13)	-	France (Iecsi)	Conseiller en placement.	350	Règle de production	-	Recher-che	-	-	-	1) Moteur d'ordre 0.

AGRICULTURE

Description des systèmes experts

POMME

WHEAT COUNSELOR

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
POMME	J.W. Roach, R.S. Virkar, M.J. Weaver, C.R. Drake.	(31)	1985	États-Unis (Virginia Polytechnic Institute)	Conseille les fermiers sur la gestion des vergers de pommes (emploi de pesticides, etc.)	-	-	-	-	-	-	-	
WHEAT COUNSELOR	-	(31)	1985	États-Unis (ICI)	Conseils sur le contrôle des maladies lors de la récolte du blé.	-	-	-	Fonc-tionnel	-	-	-	

AGRONOMIE

Description des systèmes experts

AQ-11

PLANT/dc

PLANT/da

TOM

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
AQ-11	R.S. Michalski	(5)	1978	États-Unis (Univ. d'Illinois)	Inférence de règles de diagnostic concernant les maladies du soja.	-	Règles de production	-	-	-	Sur 430 ex. choisis: a découvert le meilleur diagnostic dans 97,6% des cas et a émis une liste des hypothèses qui contenait le meilleur diagnostic dans 100% des cas.	-	1) Fonctionne bien car il y a correspondance directe entre symptômes et diagnostics: "fonctionnerait beaucoup moins bien dans le cas où il faudrait utiliser des conclusions intermédiaires et des stratégies conditionnelles à ces conclusions". 2) Les formes syntaxiques générées sont limitées. 3) Certaines règles sont logiquement lourdes et quelquefois simplifiables (si le système possédait "le bon sens"). 4) Plutôt algorithme traditionnel... 5) Apprentissage à l'aide d'exemples. 6) Moteur de catégorie 0.
PLANT/dc	R.S. Michalski	(1)	-	États-Unis (Univ. d'Illinois)	Prévoir les dégâts provoqués par un parasite sur le maïs.	-	-	-	-	-	-	ADVISE	1) Dérivé de PLANT/ds.
PLANT/ds	R.S. Michalski	(1)	1982	États-Unis (Univ. d'Illinois)	Diagnostic des maladies du soja.	-	-	-	-	-	-	ADVISE	1) L'ordre dans lequel les règles sont déclenchées influence le résultat. 2) Capable de généraliser des règles. 3) Système adaptatif. 4) Il utilise, outre les règles fournies par les experts, des règles automatiquement dérivées par généralisation de cas.
TOM	-	(1), 9	1984	France (Cognitech, Paris)	Prototype développé afin d'évaluer l'opportunité et la faisabilité d'un service national de consultation pour les maladies des plantes.	Environ 200	-	-	-	-	-	EMYCIN	1) Appareil destiné aux professionnels de l'agriculture.

ARCHÉOLOGIE

Description des systèmes experts

SUPERIKON

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
SUPERIKON	M.S. Lagrange, M. Renaud.	(1)	1984	France	Archéologie.	Environ 100	-	-	-	-	-	SNARK	

BIOLOGIE

Description des systèmes experts

GENESIS

MOLGEN

PROTEAN

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
GENESIS	-	(2)	-	États-Unis (Intelli-Genetics Inc.) Palo Alto	Système qui aide les scientifiques à planifier et simuler des expériences d'épissage des gènes.	-	-	-	Fonc-tionnel	-	-	-	
MOLGEN	M. Stefik	1, 2, 5, H, 16	1977	États-Unis	Aide des généticiens pour la planification d'expériences sur l'ADN.	-	-	-	Fonc-tionnel	-	-	EMYCIN	1) Utilisé dans les universités et les laboratoires industriels de biologie moléculaire et de génétique. 2) Offre la possibilité aux utilisateurs de pouvoir structurer les connaissances (règles, faits) en conglomérats significatifs dans l'application.
PROTEAN	Groupe HELIX	(32)	1986	États-Unis (Univ. Stanford)	Déterminer la structure tridimensionnelle des protéines.	-	-	-	Recher-che	-	-	-	1) Pour atteindre le but visé les auteurs veulent utiliser l'information provenant de la résonnance magnétique nucléaire, la spectroscopie, etc. 2) Programme de recherche relativement récent.

CARTOGRAPHIE

Description des systèmes experts

HAWKEYE

ENSEIGNEMENT ASSISTÉ PAR ORDINATEUR (EAO)

Description des systèmes experts

DECGUIDE

EXCHECK

GENIUS

GUIDON

KBEMACS

LAURA

MENO-II

MYCROFT

NEOMYCIN

PHENARETE

PIXIE

PROGRAMMER'S APPRENTICE

PROUST

PSM-NMR

SCENT

SCHOLAR

SITS

SPADE

TVX

WHY

DROIT

Description des systèmes experts

LDS

TAXMAN

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
LDS	-	(2)	-	États-Unis (Rand Cor-poration)	Modèle les processus de prise de décisions des juristes et procède à des ajuste-ments de la légis-lation sur les produits.	-	-	-	Expéri-mental	-	-	-	
TAXMAN	-	(2)	-	États-Unis (Univ. de Rutgers)	Traite des règlements implicites dans les lois fiscales et suggère une suite d'arrangements con-tractuels que peut utiliser une entre-prise pour atteindre ses objectifs finan-ciers.	-	-	-	-	-	-	-	

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
DECGUIDE	W. Perkins	(31)	-	États-Unis (Lockheed)	Tutor designers in design checking.	-	-	-	-	-	-	-	
EXCHECK	P. Suppes et al.	(7)	-	États-Unis (Univ. Stanford)	Enseignement des mathématiques au niveau universitaire.	-	-	-	-	-	-	-	1) Comprend des esquisses de preuves, preuves sommaires, etc.
GENIUS	G.I. McCalla, K.M. Murtagh.	(3)	1985	Canada (Univ. Saskat-chewan)	Système pour aider des étudiants pro-grammeurs à élaborer des programmes ainsi qu'à résoudre les "bugs".	-	-	-	-	-	-	-	
GUIDON	W.J. Clancey	(1), 2, 3, 4, 10, 15	1979	États-Unis (Stanford)	Enseignement assisté par ordinateur.	Environ 50	Règles de production	-	Expéri-mental	-	-	EMYCIN	1) Suit et commente la démarche d'un étu-diant en médecine essayant d'établir un diagnostic médical. Pour ce faire, il se base sur la pertinence des questions posées par l'étudiant comparativement à celles posées par MYCIN.
KBEMACS	R.C. Waters	-	1985	États-Unis (MIT)	Assiste un pro-grammeur dans l'élaboration de programmes LISP et ADA.	30 (nombre de cli-chés)	Plans + clichés	LISP	Recher-che	-	-	-	1) La version originale a été KBE en 1981. 2) Effectue très peu de raisonnement, le contrôle de l'utilisateur est employé. 3) La communication ne se fait pas en langue naturelle.
LAURA	A. Adam, J.P. Laurent.	3, (C), D	1980	France (Univ. de Caen)	Système permettant aux étudiants pro-grammeurs en FORTRAN de résoudre leurs "bugs" (erreurs de sémentique).	-	-	-	-	-	-	-	1) Il procède par comparaison avec un pro-gramme type. 2) Apporte la correction nécessaire au bon fonctionnement du programme.

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
MENO-II	E. Soloway, B. Woolf, E. Rubin, P. Barth.	(3)	1981	-	Enseignement de concepts de programmation et reconnaissance des erreurs d'un étudiant lors de l'élaboration d'un programme PASCAL.	-	-	-	-	-	-	-	
MYCROFT	I.P. Goldstein	(22)	1975	États-Unis (MIT)	"Débug" de programmes dont le but est de dessiner certaines formes.	-	-	-	-	-	-	-	
NEOMYCIN	W.J. Clancey	1, 4, 15	1981	États-Unis (Stanford)	Enseignement assisté par ordinateur.	-	Règles de production	-	-	Chaînage avant	-	-	1) Deuxième version de GUIDON. 2) La connaissance a été complètement réécrite et étendue.
PHENARETE	H. Wertz	(3)	1982	-	Aide au "débug" d'un programme à trois niveaux: -surface (lexical + syntaxique) -profond (sémantique) -conceptuel	-	-	-	-	-	-	-	
PIXIE	D.H. Sleeman	(32)	1983	États-Unis (Univ. Stanford)	Intelligent tutoring system that infers a model of student performance and uses that model as a basis for teaching and remedial tutoring.	-	-	-	-	-	-	-	1) Enseignement pour des étudiants âgés entre 13 et 15 ans. 2) Enseigne l'algèbre linéaire de base et l'arithmétique. 3) Fonctionne sur micro-ordinateur.
PROGRAMMER'S APPRENTICE	C. Rich, H.E. Shrobe.	(2), 5, (C)	1976	États-Unis (MIT)	Aide les programmeurs en LISP à résoudre leurs "bugs".	-	-	-	-	-	-	-	

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
HAWKEYE	H.G. Barrow, R.C. Bolles, I.D. Garvey, J.H. Kremers, K. Lantz, J.M. Tenenbaum, H.C. Wolf.	(6)	1977	États-Unis	Cartographie et interprétation de photos.	-	-	-	-	-	-	-	1) Utilise LIFER.

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
WHY	A. Collins, A. Stevens.	(7)	1977	États-Unis (BBN)	Enseignement sur la cause des précipi-tations (pluie).	-	-	-	Recher-che	-	-	-	1) Extension de SCHOLAR.

CHIMIE

Description des systèmes experts

CONGEN

CRYSALIS

DENDRAL

LHASA

METADENDRAL

PIES

SECS

SYNCHEM

TQMSTUNE

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
CONGEN	R.E. Carhart	5	1979	-	-	-	-	-	-	-	-	-	
CRYSALIS	R. Engelmore, A. Terry.	1, 5, H	1979	États-Unis (Univ. Stanford)	Recherche de la structure de protéines à partir de résultats d'analyse cristallographique.	-	-	-	-	-	-	-	
DENDRAL	B.G. Buchanan	1, 2, (5), 9	1969	États-Unis (Univ. Stanford)	Recherche de la formule développée d'un corps organique.	-	Règles de production	-	Fonctionnel (Univ. Stanford + Labo. pharmaceutiques)	-	Son aptitude à expliquer les détails d'une structure moléculaire à partir de données chimiques dépasse celle des humains.	Aucun	
LHASA	E.J. Corey, W.T. Wipke.	(7)	1969	États-Unis (Univ. Harvard)	Analyse synthétique (chimie organique).	-	-	-	-	-	Hautes performances	-	1) Système interactif.
METADENDRAL	B.G. Buchanan	1, (5) 9	-	États-Unis (Univ. Stanford)	Programme d'apprentissage inductif ayant pour but d'améliorer et de généraliser des règles existantes de DENDRAL ou de générer de nouvelles règles. (Essaie de comprendre les lois qui régissent la fragmentation des molécules).	-	-	-	-	-	-	-	

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
PIES	M. Tenenbaum	(31)	-	États-Unis (Schlum-berger)	Diagnostic de pro-blèmes sur une ligne de fabrication de circuits	-	-	-	-	-	-	-	
SECS	W.T. Wipke	(1), 2 5	1978	États-Unis (Univ. Stanford)	Assistance au chi-miste pour la concep-tion de plans de synthèse de molécules complexes.	-	-	-	-	Chaînage arrière	-	-	
SYNCHEM	H.L. Gelernter	(5)	1977	États-Unis	Synthèse organique.	-	-	-	-	-	-	-	
TQMSTUNE	C. Wong	(31)	1985	- (Lawrence Livermore Natl. Labs)	Tune triple quadru-pole mass specto-meter.	-	-	-	-	-	-	-	
-	T. Boyd	(31)	-	- (Sydney)	Conseille afin de choisir un herbicide approprié pour une application spéci-fique.	-	-	-	-	-	-	-	
-	D. Michie	(31)	-	Angleterre	Screen new chemicals for herbicidal properties based on structure-activity relationships.	-	-	-	-	-	-	-	

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
TALIB	-	(9)	-	États-Unis (Univ. Carnegie-Mellon)	Conception de circuits intégrés.	-	-	-	Recher-che	-	-	-	
VEXED-LEAP	T. Mitchell	9, 11	-	États-Unis (Univ. Rutgers)	Conception de circuits VLSI. Système qui apprendra progressivement à aider l'utilisateur à concevoir un circuit remplissant une fonction définie par lui.	-	-	-	Recher-che	-	-	-	1) L'apprentissage est permanent et invisible pour l'utilisateur (pas de "mode d'apprentissage"). 2) L'information sera extraite sans que l'utilisateur en soit informé (système espion). 3) Si le système connaît le succès, il sera distribué à de nombreux exemplaires et pourra ainsi rencontrer un nombre de problèmes de construction de circuits bien plus grand que n'importe quel humain ne le pourrait au cours de sa vie. Le niveau d'expertise obtenu sera incomparable à celui d'un humain.
VT	J. McDermott, S. Marcus	(30), 31	-	États-Unis (CMU/West-inghouse)	Configuration d'ascenseurs.	-	-	-	Recher-che	-	-	-	
WRIGHT	C.A. Baykan, M.S. Fox.	(3)	1986	États-Unis (Univ. Carnegie-Mellon)	Conception de cuisines.	-	-	-	Recher-che	-	-	-	
XPSE	-	(30)	-	États-Unis (CAMI)	Planification.	-	-	-	-	-	-	-	
-	J.C. Bocquet, S. Tichkiewitch.	(1)	1982	- (E.N.S.E.I., Cahan)	A partir de dessins industriels à vues planes, construit des vues en perspective.	-	-	-	-	-	-	-	

CONTRÔLE DE PROCESSUS

Description des systèmes experts

AIRPLAN

ESTRAC-II

PDS

PICON

YES/MVS

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
AIRPLAN	-	(1), 2	1983	États-Unis (Univ. Carnegie Mellon)	Destiné à aider les décollages et récupérations d'appareils sur un porte-avions, en déterminant les problèmes futurs prévisibles.	-	-	-	-	-	-	OPS	
ESIRAC-II	S. Araya	(1)	1984	Japon (Mitsubishi Electric Corp.) et (Kinki Nippon Railway Co.)	Bâti pour modifier, en temps réel, les horaires et aiguillages de rames de trains lorsque des perturbations sont observées par rapport au trafic prévu.	-	-	-	-	-	-	-	
PDS ✔	-	(1)	1983	États-Unis (Westinghouse)	Destiné au diagnostic en temps réel d'incidents d'exécution de processus automatisés, surveillés directement par des capteurs.	-	-	-	-	-	-	EMYCIN	
PICON	-	(9)	-	- (Graphaël)	Contrôle de processus.	-	-	-	-	-	-	-	
YES/MVS	-	(1)	1984	États-Unis (IBM)	Conçu pour assurer, en temps réel, des tâches de conduite d'ordinateurs (pupitrage).	Environ 500	-	-	-	-	-	OPS5	

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
-	B. Dufay	(1)	1983	France (L.I.F. I.A., E.N.S.I.M. A.G., Grenoble)	Destiné à faciliter la synthèse des mouvements fins de bras manipulateurs.	-	-	-	-	-	-	-	1) Pour un objectif donné, un premier module détermine un plan d'actions simple et pilote son exécution en le réajustant au fur et à mesure. Le recueil des traces de plusieurs expériences constitue la "phase d'entraînement". Un deuxième module compile les traces pour fournir un programme exécutable plus général et donc plus robuste que tous les plans simples produits par le premier module: c'est la "phase d'induction". Le premier module ("entraîneur") comporte deux S.E.: le "planificateur" et le "contrôleur d'exécution".
-	R.B. Wesson	(1)	1977	États-Unis (Univ. du Texas à Austin)	Élabore des plans d'intervention radio auprès des différents avions concernés (et suit l'exécution des plans) à partir de paramètres caractérisant l'environnement d'une tour de contrôle.	-	-	-	-	-	-	-	

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
PROUST	E. Soloway, W.L. Johnson.	(3)	1985	-	Enseignement et compréhension de la démarche empruntée lors de l'élaboration d'un programme.	-	-	-	-	-	-	-	
PSM-NMR	-	(J)	-	-	Aide l'utilisateur à résoudre certains problèmes.	-	-	-	-	-	-	-	
SCENT	G.I. McCalla	(3)	1986	Canada (Univ. de Saskat-chewan)	Aide aux programmeurs (débutants) à l'élaboration de leurs programmes et à la résolution des problèmes rencontrés ("bugs") (niveaux conceptuel et logique).	-	-	LISP	Recher-che	-	-	-	1) Contient deux parties: - environnement de programmation; - conseiller.
SCHOLAR	J.R. Carbonnell	(10), 15	1970	-	Utilisé pour la géographie.	-	Réseaux sémantiques	-	-	-	-	-	
SITS	-	(J)	-	-	Aide à l'élaboration de programmes informatiques.	-	-	-	-	-	-	-	1) Capable d'améliorer ses stratégies d'enseignement.
SPADE	M.L. Miller	(10))	1982	-	Enseignement et apprentissage de programmes informatiques élémentaires (planification et "debugging").	-	-	-	Recher-che	-	-	-	1) Est plus précisément un "framework". 2) Communication en langue naturelle (anglais). 3) Utilise le langage Lo-Turtle pour construire des graphiques.
TVX	M. Billmers, M. Carifio.	(31)	1985	États-Unis (DEC)	Tutor user of VMS operating system.	-	-	-	-	-	-	-	

CONCEPTION ET FABRICATION ASSISTÉE PAR ORDINATEUR (CFAO)

Description des systèmes experts

ALADIN

CALLISTO

COPEST

DAA

DAS/LOGIC

GARI

HI-RISE

ISIS

LEAP

MOTOR BRUSH DESIGNER

PLEX

PHOTOLITHOGRAPHY ADVISOR

PRIDE

STOWAGE PLANNER

TALIB

VEXED-LEAP

VT

WRIGHT

XPSE

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
ISIS	M.S. Fox	(2), (3)	1980	États-Unis (Univ. Carnegie-Mellon)	Système utilisé pour la planification de l'atelier travaillant à la demande.	-	-	-	Recher-che	-	-	-	1) Plusieurs versions ont été construites (cf. document (3)).
LEAP	T.M. Mitchell, S. Mahadevan, L.I. Steinberg.	(11)	1985	États-Unis (Univ. Rutgers)	Système d'apprentis-sage appliqué au développement des conceptions de VLSI.	-	-	-	-	-	-	-	1) Capable de généraliser à partir d'exemples et de justifier adéquatement sa démarche. 2) Apprentissage par génération d'exemples. 3) Employé en parallèle avec VEXED.
MOTOR BRUSH DESIGNER	S. Dourson	(31)	-	- (Delco Products)	Construct design of brushes and springs for small electrical motors.	-	-	-	-	-	-	-	
PLEX	-	(9)	-	États-Unis (Bell Labo-ratories)	Conception de circuits intégrés.	-	-	-	-	-	-	-	
PHOTOLITHO-GRAPHY ADVISOR	T. Cline, W. Fong, S. Rosenberg, M. Walker.	(31)	1985	États-Unis (Hewlett-Packard)	Troubleshoot photo-lithography steps in circuit fabrication.	-	-	-	-	-	-	-	
PRIDE	S. Mittal	(31)	-	États-Unis (Xerox Park)	Crée et analyse de nouveaux "designs" pour les photo-copieurs.	-	-	-	-	-	-	-	
STOWAGE PLANNER	A. Miyajima	(31)	-	Japon	Développe des plans d'entreposage de cargaisons dans un entrepôt	-	-	-	-	-	-	-	

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
ALADIN	M.L. Farinacci, M.S. Fox, I. Hulthage, M.D. Rychener.	(3)	1986	-	Conception d'un alliage d'aluminium	-	-	-	-	-	-	-	
CALLISTO	-	(2)	-	États-Unis (Univ. Carnegie-Mellon)	Modèle, surveille, planifie et gère de grands projets.	-	-	-	Recher-che	-	-	-	
COPEST	-	(1)	-	-	Conception de pièces estampées.	300	-	FORTRAN	-	-	-	GOSSEYN	
DAA	-	(9)	-	États-Unis (Univ. Carnegie-Mellon)	Conception de circuits intégrés.	-	-	-	Recher-che	-	-	-	
DAS/LOGIC	-	(30)	-	États-Unis (CGI/DEC)	Conception de circuits VLSI	-	-	-	Fonc-tionnel	-	-	-	
GARI	Y. Descottes	1, 5	1980	France (E.N.S.I. M.A.G.) Grenoble	Engendre des gammes d'usinages à partir des spécifications géométriques et méca-niques.	Environ 50	-	-	-	-	-	-	
HI-RISE	-	(30)	-	États-Unis (CMU)	Conception d'édifices.	-	-	-	Recher-che	-	-	-	

GÉNÉRATEUR DE SYSTÈMES EXPERTS

Description des systèmes experts

✔EXPERT EASE

GPSI

OURSE

XPER

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
EXPERT EASE	D. Michie	9, 12 13	-	États-Unis	Générateur de systèmes-experts.	300	-	UCSD PASCAL	Fonc-tionnel	-	-	-	1) Fonctionne sur IBM-PC. 2) 695 $ US. 3) Disponible chez: Human Edge Software Inc., 2445 Farber Place, Palo Alto, CA 94303, USA, Tél.: (415) 493-1593. 4) Détecte les incohérences. 5) Peut générer des règles. 6) Utilisation simple 7) Admet trois types de structures: arborescente, arborescente avec boucle et chaînée. 8) Se présente sous la forme de chiffrier électronique. 9) Le cheminement d'une consultation est facile à retrouver. 10) Les étapes de création d'un système expert à l'aide de EXPERT-EASE se résume ainsi: - création des attributs; - donner des exemples; - création automatique de l'arbre de décision; - ajout du texte; - consultation. 11) Peut contenir jusqu'à 31 attributs ayant chacun jusqu'à 32 valeurs différentes. 12) On peut séparer un système expert en diverses composantes qui peuvent se relier entre elles.

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
GPSI	M.T. Harrandi	-	1984	États-Unis (Univ. Illinois à Urbana-Champaign)	Outil de réalisation de systèmes-experts.	-	Règles de production	-	Recher-che	Chaînage bidirec-tionnel	-	-	1) Peut utiliser des coefficients de vraisemblance. 2) Vérifie la consistance des entrées et permet d'éviter les redondances. 3) A été utilisé pour construire un S.E. dont la tâche était de "débugger" des programmes PASCAL.
OURSE	R. Pasero, A. David.	-	1985	France (Société Prologia)	Outil de réalisation de systèmes-experts.	Varia-ble	Logique du 1er ordre	PROLOG II	Produc-tion	SL-Résol-ution (arrière) + produc-tion (avant)	-	-	1) L'ordre dans lequel les règles sont utilisées n'a pas d'influence sur le résultat. 2) Le système peut répondre à des questions lui demandant d'expliciter son raison-nement. 3) Pendant la "phase de consultation" l'utilisateur peut demander au système où il en est rendu. 4) La communication entre l'utilisateur et le système se fait partiellement en langue naturelle. 5) Le système est capable de détecter les contradictions.
XPER	J. Lebbe	(13)	1984	France (Micro-applic-ation)	Générateur de systèmes-experts.	-	-	-	Fonc-tionnel	Chaînage avant	Capacité un peu limitée	-	1) Plus un système d'identification assisté par ordinateur qu'un S.E. 2) Fonctionne sur IBM PC/XT et compatibles, Apple et Commodore 64. 3) La base de faits est composée de trois groupes d'éléments: les individus, les variables et les modalités. 4) Possède la structure de base d'un S.E. mais en plus simple. 5) Comprend quatre modules: l'éditeur (création de la base des connaissances), le détermineur (moteur d'inférence

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
XPER (suite)													élémentaire), le réorganisateur (regroupement des bases de connaissances), module d'impression (impression). 6) Pour obtenir un système fiable, il faut plus qu'un IBM PC. 7) Moteur d'ordre 0. 8) Les entrées ne doivent être faites qu'en minuscules (français). 9) Le maximum possible, côté rubriques, est de 300 individus, 50 variables, 300 modalités (pas plus de 14 par variable).

GÉNIE

Description des systèmes experts

CARTER

CESSOL-O

COPART

DISPATCHER

ENGINE COOLING ADVISOR

EURISKO

ICLX

ISA

KBVLSI

SACON

SEARCH

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
CARTER	-	(9)	-	- (E.N.S. E.T.)	Ingénieur conseil en mécanique.	-	-	-	-	-	-	-	
CESSOL-0	M. Ayel, J.P. Laurent, M. Soutif.	(1), 13	1983	France (L.I.A., Univ. de Savoie)	Détermine les configurations d'essais géotechniques qui permettent de connaître la nature et les caractéristiques mécaniques des différentes couches d'un terrain (afin de définir les fondations à entreprendre lors de l'implantation d'un bâtiment).	-	-	PASCAL	-	-	-	Aucun	1) Peut justifier de manière très précise ses conclusions (explique pourquoi il rejette une telle conclusion, etc.).
COPART	—	(9)	-	- (E.N.S. E.T.)	Ingénieur conseil en mécanique.	-	-	-	-	-	Appliqué au domaine du jeu, il a gagné des championnats contre des joueurs humains en 1980, 1981, 1982. (Voir Mycro-Systèmes no 54, p. 151).	-	
DISPATCHER	J. McDermott	(31)	-	États-Unis (Carnegie-Mellon)	Schedule dispatching of parts for robots.	-	-	-	-	-	-	-	

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
ENGINE COOLING ADVISOR	S. Dourson	(31)	-	États-Unis	Diagnostic des causes de bruit dans le système de refroidissement des moteurs.	-	-	-	-	-	-	-	
EURISKO	D. Lenat	1, 2, 3, 9, 13	1980	États-Unis (Univ. Carnegie Mellon)	Découverte de concepts et d'heuristiques appliquées à: -jeux (wargame naval) -programmation -circuits électroniques à trois dimensions.	-	-	-	-	-	Appliqué au domaine du jeu, il a gagné des championnats contre des joueurs humains en 1980-81-82. (Voir Micro-Système No 54, p. 151).	-	
ICLX	B. Hakami, J. Newborn.	(31)	1983	-	Aid technicians diagnose faults in rod milling process.	-	-	-	-	-	-	-	
ISA	-	(31)	-	États-Unis (DEC)	Schedule orders for manufacturing and delivery.	-	-	-	-	-	-	-	
KBVLSI	-	2, 9	-	États-Unis (Xerox, Palo Alto) + (Univ. Stanford)	Système d'aide au développement des conceptions de VLSI.	-	-	-	Expérimental	-	-	-	

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
SACON	J.S. Bennett, R. Engelmore.	(1), 2, 5, 9	1979	États-Unis (Univ. Stanford)	Résistance des matériaux (aide des mécaniciens à communiquer avec un programme spécialisé écrit en FORTRAN).	160	-	-	-	-	-	EMYCIN	
SEARCH	C. Marche, M. Gaudette, G. Leclerc, D. Delmas.	(15)	-	Canada (École Polytechnique de Montréal)	Localisation des contaminants d'aquifères.	-	-	-	Développement	-	-	-	1) Conçu pour identifier rapidement la position et les caractéristiques d'une source contaminante responsable de la détérioration de la qualité de l'eau d'une nappe souterraine.
-	E.A. Feigenbaum	(31)	-	États-Unis	Contrôle le freinage de trains pour la précision et le confort.	-	-	-	-	-	-	-	
-	A. Miyajima	(31)	-	Japon	Detect cracks in billets and direct grinding.	-	-	-	-	-	-	-	

GÉOLOGIE

Description des systèmes experts

CUVEX

DIPMETER ADVISOR

DRILLING ADVISOR

ELFIN

GEOX

HYDRO

LITHO

MUDMAN

PROSPECTOR

SECOFOR

SIMMIAS

WAVES

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
CUVEX	-	(9)	-	France (ELF-Aqui-taine)	Recherche pétroli-fère.	-	-	-	-	-	-	-	
DIPMETER ADVISOR	R. Davis	1, 2, 5, 9	-	États-Unis (Schlumber-ger-USA)	Se préoccupe de la détermination des lithofaciès les plus plausibles en fonc-tion de paramètres recueillis au cours de la descente d'outils de forage (pétrole).	-	-	-	-	-	-	-	
DRILLING ADVISOR	C.R. Hollander, Y. Iwasaki, J.M. Courteille, M. Fabre.	2, 5	1983	France (ELF-Aqui-taine)	Diagnostique une va-riété de problèmes de forage et offre des conseils pour les rectifications néces-saires ainsi que les recommandations pour la prévention de ce genre de problèmes à l'avenir.	-	-	-	-	-	-	EMYCIN	1) Première version de SECOFOR.
ELFIN	R. Martin-Clouaire	(1)	1984	France (ELF-Aqui-taine, L.S.I., E.N.S.E.E. I.N.P.T., Univ. P. Sabatier)	Recherche les voies de migration d'hydro-carbures à partir des roches-mères jusque vers des roches-réservoirs, en pre-nant en compte les paramètres géolo-giques relatifs au sous-sol.	-	-	-	-	-	-	-	1) Capable de réaliser des raisonnements approximatifs.

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
GEOX	-	(31)	-	États-Unis (NASA)	Identify earth surface minerals from remotely sensed hyperspectral image data.	-	-	-	-	-	-	-	
HYDRO	-	(2)	-	États-Unis (SRI International)	Résolution des problèmes de ressources en eau.	-	-	-	-	-	-	-	
LITHO	A. Bonnet, J.G. Ganascia, J. Mary.	1, (5), 9 La recherche No 151	1981	France Clamart (Schlumberger-France)	Se préoccupe de la détermination des lithofaciès les plus plausibles en fonction de paramètres recueillis au cours de la descente d'outils de forage (pétrole).	500	Règles de production	-	-	-	La description des puits est généralement très proche de celle fournie à la main par un géologue humain.	EMYCIN	1) Les informations qui sont extraites se recoupent, souvent redondantes, quelquefois contradictoires. 2) Utilise des facteurs de crédibilité pour représenter les connaissances incertaines. 3) Possède une large connaissance. 4) Bonne interprétation des règles (sait reconnaître une même règle entrée sous plusieurs formes). 5) LITHO possède huit catégories de connaissances: géographie, activités tectoniques, stratigraphie, paléontologie, minéralogie, pétrographie, sédimentologie, pétrophysique. 6) Comme sortie finale, LITHO produit une description complète des puits.
MUDMAN	J. McDermott	(31)	-	États-Unis (CMU)	Diagnose problems in composition of drilling mud during oil well drilling.	-	-	-	-	-	-	-	

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
PROSPECTOR	P.E. Hart, R.O. Duda.	1, 2, 3,(5), 10, La re-cher-che No 151	1978	États-Unis (SRI)	Spécialisé dans la prospection minière et plus particulièrement la recherche pétrolière.	1 600	Réseaux sémantiques	-	-	Chaînage bidirec-tionnel	Découverte de deux dépôts importants de pétrole.	KAS	1) Reflète judicieusement une partie limitée mais appropri e du savoir et de l'expérience d'un expert. 2) Communication interactive en langage naturel. 3) Peut fournir des explications à tout moment de la consultation. 4) Représentation de l'incertitude selon un modèle de type Bayesien. 5) Utilise KAS pour la communication en langage naturel.
SECOFOR	-	(1), 9	1983	France (ELF-Aquitaine)	Diagnostic d'incidents de forage.	350	-	-	-	-	-	EMYCIN	
SIMMIAS	J.L. Laurière	(1)	1981	France (ELF-Aquitaine) + (Univ. Pierre et Marie Curie)	Recherche les voies de migration d'hydrocarbures à partir des roches-mères jusque vers des roches-réservoirs, en prenant en compte les paramètres géologiques relatifs au sous-sol (application de SNARK pour le suivi de forage).	-	-	-	-	-	-	-	
WAVES	-	(2)	-	-	Conseille les ingénieurs sur l'utilisation des programmes d'analyse des données sismiques: industrie pétrolière.	-	-	-	-	-	-	KS-300	

INFORMATIQUE

Description des systèmes experts

AI-SPEAR

BDS

CDx

CONAD

CRIB

CSS

DART

DAS

DIAG8100

DIG VOLTAGE TESTER

DOC

FAULTFINDER

HICLASS

IDT

INFORMAT

INFORMATIQUE (suite)

Description des systèmes experts

IPWBIS

NTC

OCEAN

PINE

R1 (XCON)

SPEC

SYERA

XSEL

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
AI-SPEAR	-	(1), 2, 9	1984	États-Unis (D.E.C.)	Développé pour l'analyse des "cahiers de bord" d'ordinateurs afin de détecter et diagnostiquer d'éventuelles anomalies.	700	Règles de production	-	-	-	-	OPS5	
BDS	W. Perkins	(32)	-	États-Unis (Lockheed)	Troubleshoot baseband distribution subsystem of communications hardware.	-	-	-	-	-	-	-	
CDx	N. Pundit	(31)	-	États-Unis (DEC)	Analyse VMS dump files after system crashes.	-	-	-	-	-	-	-	
CONAD	S. Savory	(31)	1984	-	Check order entry and configure computer systems.	-	-	-	-	-	-	-	
CRIB	R.T. Hartley	), (18)	1981	États-Unis	Diagnostic de pannes d'ordinateurs (émet des hypothèses des causes possibles de ces pannes).	-	-	-	-	-	-	-	1) L'acquisition des connaissances est plutôt ardue.
CSS	A. Wayne Elwood	(31)	-	États-Unis (IBM)	Aid in planning relocation reinstallation and rearrangement of IBM mainframes.	-	-	-	-	-	-	-	
DART	J.S. Bennett	-	1981	États-Unis (IBM)	Conçu pour le diagnostic de pannes d'ordinateurs.	190	-	LISP	-	-	-	EMYCIN	

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
DAS	-	(30)	-	États-Unis (CMU)	Conception d'ordina-teurs.	-	-	-	-	-	-	-	
DIAG8100	L. Weeks	(31)	1985	-	Diagnostique les défauts dans "DP equipment".	-	-	-	-	-	-	-	
DIG VOLTAGE TESTER	T. Laffey	(31)	-	États-Unis (Lockheed)	Aid troubleshooting digital voltage sources in testing lab.	-	-	-	-	-	-	-	
DOC	-	(30)	-	États-Unis (Prime)	Diagnostic de pannes d'ordinateur.	-	-	-	Fonc-tionnel	-	-	-	
FAULTFINDER	S. Savory	(31)	1984	-	Diagnostique les défauts dans les "disk drives".	-	-	-	-	-	-	-	
HICLASS	L. Waller	(31)	1985	États-Unis (Hughes Electro-Optical and Data Systems)	Sequence steps in PC board assembly.	-	-	-	-	-	-	-	
IDT	-	(30)	-	États-Unis (DEC)	Vérification d'ordinateurs.	-	-	-	Fonc-tionnel	-	-	-	
INFORMAT	J. Alden	(31)	-	États-Unis	Conseille les con-sommateurs sur les achats d'ordinateurs.	-	-	-	-	-	-	-	
IPWBIS	-	(30)	-	États-Unis (CMU/West-inghouse)	Circuit board inspection.	-	-	-	Fonc-tionnel	-	-	-	

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
NTC	P. Politakis, W. Hichson.	(31)	1984	États-Unis (DEC)	Troubleshoot problems related to Ethernet & DECnet networks.	-	-	-	-	-	-	-	
OCEAN	—	(30)	-	États-Unis (NCR)	Configuration d'ordinateurs.	-	-	-	Fonc-tionnel	-	-	-	
OCEAN	B. Plotkin	(31)	-	États-Unis (Teknowl-edge)	Check orders for computer systems, configure orders.	-	-	-	-	-	-	-	
PINE	A. Wayne Elwood	(31)	-	États-Unis (IBM)	Guide les personnes lors de l'élaboration de rapports sur l'analyse de problè-mes de logiciels.	-	-	-	-	-	-	-	
R1 (XCON)	J. McDermott	1, 2, 13, 14	1980	États-Unis (Univ. Carnegie Mellon)	Se charge de la con-figuration des ordi-nateurs VAX.	3 000	-	-	-	-	ses premières utilisations pour les ré-solutions de problèmes, sa précision n'était que de 60% alors qu'en labo-ratoire elle atteignait 90%.	-	
SPEC	-	1, 9	1983	France (Louve-ciennes)	Aide à la conception de configurations particulières des ordinateurs VAX.	-	-	-	Recher-che	-	-	-	

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
SYERA	-	(9)	-	- (Cap-Sogeti Innovation)	Réalisation de conseillers en conception de gros programmes.	-	-	-	-	-	-	-	
XSEL	J. McDermott	1, 2, 5, 9	1982	États-Unis	Destiné à faciliter la communication entre RI et les ingénieurs technico-commerciaux amenés à l'utiliser.	-	-	-	-	-	-	OPS	
-	-	9	-	États-Unis (MIT)	Diagnostic des défaillances des VAX.	-	-	-	-	-	-	-	
-	T. Boyd	(31)	-	- (ICL, Sydney)	Configuration des ordinateurs, séries 39.	-	-	-	-	-	-	-	
-	D. Michie	(31)	-	Écosse (Turing Institute)	Diagnose faults on printed circuit boards.	-	-	-	-	-	-	-	
-	D. Michie	(31)	-	Écosse (Turing Institute)	Intelligent front-end for complex software.	-	-	-	-	-	-	-	

JEUX

"... Il n'y a plus qu'une centaine de chercheurs dans le monde qui s'intéressent à ce domaine."

(La Recherche no 170)

Description des systèmes experts

AL3

ATHENA

BELLE

BKG 9.8

CHELEM

CHESS

CHUNKER

IAGO

JOSEPHINE

PARADISE

ROBIN

WUSOR-1

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
AL3	-	(9)	-	-	Échecs.	-	-	-	-	-	-	-	1) De la trame de CHOKER.
ATHENA	S. Chapleau, G. Lapalme.	(9)	1983	Canada	Scrabble.	-	-	-	-	-	En version "duplicatée" le système joue mieux que les champions du monde (98% du top, c'est-à-dire le maximum possible.	-	1) Système simple qui, à l'aide d'un dictionnaire adapté et d'une petite fonction heuristique de choix guidant une recherche en une seule passe et sans retour en arrière, joue mieux lors de tournois "duplicatés" (tous ont les mêmes lettres) que les champions du monde.
BELLE	-	9, 13	-	États-Unis	Échecs.	-	-	-	-	-	Loin derrière les grands maîtres (reste quand même le meilleur programme d'échecs à l'heure actuelle).	-	1) Utilise un matériel spécialisé qui lui permet d'étudier 150 000 positions par seconde.
BKG 9.8	H.J. Berliner	9, 13, G	1979	États-Unis (Univ. Carnegie-Mellon)	Backgammon.	-	-	-	-	-	A vaincu le champion du monde le 15 juillet 1979.	-	1) Utilise de connaissances et des propriétés du jeu, sous forme d'évaluations fonctionnelles adéquates et non linéaires et fait ainsi très peu de recherche dans les arbres des états.
CHELEM	-	(13)	-	-	Bridge.	-	-	-	-	-	-	-	

NOM	AUTEUR(S)	RÉFÉRENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉSENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTIONNEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
CHESS	-	9, 13, F	1970	États-Unis (Univ. de la Côte Ouest)	Échecs.	-	-	-	-	-	Meilleur programme dans les années '70.	-	1) Comprend certaines connaissances stratégiques et positionnelles. 2) Plusieurs versions.
CHUNKER	H.J. Berliner, M. Campbell.	9, 13	1983	-	Échecs.	-	-	-	-	-	Bat régulièrement ses auteurs dans des situations nouvelles.	-	1) Raisonnement de fin de jeu assez rapide (une minute alors qu'il prendrait 10^{13} années aux programmes traditionnels les plus puissants). 2) Possède la particularité de pouvoir raisonner globalement sur tout un groupe de pièces et de tenir compte des concepts géométriques. 3) Traite des finales Roi-pions avec des temps de réponse très satisfaisants.
IAGO	P. Rosenbloom	(13)	1981	États-Unis	Othello.	-	-	-	-	-	Il atteint un "top niveau" mais n'a pu être confronté avec le champion du monde.	-	1) Construit en cinq mois. 2) Sa fonction d'évaluation tient compte de la stabilité territoriale et de la mobilité.
JOSEPHINE	B. Faller	(13)	-	France (L.R.I.) Orsay	Bridge (fournit des plans de jeu pour les déclarants à sans-atout, après saisie des enchères, des mains du déclarant et du mort, de l'entame).	Plusieurs centaines	-	-	-	-	-	-	1) Capable de formuler des hypothèses sur les répartitions des couleurs et le placement de certaines cartes, et d'en déduire la ligne de jeu optimale. 2) Conduit un raisonnement explicite.

NOM	AUTEUR(S)	RÉFÉRENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉSENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTIONNEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
PARADISE	D. Wilkins	9, 13, F	1979	États-Unis (SRI International)	Échecs.	Environ 200	Règles de production	-	-	Chaînage avant	-	-	1) Essaie d'expliciter certaines connaissances, certains concepts spécifiques des échecs, à la manière des humains, afin de diminuer le nombre de coups à considérer.
ROBIN	J. Pitrat	9, 13	1977	France	Échecs.	-	-	-	-	-	-	-	
WUSOR-1	J. Stansfield, B. Carr, I.P. Goldstein.	(10)	1976	États-Unis (MIT)	Wumpus.	-	-	-	-	-	-	-	
-	B. Safar	(9)	-	France (L.R.I., Univ. Paris-Sud)	Bridge (enchères).	-	-	-	-	-	-	-	1) Le système d'enchères est accompagné d'un module très sophistiqué d'explication capable de répondre de façon adéquate à des questions du style: "Pourquoi le système n'a-t-il pas annoncé telle enchère?".
-	R. Popescu	(9)	1983	-	Bridge.	Plusieurs centaines	Règles de production	-	-	-	S'approche du niveau de bons joueurs humains.	SNARK	
-	A. Wasserman	(9)	1970	-	Bridge.	-	-	-	-	-	-	-	1) Capable d'enchérir convenablement.
-	A. Samuel	(13)	1959	-	Dames.	-	-	-	Fonctionnel	-	A battu de grands joueurs.	-	1) Capable d'acquérir une certaine expérience en se rappelant des situations analysées (180 000 parties). 2) L'un des premiers programmes à utiliser des techniques arborescentes et aussi l'un des premiers grands succès de l'intelligence artificielle.

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
-	E. Jensen, T. Truscott.	(13)	1974	États-Unis	Dames.	-	-	-	-	-	-	-	1) Même principe que le système de A. Samuel.
-	D.A. Waterman	9, 13, 14	1970	-	Poker.	20	-	-	-	-	Gagne contre des joueurs moyens et joue souvent au niveau des experts.	-	1) Capable de bluffer. 2) Afin de savoir s'il doit suivre, relancer ou abandonner, le système tient compte de plusieurs paramètres: - valeur de sa main; - valeur du "pot"; - inclinaison à bluffer de l'adversaire (a tendance à toujours suivre, au début, afin de déterminer ce paramètre).
-	E. Berlekemp	9, 13	-	-	Bridge.	-	-	-	-	-	-	-	
-	P. Pionchon	9, 13	1983	-	Bridge.	-	-	-	-	-	-	-	1) Appliqué sur micro-ordinateur.

LANGUE NATURELLE

Description des systèmes experts

AUTOLING

BAOBAB

BASEBALL

BORIS

CONVERSE

COOP

CYRUS

DEACON

DIALECT

DRAGON

ELIZA

ESOPE

GENIAL

HACKER

HARPY

✔ HEARSAY (I et II)

HWIM

INTELLECT

KEAL

LANGUE NATURELLE (suite)

Description des systèmes experts

LADDER/INLAND

LANDSCAN

LIFER

LUNAR

MARGIE

MARWORDS

MYRTILLE (I et II)

NEXUS

PAM

PARRY

PLANES

QUIST

RAREAS

RESEARCHER

SAPHIR

SDC

SESAME

SHRDLU

SIR

LANGUE NATURELLE (suite)

Description des systèmes experts

SPEECHLIS

SPIRIT

STUDENT

TLC

TRACK

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
AUTOLING	S. Klein	(9)	1968	États-Unis	Apprend la grammaire d'une langue à partir de phrases correctes et incorrectes.	-	-	-	-	-	-	-	
BAOBAB	-	(5)	-	-	Analyse de résumés d'états de patients décrits en un langage quasi naturel par des médecins.	-	Frames	-	-	-	-	EMYCIN	1) La compréhension du programme est reflétée par sa capacité de: suggérer des informations manquantes, faire des inférences, faire des vérifications de cohérence, interrompre opportunément l'utilisateur en demandant de clarifier certaines informations. 2) Le but est atteint au moyen d'un dialogue en langue naturelle, l'analyseur utilisant une grammaire de nature sémantique.
BASEBALL	B. Green et al.	(6)	1960	États-Unis (Lincoln Labora-tories)	Réponse à des questions se rapportant au baseball.	-	-	IPL-V	-	-	Se rapprochant de la médiocrité.	Aucun	1) Programme très élémentaire. 2) Les questions posées ne peuvent avoir plus d'une proposition ou contenir des conjonctions et/ou des superlatifs. 3) Le programme n'est jamais modifié (base des données). 4) Le seul trait intéressant est qu'il peut reconnaître une même question posée de manières différentes.
BORIS	W. Lehnert, M.G. Dyer, P.N. Johnson, S. Harley.	5, 10, (13)	1983	États-Unis	Essaie de comprendre "en profondeur" les motivations des acteurs impliqués dans des histoires de divorce.	-	-	-	-	-	-	-	1) Les "analyseurs" supposent que toutes les éventualités, toutes les situations qui peuvent se produire dans le déroulement d'une scène se trouvent déjà inscrites dans la connaissance du système (ex: utilisations d'une chambre à coucher). Un manque de données peut jouer un rôle déterminant.

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
BORIS (suite)													2) La compréhension qu'il tire des histoires est très riche en détails. 3) Capable de trouver les motivations des personnages, d'identifier des situations caractéristiques et mettre en relation des événements (sans mécanismes tels les scénarios, évidemment). 4) Les résultats (performances) peuvent être fautifs puisque le système n'a été appliqué qu'à quatre histoires. 5) Représentation des connaissances par des MOPs. 6) Associe étroitement compréhension du texte et différentes formes d'analyse syntaxique et sémantique.
CONVERSE	Kellogg	(6)	1968	-	Langue naturelle.	-	-	-	-	-	-	-	
COOP	S.J. Kaplan	(7)	1979	États-Unis (Univ. de Pennsylvanie)	Langue naturelle.	-	-	-	-	-	-	-	1) Corrige un manque (erreur) dans les questions posées (ex: combien d'étudiants ont échoué un certain cours à l'automne dernier; le système répond que ce cours ne se donnait pas à l'automne mais à l'hiver).
CYRUS	-	(5)	-	-	Teste l'hypothèse émise par de nombreux psychologues, à savoir la nature "reconstructive" des mécanismes impliqués dans la mémorisation et le souvenir.	-	-	-	-	-	-	-	
DEACON	Thompson	(6)	1966	-	Langue naturelle.	-	-	-	-	-	-	-	

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
DIALECT	J.-C. Bassano	(3)	1986	France (Univ. Paris Sud)	Indique des sources à consulter, en réponse à des questions posées en langue naturelle (assistant en documentation).	-	-	PL1	Recher-che	-	-	-	1) Communication en français. 2) Possède un dictionnaire. 3) Évolution constante dans l'acquisition LAE dans de nombreux domaines: descriptions de documents, règles pour des stratégies de recherche, règles de linguistique, etc. 4) Capacités d'apprentissage permettant l'amélioration des performances dans les procédés linguistiques et la reformulation des questions.
DRAGON	-	(5)	1978	États-Unis (IBM)	Compréhension de la parole.	-	-	-	-	-	-	-	
ELIZA	J. Weizenbaum	5, 6, 9,(10)	1966	États-Unis (MIT)	Compréhension des langues naturelles.	-	-	SLIP	-	-	-	-	1) Aucune intelligence. 2) Programme traditionnel. 3) Simple illusion de compréhension (il n'en est rien en réalité, il procède suivant un "pattern"). 4) le plus célèbre de son domaine. 5) ELIZA simule un psychiatre avec lequel on peut converser en langue naturelle. 6) Possède un ensemble de couples (pattern reconnu = pattern réponse), c'est-à-dire que chaque fois qu'il reconnaît un certain pattern, il génère le pattern réponse correspondant.
ESOPE	-	(5)	-	États-Unis (LIMSI, Orsay)	Compréhension de la parole.	-	-	-	-	-	-	-	
GENIAL	B. Pelletier, J. Vaucher.	(3)	1986	Canada (Univ. de Montréal)	Langue naturelle (français).	-	-	PROLOG	Recher-che	-	47% des questions posées, lors	PROLOG	1) Une attention toute particulière a été portée à l'analyse morphologique.

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
GENIAL (Suite)											de la pre-mière expé-rimentation, ont été rejetées par le syatème.		2) Impossibilité de répondre à des méta-questions (ex: Que puis-je faire? Que savez-vous? A quoi sert votre sya-tème?, etc.). 3) Le traitement se divise en quatre étapes: - analyse lexicale; - analyse morphologique; - analyse syntaxique et sémantique; - génération de la réponse. 4) Peut être appliqué à n'importe quel domaine. 5) Implémenté sur VAX 780. 6) Le lexique est implanté en arbre IRIE. 7) Tolère les fautes d'accord.
HACKER	G.J. Sussman	(10)	1975	États-Unis	Langue naturelle et programmation automa-tique: écrit un pro-gramme pour répondre à une demande for-mulée par l'utilisa-teur (ex: placer le bloc A sur le bloc B).	-	-	CON-NIVER	-	-	-	-	1) Apprend à partir de ses erreurs. 2) Les programmes qu'il écrit contiennent des commentaires, tout comme ceux des humains, indiquant à quoi sert une sous-routine. 3) Ses programmes ne peuvent qu'exécuter une chose à la fois (ex: ne pourrait placer le bloc A sur le bloc B puis placer le tout sur le bloc C. Il placerait le B sur le C puis le A sur le B).
HARPY	B. Lowerre, R. Reddy.	2, 5, 9	1980	États-Unis (Univ. Carnegie-Mellon)	Compréhension de la parole.	-	-	-	-	-	-	-	1) Vocabulaire limité. 2) Passage d'un interlocuteur à un autre manquant d'élégance. 3) Primitif.

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
HEARSAY (I, II et III)	L.D. Herman, F. Hayes-Roth, V.R. Lesser, D.R. Reddy.	2, 5, H	1980	États-Unis (Univ. Carnegie-Mellon)	Compréhension de la parole.	-	-	-	-	-	-	-	1) Vocabulaire limité. 2) Les passages d'un interlocuteur à un autre manquent d'élégance. 3) Primitif. 4) Plusieurs versions ont été produites: leur but a été surtout d'apporter des idées utiles sur la façon d'organiser, d'utiliser et d'améliorer la base de connaissances. 5) La connaissance dans le système est représentée sous la forme de "sources de connaissance", une sorte de règle de production complexe. 6) Chaque "source de connaissance" contient la connaissance nécessaire pour résoudre une application particulière. 7) Il est possible de choisir la granularité de la connaissance que l'on désire. 8) Chaque "source de connaissance" est caractérisée par une condition de déclenchement qui lui est attachée.
HWIM	J. Wolf, W. Woods.	(5)	1980	États-Unis (BBN)	Compréhension de la parole.	-	-	-	-	-	-	-	
INTELLECT	L. Harris	3, 9	1979	États-Unis	Langue naturelle.	-	-	-	Commer-cialisé	-	-	-	
KEAL	G. Mercier, M. Gérard.	(5)	1981	France (CNET de Lannion)	Compréhension de la parole.	-	-	-	-	-	-	-	
LADDER/INLAND	E.D. Sacerdoti	3, (5)	1977	États-Unis	Interrogation de bases de données sur les navires.	-	-	-	-	-	-	-	

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
LANDSCAN	R. Bajcsy, A. Joshi, E. Krotkov, A. Zwarico.	(12)	1985	États-Unis (Univ. de Pennsylvanie)	Langage naturel et système de vision.	-	-	-	-	-	-	-	
LIFER	G. Hendrix	5, 9	1974	États-Unis (SRI International)	Interpréteur de règles de grammaire.	-	-	-	-	-	-	-	
LUNAR	W. Woods, R. Kaplan, B. Nash-Weber.	3, 5, (6)	1972	États-Unis (BBN)	Compréhension de phrases isolées (interrogation de bases de données en langue naturelle) se rapportant à la géologie lunaire.	-	Réseaux sémantiques	-	Expérimental	-	-	-	1) Compréhension au niveau des phrases prises individuellement sans essayer de les relier en un discours continu. 2) Technique d'analyse ATN. 3) Possède un dictionnaire d'environ 3 500 mots anglais. 4) L'analyse des questions se fait en trois étapes: - analyse syntaxique; - interprétation sémantique; - exécution de la requête formelle commandant la recherche dans la base de données et la réponse en langue naturelle. 5) A donné naissance à INTELLECT, commercialisé.
MARGIE	C. Riesbeck, C. Rieger, N. Goldman.	(5)	-	-	Analyse des phrases exprimées en langue naturelle.	-	-	-	-	-	-	-	1) L'analyse des phrases est transformée en une représentation de dépendance conceptuelle. 2) Peut travailler en mode paraphrase ou inférence.

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
MARWORDS	R. Kittredge, A. Polguere, E. Goldberg.	(3)	-	Canada (Univ. de Montréal)	Compile les prévi-sions de paramètres météorologiques basées sur des obser-vations à grande échelle.	-	-	-	Recher-che	-	-	-	
MYRTILLE (I et II)	J.P. Haton, G. Messenet, J.M. Pierrel, C. Sanchez.	(5)	1978-1982	France (CRIN, Univ. de Nancy)	Compréhension de la parole.	-	-	-	-	-	-	-	1) Capable de reconnaître des phrases de langages artificiels très contraints et un vocabulaire inférieur à 100 mots. 2) Appliqué au domaine des renseignements météorologiques, avec un vocabulaire de 375 mots et une syntaxe assez proche du français parlé.
NEXUS	R. Alterman	(28)	1985	États-Unis (Univ. du Texas à Austin)	Résume des histoires et répond à des questions s'y rap-portant.	-	-	LISP	Recher-che	-	Très promet-teuses	-	1) Dictionnaire comprenant entre 100 et 150 concepts (événement/état).
PAM	R. Wilensky	5, 13, E	1978	États-Unis (Univ. de Yale)	Compréhension des textes.	-	-	-	-	-	Résultats plutôt ternes	-	1) Grande richesse des mécanismes internes.
PARRY	-	(10)	-	-	Simule un psychiatre.	-	-	-	-	-	-	-	
PLANES	D. Waltz, B. Goodman.	(5)	1977	États-Unis (MIT)	Interface en langue naturelle avec des bases de données, pour les avions.	-	-	-	-	-	-	-	
QUIST	J.J. King	(32)	1981	États-Unis (Univ. Stanford)	Langue naturelle.	-	-	-	Recher-che	-	-	-	1) Reformule (interprète) des questions afin de trouver une réponse dans la base des données.

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
RAREAS	R. Kittredge, A. Polguere, E. Goldberg.	(3)	-	Canada (Univ. de Montréal)	Synthèse des bulletins météo de la marine à partir des prévisions météorologiques.	-	-	-	Recher-che	-	-	-	1) Manque de raffinement.
RESEARCHER	M. Lebowitz	(12)	1985	États-Unis (Univ. New York)	Système d'information intelligent.	-	-	-	Recher-che	-	-	-	
SAPHIR	-	5, 9	1983	France (ERLI)	Interrogation d'une base de données en langue naturelle.	-	-	-	-	-	-	-	1) Domaine d'application limité (minuscule). 2) Compréhension fine. 3) Trois étapes à l'analyse: morpho-lexicale, syntaxique (200 règles), sémantique (environ 100 règles).
SDC	-	(6)	-	États-Unis (SRI International)	Langue naturelle.	-	-	-	-	-	-	-	
SESAME	Y. Ali, R. Aubin, B. Hall.	3	1986	Canada (BNR, Ottawa)	Langue naturelle.	-	-	-	Recher-che	-	-	-	1) Lorsqu'une entrée est faite le système peut procéder temporairement à un changement de mot, pour un synonyme, permettant ainsi de ne pas encombrer l'espace-mémoire. 2) Corrige (ou du moins essaie) les erreurs de frappe. 3) Indépendant du domaine d'application.
SHRDLU	T. Winograd	5,(6), 9, 10	1972	États-Unis (MIT)	Dialogue interactif en langue naturelle avec un utilisateur (le système simule un bras manipulateur déplaçant des blocs	-	-	LISP + MICRO-PLANNER	-	-	-	MICRO-PLANNER	1) Accepte des phrases et des commandes aussi bien que des questions sur son raisonnement. 2) Possède quatre parties: - analyse morphologique; - grammaire anglaise;

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
SHRDLU (suite)					(jouets) sur une table).								- analyse sémantique; - résolution du problème. 3) La communication avec le système ne doit pas se faire en anglais trop complexe (structure de phrases).
SIR	B. Raphael	5, (6)	1968	États-Unis (MIT)	Conversation en langue naturelle (anglais).	-	Logique des prédicats	LISP	Proto-type	-	Modestes vu le nombre de connaissan-ces.	Aucun	1) Base de connaissances passablement limitée. 2) Conversation plutôt modeste. 3) Le système peut interroger l'utilisateur afin de résoudre certaines ambiguïtés.
SPEECHLIS	-	(5)	-	États-Unis (MIT)	Compréhension de la parole.	-	-	-	-	-	-	-	
SPIRIT	-	(9)	-	France (Société Systex)	Gère de grosses bases de données textuel-les, principalement à des fins documen-taires (utilisé dans les administrations pour divers usages tels que la recherche documentaire en matière énergétique ou de textes juri-diques).	-	-	-	Commer-cialisé par la CISI	-	90% à 96% de résultats satisfai-sants suivant la nature du texte.	-	1) Étapes: - analyse morpholexicale; - analyse syntaxique; - analyse grammaticale (utilisation de probabilités pour les mots ambigüs, leur assignant une catégorie la plus probable); - analyse sémantique. 2) Une extension est envisagée à la gestion de coupures de presse et à celle du courrier.
STUDENT	D. Bobrow	1, (6), 9	1968	États-Unis (MIT)	Lecture et résolution de problèmes d'algè-bre (de niveau secon-daire donnés sous forme d'histoires).	-	-	-	-	-	Résout les problèmes aussi bien qu'un élève et surtout plus rapi-dement.	Aucun	1) Si le système est incapable de résoudre l'équation, il peut demander à l'utili-sateur de lui fournir de l'information supplémentaire.

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
TLC	Quillian	(6)	1969	-	Langue naturelle.	-	-	-	-	-	-	-	
TRACK	S. Carberry	(3)	1986	États-Unis (Univ. du Delaware)	Se veut un système pouvant entretenir un dialogue cohérent en langue naturelle.	-	-	-	-	-	-	-	
-	S.G. Pulman, G.J. Russell, G.D. Ritchie, A.W. Black.	(32)	1986	États-Unis (Univ. Cambridge) + Royaume-Uni (Univ. d'Edin-burgh)	Langue naturelle.	-	-	-	Recher-che	-	-	-	1) Le système devrait être complété en 1987.
-	H. Thompson, J. Phillips.	(32)	1986	Royaume-Uni (Univ. d'Edin-burgh)	Langue naturelle (parsing system).	-	-	-	Recher-che	-	-	-	
-	G. Sabah	(9)	-	France	Compréhension d'une langue naturelle (histoires pour enfants).	-	-	-	-	-	-	-	1) Comprend les fables de Lafontaine.
-	P. Harrison, M. Maxwell.	(9)	1984-1986	États-Unis (Boeing Artificial Intelli-gence Center)	Langage naturel.	Environ 200	-	LISP	Recher-che	-	-	-	
-	R. Schank et al.	(9)	-	États-Unis (Yale)	Langue naturelle.	-	-	-	-	-	-	-	

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
-	P. Harrison, M. Maxwell.	(3)	1984	États-Unis (Boeing Artificial Intelli-gence Center)	Réponse à des questions posées en langue naturelle.	-	-	-	Recher-che	-	-	-	1) La communication se fait par l'entremise du clavier et de l'écran. 2) A été implémenté sur le VAX 11/780. 3) Système incomplet.
-	A. Amin	(9)	-	France (Nancy)	Reconnaissance de l'écriture arabe.	-	-	-	-	-	-	-	

MATHÉMATIQUES

Deecription des syatèmes experts

ALICE

AM

BUGGY

CAMELIA

CEG

DEBUGGY

GELERNTER

GPS

GRAPHER

HARP

KRYPTON

LEX

LOGIC-THEORIST

LPS

MACSYMA

MUSCADET

PM

PREDICTOR

PROVER

MATHÉMATIQUES (suite)

Description des systèmes experts

QA3

REDUCE

REF-ARF

SAINT

SEME

SIN

SNARK-INTEGRATION

WEST

ZORBA

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
ALICE	J.L. Lauriere	(5)	1978	France (Univ. Paris VI)	Résout des problèmes de logique et de mathématiques posés dans un langage utilisant le vocabulaire de la théorie des ensembles et de la logique classique.	-	-	-	-	-	-	-	
AM	D. Lenat	(1), 5, 9, 10	1977	États-Unis (Univ. Carnegie-Mellon)	Proposer des conjectures - sans les démontrer - et élaborer de nouveaux concepts (plusieurs centaines) à partir d'une base initiale de concepts généraux (une centaine).	500	Règles de production	-	-	-	-	-	1) Pas d'apprentissage (n'utilise pas ses nouvelles connaissances pour améliorer ses performances). 2) Ne peut découvrir de nouvelles heuristiques.
BUGGY	R. Burton, J.S. Brown, K.M. Larkin.	(7), 15	1982	États-Unis (BBN)	Habiletés arithmétiques (diagnostic des "bugs" ou erreurs procédurales).	-	-	-	-	-	-	-	1) - Build a differential student model. - Explicate rules of when to interrupt. - Procedural representation for bugs. 2) Explique pourquoi un étudiant fait des fautes.
CAMELIA	M. Vivet	(1)	1984	États-Unis (Univ. du Maine)	Destiné à conjuguer des connaissances heuristiques avec les algorithmes puissants du système de calcul mathématique REDUCE.	-	-	-	-	-	-	-	
CEG	-	(11)	-	-	Génération d'exemples mathématiques.	-	-	LISP	-	-	-	-	1) Puissance et habileté d'expression limitées.

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
DEBUGGY	Burton, Brown.	(15)	1982	États-Unis	Habiletés arithmé-tiques (diagnostic des "bugs" ou erreurs procédurales).	-	-	-	-	-	-	-	1) - Build a differential student model. - Explicate rules of when to interrupt. - Procedural representation for bugs. 2) Explique pourquoi un étudiant fait des fautes.
GELERNTER	H. Gelernter	(6)	1959	États-Unis (New-York)	Résolution de pro-blèmes de géométrie.	-	Règles de production	FORTRAN	Fonc-tionnel	-	-	Aucun	1) Fonctionne sur IBM 704.
GPS	A. Newell, J.C. Shaw, H.A. Simon.	1, 6, H	1969	États-Unis (Univ. Carnegie-Mellon)	Résolution de pro-blèmes mathématiques; preuve de théorèmes.	-	Règles de production	-	Recher-che	-	Performances très mo-destes; il s'agit d'un des premiers modèles ex-périmentaux ayant été fabriqués. On a d'ail-leurs cessé les travaux sur ce mo-dèle.	Aucun	1) Se rapproche beaucoup du programme tra-ditionnel. 2) Successeur de Logic Theorist.
GRAPHER	A.H. Ballantyne, W.W. Bledsoe.	(11)	-	États-Unis (Univ. du Texas) Austin	Génération d'exemples	-	-	-	-	-	-	-	
HARP	F. Oppacher, E. Suen.	(3)	1986	Canada (Univ. Carleton) Ottawa	Preuve de théorèmes (géométrie plane).	-	-	ZETA-LISP	Recher-che	-	-	-	1) Implémenté sur Symbolics 3600. 2) Peut résoudre: - problème du singe et de la banane; - problèmes de géométrie plane; - Schubert's steamroller problem (en 14 secondes pour la version non modifiée).

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
KRYPTON	R.J. Brachman, V.P. Gilbert, H.J. Levesque.	(11)	1983	États-Unis	Preuve de théorèmes.	-	-	-	-	-	-	-	1) Développé à partir des résultats obtenus de KL-ONE. 2) Deux composantes majeures se spécialisant respectivement dans le raisonnement terminologique. 3) Chaque composante a son propre langage.
LEX	T. Mitchell et al.	5, 9	1983	-	Apprentissage de techniques de résolutions d'intégrales.	-	-	-	-	-	Comparables à celles d'un homme entraîné.	-	1) Ne peut pas calculer d'intégrales définies. 2) Premier S.E. qui ait été capable d'acquérir son expertise tout en fournissant des explications sur ses processus d'acquisition. 3) Capable de trouver, sans essayer toutes les méthodes systématiquement, la bonne méthode d'intégration. 4) Capable d'auto-correction (d'où améliorations possibles). 5) Apprentissage par génération automatique de problèmes et par contre-exemples. 6) Capable de généralisation par démonstration.
LOGIC-THEORIST	A. Newell	6, 9	1956	États-Unis	Démonstration de théorèmes de la logique des propositions.	-	-	-	-	-	-	-	1) Premier programme d'I.A.
LPS	-	(1)	1979	Japon (Univ. de Keio)	Spécialisé dans la résolution de problèmes de géométrie.	-	-	-	-	-	-	-	
MACSYMA	J. Moses	5, 10	1967	États-Unis (MIT)	Résolutions d'intégrales et équations différentielles, calcul matriciel et vectoriel, etc.	-	-	LISP	Fonctionnel	-	-	-	1) Équivalent de 50 années-personnes de recherche pour la construction du système. 2) Le système contient: - des règles de simplification ou

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
MACSYMA (suite)													d'équivalence facilement accessibles (identités); - des procédures de "reconnaissances de formes d'expressions" à un niveau sémantique et syntaxique (ex: $x^2 - 2x + 1$; forme $ax^2 + bx + c$ mais reconnoît aussi $(x - 1)(x + 1)$); - des heuristiques de résolution de problèmes et des procédures d'inférences.
MUSCADET	D. Pastre	(1)	1984	France (Univ. Paris VI) (G.R. 22)	Expérimenté en théorie des ensembles, espaces topologiques et espaces vectoriels topologiques.	50-150	-	PL/1	-	-	Capable d'établir des démonstrations de théorèmes complexes.	-	1) Les démonstrations sont rendues possibles grâce aux métaconnaissances. 2) Établit des démonstrations de théorèmes complexes. 3) Expérimenté en théorie des ensembles, des espaces topologiques et espaces vectoriels.
PM	M. Koehen, P. Resnick.	-	1986	États-Unis (Univ. Michigan)	Fait des découvertes et forme des conjectures en géométrie plane.	100	Frames	KEE (Inter-LISP D.)	Recher-che	Chaînage bidirec-tionnel	-	-	1) La prochaine version sera capable d'auto-correction. 2) L'ordre dans lequel les règles sont utilisées peut influencer le résultat. 3) Utilise des coefficients de vraisemblance pour représenter les connaissances incertaines. 4) Ne peut expliciter son raisonnement. 5) La communication avec le système ne se fait pas en langue naturelle. 6) ne peut détecter les contradictions.
PREDICTOR	A.P. White	-	-	Royaume-Uni (Univ. de Birmingham)	Probalistic induction.	-	Base de données	FORTRAN	Conçu pour fins de recher-che	-	-	-	1) Toutes les éditions du système ont été conçues pour fins de recherche.

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
PROVER	W.W. Bledsoe	(19)	1970	États-Unis	Preuve de théorèmes mathématiques (théorie des ensembles).	-	-	LISP	-	-	Un grand nb. de théorèmes ont été prouvés mais les + difficiles ne le sont pas.	-	1) Comprend trois parties: - SPLIT; - REDUCE; - RESOLUTION. 2) "Le système prouve habituellement un théorème rapidement ou pas du tout".
QA3	-	(10)	-	-	Démonstration de théorèmes.	-	-	-	-	-	-	-	
REDUCE	A. Hearn	1, 5	1969	États-Unis (Stanford)	Résolution d'intégrales, équations différentielles, etc.	-	-	-	-	-	-	-	
REF-ARF	R.E. Fikes	(11), (3)	1970	-	Résolution de problèmes mathématiques (cryptoarithmétique + problème des 8-reines).	-	-	-	-	-	-	-	
SAINT	J. Sagle	(6)	1961	États-Unis (MIT)	Intégration de fonctions élémentaires (principalement intégrales indéfinies).	-	Règles de production	LISP	Fonctionnel	-	Se comparant à un bon étudiant de niveau collégial des années 1960... Résolutions de 84 intégrales sur 86 dont 54 assez complexes.	Aucun	1) Très faible dans l'intégration définie. 2) Fonctionne sur IBM 7090. 3) Fonctions qu'il est capable d'intégrer: fonctions constantes; fonction identité; somme, produit et fonctions de puissances élémentaires; fonctions trigonométriques inverses élémentaires.

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
SEME	M. Baron	(1)	1982	France (Univ. Paris VI)	Expérimenté en manipulation de connaissances concernant les traitements formels en analyse combinatoire et dérivation.	-	-	-	-	-	-	-	
SIN	J. Moses	(6), 10	1967	États-Unis (MIT)	Intégration de fonctions.	-	Règles de production	-	-	-	Mêmes performances que celles de SAINT mais trois fois plus rapide. Peut résoudre des intégrations s'avérant fort complexes.	Aucun	
SNARK-INTEGRATION	J.L. Laurière	(5)	1982	France	Résolution d'intégrales.	-	-	-	-	-	-	SNARK	
WEST	Burton, Brown.	(15)	1982	États-Unis	Habiletés arithmétiques (diagnostic des "bugs" ou erreurs procédurales).	-	-	-	-	-	-	-	1) - Build a differential student model. - Explicate rules of when to interrupt. - Procedural representation for bugs. 2) Explique pourquoi un étudiant fait des fautes.
ZORBA	R.E. Kling	(19)	1971	États-Unis (SRI International)	Preuve de théorèmes de logique.	-	-	LISP	-	-	-	-	1) Peut acquérir de l'expérience (peut résoudre des problèmes par analogie). 2) La vitesse d'exécution est plus rapide puisqu'il peut raisonner par analogie.

MÉDECINE

Description des systèmes experts

ABEL

AI/COAG

ANTICIPATOR

APES

ARAMIS

BABY

✔ BASIC-PUFF

CADIAG 1

CASNET (GLAUCOMA)

✔ CENTAUR

CLOT

CODIAPSY

DIABETO

DIGITALIS

EMERGE

HEADMED

• HEMOCAD

HODGKINS

INTERNIST

IRIS

MÉDECINE (suite)

Description des systèmes experts

MYCIN

NEPHROS

NESTOR

NEUROLOGIST

ONCOCIN

ONYX

PATHFINDER

PIP

PROTIS

PUFF

RADEX

REINART

RHINOS

RX (RADIX)

SAM

SPHINX

TOUBIB

TROPICAID-2

VM

WHEEZE

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
ABEL	R. Patil	2, 5, 10	1982	États-Unis (MIT)	Régulation du pH.	-	-	-	Recher-che	-	-	-	1) Peut être appelé un S.E. médical de la deuxième génération.
AI/COAG	-	(4)	-	-	Hémostase.	-	Règles de production	-	-	-	-	-	
ANTICIPATOR	-	(4)	1983	-	Prescription anti-biotique.	-	- Règles de production; - Couplé avec une base de données.	PROLOG	-	-	-	-	
APES	-	(4)	1982	-	Appliqué en derma-tologie.	-	Règles de production	-	-	-	-	-	
ARAMIS	J. Fries	(5)	1972	États-Unis (Univ. Stanford)	Traitement des maladies.	-	-	-	-	-	-	-	1) Raisonnement semi-automatique par ana-logie à des cas antérieurs. 2) La possibilité de données erronées ou manquantes peut dégrader sérieusement le diagnostic. 3) Chaque fois qu'un nouveau patient est examiné, un praticien choisit les in-dices du patient qui vont servir à cher-cher, dans la base de données, un ou plusieurs cas de patients proches dont le traitement et l'évolution de la mala-die ont été suivis. C'est ainsi qu'un traitement pour le patient présent est suggéré.
BABY	-	(1)	-	-	Aide aux soins intensifs de nouveaux-nés.	-	-	-	-	-	-	ADVISE	

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
BASIC-PUFF	C. Levesque	(1)	-	France	Employé par des pneumologues pour dégrossir les interprétations spirométriques.	100	Règles de production	BASIC sur micro-ordinateur	Fonctionnel	-	85% d'interprétations correctes	EMYCIN	1) Version simplifiée de EMYCIN. 2) Employé par des pneumologues pour dégrossir les interprétations spirométriques.
CADIAG 1	-	(4)	-	-	Appliqué à la médecine interne.	-	- Règles de production; - Couplé avec une base de données.	-	-	-	-	-	
CASNET (GLAUCOMA)	C. Kulikowski, S. Weiss.	(1), 2, 4, 5, 9, 10	1978	États-Unis (Univ. Rutgers)	Application en ophtalmologie: traitement à long terme et diagnostic du glaucome.	Plus de 2 000	Réseaux sémantiques	-	-	-	-	-	1) Connu aussi sous le nom de GLAUCOMA. 2) A donné naissance au système général EXPERT.
CENTAUR	J.S. Aikins	1, 3	1983	États-Unis	Même application que PUFF (maladies pulmonaires) mais avec un meilleur contrôle de l'usage des règles.	-	- Frames; - Règles de production.	-	-	-	-	EMYCIN	1) Résulte d'une extension de EMYCIN visant à mieux contrôler l'usage des règles. 2) Offre des possibilités à l'utilisateur de structurer les connaissances (faits et règles) en conglomérats significatifs dans le domaine d'application. 3) Partition de la base des règles.
CLOT	-	(4)	-	-	Appliqué aux problèmes de coagulation.	-	-	-	-	-	-	EMYCIN	
CODIAPSY	M. Maury, A.-M. Massotte, H. Betaille, J.C. Penochet, M. Negre.	(3)	1985	France (CRIM + GRIP Hôpital Colombière)	Psychiatrie	-	Règles de production	-	-	Chaînage avant	-	-	1) Moteur d'ordre 0. 2) Reconnaissance des diverses formulations possibles d'un même terme.

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
DIABETO	J.C. Buisson, H. Farreny, H. Prade.	(1), 13	1984	France (Labo. d'Endo-crinologie et Labo L.S.I., E.N.S.E.E.I.H.I. et Univ. P. Sabatier)	Destiné à aider au diagnostic et au traitement des dia-bètes.	200	-Frames -Règle de production	-	Recher-che	-	-	-	1) Le système sera accessible par réseau vidéotex à partir d'un minitel et pourra être consulté par les généralistes depuis leurs cabinets.
DIGITALIS	G.A. Gorry, H. Silverman, S.G. Pauker.	(5)	1970	États-Unis	Utilisé en cardio-logie.	-	-	-	-	-	-	-	
EMERGE	-	(1)	1984	États-Unis (Univ. de Californie à Los Angeles)	Conçu comme un outil à consulter dans un bloc d'urgence d'hô-pital pour analyser des douleurs de poi-trines.	-	-	PASCAL	-	-	-	-	1) Fonctionne sur Apple-II.
HEADMED	-	1, (4)	-	-	Système utilisé en psycho-pharmacologie, en pratique psychia-trique.	275	Règles de production	-	-	-	-	EMYCIN	1) Moteur d'ordre 0.
HEMOCAD	-	(1)	1984	France (Hôpital d'Ivry, Faculté de Pharmacie de Chatenay Malabry)	Système simple, expert en hémostase, destiné à aider des biologistes à com-pléter leurs connais-sances ainsi qu'à entraîner des étu-diants.	-	-	-	-	-	-	-	

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
HODGKINS	C. Safrans, J. Desforges, P. Taichlis.	(7)	1976	États-Unis (MIT)	Diagnostic de la maladie de Hodgkins.	-	-	-	-	-	-	-	
INTERNIST	J. Meyers, H. Pople.	2, 4, (5), 9, 10	1982	États-Unis (Univ. Pittsburg)	Diagnostic en médecine interne (couvre 80% de la médecine interne). Permet de résoudre la plupart des cas cliniques et pathologiques.	-	Réseaux sémantiques	-	Recherche	-	Explique près de 80% des cas qui lui sont soumis.	-	1) Ne sait pas prendre une vue synthétique des problèmes, symptômes, ce qui mène parfois à des phases de consultation interminables. 2) Pas d'auto-correction (peut répéter plusieurs fois les mêmes erreurs). 3) S.E. le plus complet existant actuellement en médecine. 4) Sa connaissance renferme environ 500 maladies et 3 550 symptômes. 5) Son successeur sera CADUCEUS. 6) INTERNIST privilégie trois stratégies: - élimination; - discrimination; - poursuite.
IRIS	M. Tribogoff, C. Kulikowski.	4, 5	1977	-	Application en ophtalmologie.	-	- Règles de production; - Réseaux sémantiques; - Frames.	-	-	-	-	-	
MYCIN	E.H. Shortliffe	(1),2, 4,5,9, 10, + La Recherche No 151	1974	États-Unis (Univ. Stanford)	Système de consultation, dans le cadre de maladies infectieuses, destiné à fournir des conseils pour la prescription d'une antibiothérapie à des médecins généralistes (essaie d'identifier les	500	Règles de production	LISP	Recherche (système conçu à cette fin).	Chaînage bidirectionnel	En matière de diagnostic et de thérapie, on a jugé qu'il était au même niveau que les spécialistes humains des	TEIRESIAS: communication en langage naturel	1) Moteur d'ordre 0. 2) Ne prend pas en compte toutes les infections (ex.: causées par les champignons pathogènes ou les virus). 3) L'utilisateur ne peut donner des informations spontanées au système en cours de traitement. 4) Les connaissances incertaines peuvent être introduites avec des facteurs de certitude sur une échelle de 0 à 10,

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
MYCIN (suite)					germes responsables de l'infection du patient ainsi qu'une thérapeutique avec la dose requise).						maladies infectieuses et qu'il dépasse (quelquefois de beaucoup) les médecins non spécialistes.		selon la gravité (cet aspect peut différer d'un utilisateur à l'autre). 5) Pas d'auto-correction (peut répéter la même erreur). 6) L'ordre dans lequel les règles sont utilisées peut influencer le résultat. 7) DO à un manque de renseignements sur un certain facteur, il lui effecte un coefficient de vraisemblance de 9 (ce qui peut s'avérer faux dans certains cas). 8) A tout moment de la consultation, cependant, l'utilisateur peut interrompre le système pour poser des questions (ex: pourquoi cette question). 9) Construit un dossier, pour chaque patient, qu'il met à jour. 10) Possibilité d'écrire des règles en chaînage avant. 11) Communication en langage naturel (+ ou -), l'anglais, avec l'utilisateur (800 mots différents). 12) Composé de trois parties: - système de consultation proprement dit; - système d'explications (réponse aux questions de l'utilisateur); - système d'acquisition des règles.
NEPHROS	Asbell	(4)	1981	États-Unis (MIT)	Insuffisance rénale.	-	Réseaux sémantiques	-	-	-	-	-	
NESTOR	G.F. Cooper	(32)	1984	États-Unis	Aide au diagnostic médical.	-	-	-	-	-	-	-	1) Utilise la théorie des probabilités.

NOM	AUTEUR(S)	RÉFÉRENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉSENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTIONNEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
NEUROLOGIST	Z. Xiang, S. Srihari.	(4)	-	États-Unis (Univ. Buffalo)	Problèmes de localisation en neurologie.	-	- Règles de production; - Réseaux sémantiques.	-	-	-	-	-	
ONCOCIN	E.H. Shortliffe	2, 4, 5	1981	États-Unis	Protocoles en cancérologie.	-	- Règles de production; - Frames.	-	-	-	-	-	
ONYX	C. Langlotz, L. Fagan, S. Tu, J. Williams, B. Sikic.	(11)	1985	États-Unis (Univ. Stanford)	Construit pour la planification dans des régions d'application où les méthodes de planification traditionnelles sont difficiles à appliquer (thérapie du cancer).	-	-	-	Recherche	-	-	-	1) Procède en trois étapes: - génération de plans (envisage les possibilités de traitement vu l'état du patient); - simulation symbolique (prédiction des états futurs du patient); - classification des plans (classifie les plans d'action par ordre de priorité et indique les résultats possibles). 2) Appliqué à la base de connaissances de ONCOCIN.
PATHFINDER	E.J. Horvitz, D.E. Heckerman, B.E. Nathwani, L.M. Fagan.	(32)	1985	États-Unis (Univ. Stanford)	Reason about lymph node pathology findings noted upon microscopic examination of lymph node tissue.	-	-	-	-	-	-	-	1) Emploi des facteurs de crédibilité. 2) The system considers more than 500 high-and-low power histologic findings.
PIP	S. Pauker, A. Gorry, J. Kassirer, W.S. Schwartz.	5, 9	1976	États-Unis (MIT)	Maladies rénales.	-	-	-	-	-	-	-	

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
PROTIS	-	1, 14	1982	France (Faculté de Médecine d'Aix-Marseilles)	Appliqué au traite-ment des ictères, des hypertensions et du diabète.	-	-	-	-	-	-	-	
PUFF	J.C. Kunz	(1),2, 3,4,5, 10,11	1980	États-Unis (Univ. Stanford)	Diagnostic des mala-dies pulmonaires.	250	Règles de production	BASIC	Commer-cialisé	Chaînage avant	-	EMYCIN	1) Moteur d'ordre 0. 2) Implémenté sur Apple-II.
RADEX	-	(4)	1980	-	Radiologie.	-	Frames	-	-	-	-	-	
REINART	-	(1)	1984	France (CRIM et AIDER, Univ. de Mont-pellier).	Système simple desti-né à guider médecins et patients pour l'utilisation d'appa-reils d'hémodyalise.	-	-	PASCAL	Recher-che	-	-	HAMEX	1) Fonctionne sur IBM-PC.
RHINOS	M. Kimura	(11)	1985	Japon	Diagnostic de maux de tête et de douleurs faciales (36 mala-dies).	219	Règles de production	PROLOG-KABA	Proto-type (utilisé au dép. de neu-ro. de l'uni-versité d'Osaka)	Chaînage avant	Sur 50 pa-tients: 82% des diagnos-tics identi-ques à ceux des spécia-listes; 16% près de ceux des spécia-listes.	PROLOG	1) Le système ne peut effectuer le même raisonnement que les spécialistes (d'où les 16% d'erreurs). 2) Système portatif. 3) Le temps d'attente maximal pour obtenir un diagnostic est 20 secondes! 4) Opérationnel sur NEC PC-9801. 5) Les connaissances incertaines peuvent être insérées avec des facteurs de certitude. 6) Un tel système peut être disponible pour environ 15 000 $ US.

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
SPHINX (suite)													8) Communication en langage naturel. 9) L'utilisateur peut interrompre le système à tout moment afin de lui poser des questions. 10) MEDIUM: assure un discours cohérent. EXPERT: assure la pertinence du discours du système lors de la recherche des signes cliniques ou la demande d'examens paracliniques. 11) Peut exploiter des connaissances incertaines ou imprécises. 12) Met en oeuvre la théorie des possibilités. 13) Partition de la base des règles.
TOUBIB	-	(1)	1982	France (IBM, Paris)	Système de diagnostic en médecine d'urgence (médecine générale)	300	- Règles de production - Frames	-	Recherche (Projet bateau sans médecin)	-	-	-	
TROPICAID-2	-	(1)	1984	France (C.H.U. La Pitié-Salpétrière et Médecins sans frontières, Paris)	Système d'aide à la décision médicale pour les infirmiers des pays en voie de développement.	-	-	-	-	-	-	EMYCIN	1) Comporte une base de données pharmacologiques concernant 200 médicaments, une base de données thérapeutiques concernant 500 maladies et une base de données diagnostiques se référant à 2 000 signes cliniques ou paracliniques. 2) Offre la possibilité de structurer les connaissances (règles, faits) en conglomérats significatifs dans le domaine d'application. 3) Peut exploiter les connaissances incertaines.

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
VM	-	(1), 2, 4	1979	-	Interprétation des données quantitatives relevées dans le cadre d'une unité de soins intensifs. Ces données sont utilisées pour piloter un respirateur artificiel. (Assistant à la réanimation).	-	Règles de production	-	-	Chaînage avant	-	-	1) Moteur d'ordre 0. 2) Pas d'auto-correction. 3) Effectue un suivi des patients, depuis leur admission jusqu'à leur sortie du service de soins intensifs. 4) Quatre tâches principales: - détecter les principales erreurs de mesure; - déclencher des alarmes sur détection d'événements particuliers concernant soit le malade, soit la machine elle-même; - résumer l'état du patient; - suggérer des ajustements thérapeutiques.
WHEEZE	Smith et Clayton	(3)	1977	États-Unis	Même application que PUFF (maladies pulmonaires) mais avec un contrôle de l'usage des règles qui s'avère meilleur (même que celui de CENTAUR).	-	- Frames; - Règles de production.	-	-	-	-	EMYCIN	
-	S. Kaihara et al.	(4)	1978	-	Maladies cardiovasculaires.	-	Règles de production	-	-	-	-	-	
-	B.M. Masand	(4)	1982	États-Unis	Protocoles en cancérologie.	-	Réseaux sémantiques	-	-	-	-	-	
-	S.M. Weiss, C. Kulikowski, R. Galen.	(31)	1981	-	Analyse de "serum protein".	-	-	-	-	-	-	-	

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
-	K.A. Korn, P. Compton, L. Lazarus, J.R. Quinlan.	(31)	1985	-	Interpret thyroid hormone assays.	-	-	-	-	-	-	-	
-	I. Futo, P. Szeredi, F. Darvas.	(4)	1978	-	Interactions médi-caments.	-	- Règles de production; - Couplé avec une base de données.	PROLOG	-	-	-	-	

MILITAIRE

Description des systèmes experts

AALPS

FAITH

HASP/SIAP

TATR

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
AALPS	C. Rich	(31)	1985	États-Unis (MIT)	Planification de chargement optimal d'équipement et de cargaisons sur les avions.	-	-	-	-	-	-	-	
FAITH	L. Friedman	(11)	1984	États-Unis	-	-	Logique des prédicats	FCL	Recher-che	Chaînage schemixte	A produit les dia-gnostics désirés; a obtenu les mêmes réaul-tats que les experts.	-	1) Moteur d'ordre 1. 2) Peut extraire et ajouter des règles, axiomes. 3) FCL est un langage utilisable par des non-experts en informatique (program-mation).
HASP/SIAP	H.P. Nii, E.A. Feigenbaum, J.J. Anthon, A.J. Rockmore.	2, 5	1982	États-Unis	Interprétation de signaux et liaison avec la défense.	-	-	-	-	-	Lors de tests pratiqués par les savants de la défense, le système avait des performances égales ou supérieures aux ré-sultats humains.	-	1) Capable de raisonner. 2) Sonar passif conçu pour interpréter les bruits de l'océan dans un environnement très bruyant.
IATR	-	(2)	-	États-Unis (Rand Corp.) (U.S. Air Force)	Établissement des trajectoires.	-	-	-	-	-	-	ROSIE	

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
-	-	(2)	-	- (ESL Inc. et Teknow-ledge Inc.)	Destiné à l'analyse des communications tactiques sur le champ de bataille.	-	-	-	Proto-type	-	-	-	
-	-	(2)	-	- (ESL Inc. et Teknow-ledge Inc.)	Destiné à l'analyse des indicateurs et des avertissements stratégiques.	-	-	-	Proto-type	-	-	-	

MULTI-EXPERTS

Description des systèmes experts

SATIN

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
SATIN	-	(1)	-	-	Faire coopérer des expertises juridique, économique, technologique et documentaire en vue de réaliser des transferts de technologies depuis les laboratoires vers les utilisateurs.	-	-	-	Recher-che	-	-	MIME	

MYCOLOGIE

Description des systèmes experts

MYCOLOG

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
MYCOLOG	-	(1)	1984	France	Mycologie.	-	-	LISP	-	-	-	MICROMYC-2	1) Fonctionne sur TRS-80. 2) Ne peut répondre à des questions sur son raisonnement. 3) Peut utiliser des coefficients d'atténuation et des facteurs de certitude.

PHYSIQUE

Description des systèmes experts

BACON-3

EL

MECHO

PEACE (1 et 2)

PROMPT

QPT

SOPHIE

SYGAL

NOM	AUTEUR(S)	RÉFÉRENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉSENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTIONNEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
BACON-3	P. Langley	(5)	1979	-	Découverte de lois en physique.	-	-	-	-	-	-	-	1) Redécouvre des lois empiriques, notamment certaines lois de physique (loi des gaz parfaits, ...). 2) Applique des heuristiques détectant des régularités ou des tendances dans les données, ce qui le conduit à formuler des hypothèses qu'il essaie ensuite de vérifier.
EL	A. Brown, G.J. Sussman.	(1),5	1974	États-Unis (MIT)	Électronique, analyse de circuits électriques.	-	-	-	-	-	-	ARS	
MECHO	A. Bundy et al.	9, 10	1979	Royaume-Uni (Univ. d'Edinburgh)	Résolution de problèmes de mécanique.	-	-	-	-	-	-	-	
PEACE (1 et 2)	H. Dincbas	(1), 5	1980	France (D.N.E.R.A. - C.E.R.T.) Toulouse	Système pour l'analyse et la synthèse de circuits électroniques passifs.	-	-	METALOG	-	-	-	METALOG	
PROMPT	S. Addanaki, E. Davis.	(11)	1985	États-Unis	Essaie de résoudre certains problèmes physiques. Exemples: - Concevoir un stylo qui peut écrire la tête en bas; - Pourquoi un stylo n'écrit pas sur du verre; - Qu'arriverait-il à un stylo à 200°F.	-	-	-	Recherche	-	-	-	1) Raisonne au même niveau que les experts.

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
QPT	K.D. Forbus	(11)	1984	-	Essaie de résoudre certains problèmes physiques.	-	-	-	Recher-che	-	-	-	1) Raisonnement naïf.
SOPHIE	J.S. Brown, R. Burton.	(1), 5 10, 15	1975	États-Unis (Univ. Stanford)	Système d'aide à l'enseignement par ordinateur, de la détection des pannes dans un circuit électrique.	-	-	-	-	-	-	-	
SYGAL	-	(1)	-	-	Classification des galaxies.	-	-	-	-	-	-	CRIQUET	
-	G. Novak	-	1977	États-Unis (MIT)	Mécanique.	-	-	-	-	-	-	-	

PRODUCTION MANUFACTURIÈRE

Description des systèmes experts

GUMMEX

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
GUMMEX	-	(1)	1984	R.F.A. (Batelle-Institut)	Engendre des gammes de fabrication de produits en caout-chouc.	300	-	PROLOG	-	-	-	PROLOG	

PROGRAMMATION AUTOMATIQUE

Description des systèmes experts

DEDALUS

EXPERT MACHINIST

MAIA

M3L

PECOS

PROGRAMMER'S APPRENTICE

PSI

SAFE

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
DEDALUS	Z. Manna, R. Waldinger.	(5)	1975	-	Synthèse de programmes.	-	-	-	-	-	-	-	
EXPERT MACHINIST	-	(30)	-	États-Unis (CMU/CMI)	Programmation automatique.	-	-	-	Recher-che	-	-	-	
MAIA	J.P. Sansonnet	9, (13)	1984	France (CNET, CGE)	Synthèse de programmes (machine-langage).	-	-	LISP PROLOG	Recher-che	-	-	PROLOG	
MSL	J.P. Sansonnet	9, 13	1977	France (Univ. Toulouse)	Synthèse de programmes.	-	-	-	Conçu pour fins de recher-che	-	-	-	1) Prédécesseur de MAIA.
PECOS	D. Barstow	(5)	1979	États-Unis (New York)	Synthèse de programmes.	-	-	-	-	-	-	-	
PROGRAMMER'S APPRENTICE	C. Rich, H.E. Shrobe.	(2), 5, (C)	1976	États-Unis (MIT)	Synthèse de programmes.	-	-	-	-	-	-	-	
PSI	C. Green	2, 5, 9	1976	États-Unis (Kestrel Institute)	Synthèse de programmes.	-	-	-	-	-	-	-	
SAFE	R.M. Balzer, N. Goldman, D. While.	(5)	1977	-	Synthèse de programmes.	-	-	-	-	-	-	-	

RAISONNEMENT

Description des systèmes experts

ORDF

SNIFFER

XPLAIN

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
ORDF	-	(1)	-	-	Problèmes d'ordonnan-cement.	-	-	-	-	-	-	EAQUE	
SNIFFER	R.E. Fikes, G. Hendrix.	(6)	1977	États-Unis	Capable de faire des déductions.	-	-	-	-	-	-	-	
XPLAIN	W.R. Swartout	(25), 11	1983	États-Unis (MIT)	Système conçu pour expliquer des raisonnements en profondeur.	-	-	-	-	-	-	-	

RECONNAISSANCE DE LA PAROLE

Description des systèmes experts

MOZART

SERAC

SONEX

V. ZUE

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
MOZART	-	(9)	-	France (L.I.M.S.I.)	Reconnaissance de la parole.	-	-	-	-	-	-	-	1) La méthode employée permet de reconnaître un mot avant que celui-ci ait été complètement prononcé.
SERAC	M. Gilloux, G. Mercier, C. Tarridec.	(1), 9	1984	(CNET, Lannion)	Interprétation de sonagrammes pour la reconnaissance de la parole en environnement normal.	-	-	-	-	-	-	-	1) Lorsqu'on écrit "environnement normal" on englobe: vocabulaires importants, locuteurs variables, parole continue.
SONEX	D. Memmi, M. Eskenazi, J. Mariani.	(1)	1984	France (L.I.M.S.I., Orsay)	Développé pour interpréter des sonagrammes à partir des règles fournies par des experts en phonétique.	-	-	-	-	-	-	-	
V. ZUE	-	(9)	-	États-Unis (MIT)	Interprétation de sonagrammes.	-	-	-	-	-	Moins de 70% d'erreur, quel que soit le locuteur.	-	

RÉPARATION

Description des systèmes experts

CAMA

CATS-1

DELTA

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
CAMA	-	(1)	1982	Italie	Orienté vers le dépannage automatique de mécanismes complexes (pannes d'automobiles).	100	-	-	-	Chaînage arrière	-	-	1) CAMA admet que l'état du mécanisme observé évolue en cours de dépannage, contrairement à d'autres systèmes de diagnostic qui étudient un état préexistant à la session de consultation, sans le modifier. 2) Peut fonctionner de manière non monotone.
CATS-1	-	9, 12	-	-	Le plus connu des systèmes de diagnostic de pannes (locomotives).	1 200	Règles de production	-	-	-	-	-	1) Lorsqu'il est utilisé avec un vidéodisque, il peut montrer des diagrammes et des films d'animation se rapportant à la consultation. 2) L'expertise qu'il contient a été collectée chez General Electric auprès de techniciens qui réparaient, pour certains depuis plusieurs décennies, les locomotives fabriquées par cette société.
DELTA	P.P. Bonissone	1,(15)	1982	États-Unis (General Electric)	Aide au diagnostic de pannes de moteurs.	140	-	-	-	-	-	-	-
-	-	(9)	-	France (Cap-Sogeti Innovation, Grenoble)	Système en réparation automobile réalisé pour la régie Renault.	-	-	-	-	-	-	-	

RISQUES

Description des systèmes experts

MEPRA

PRAS

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
MLPRA	J.M. Lefebvre	(1)	1984	France (L.I.F. I.A., E.N. S.I.M.A.G.) Grenoble	Destiné à évaluer des risques d'avalanches.	-	-	-	Recher-che	-	-	-	
PRAS	-	(2)	-	Japon (Hitachi, Tokyo)	Système permettant d'évaluer les risques dans un projet.	-	-	-	Recher-che	-	-	-	

ROBOTIQUE

Description des systèmes experts

ABSTRIPS

BUILD

SIPE

SPAN

STRIPS

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
ABSTRIPS	E. Sacerdoti	(6)	1974	-	Module de décision pour un robot.	-	-	-	-	-	-	-	1) Partant d'un plan, le système doit planifier une démarche à suivre pour atteindre un certain but. 2) Extension de STRIPS. 3) Utilise des "valeurs critiques" afin de définir le monde dans lequel il doit évoluer. 4) Si deux plans semblent identiques ABSTRIPS continue de les comparer afin de choisir le plus efficace au lieu de choisir arbitrairement comme le fait STRIPS. 5) Alors que STRIPS a pris 30 minutes pour accomplir une scène ABSTRIPS a pris 5:28 minutes.
BUILD	S.E. Fahlman	10, (17)	1974	États-Unis (Cambridge)	Système de planifica-tion pour les tâches de construction des robots.	-	-	CON-NIVER	Recher-che	-	Modestes	-	1) Pas d'apprentissage. 2) Limité (primitif). 3) Système qui permettrait à un robot d'établir une stratégie lors de la cons-truction de certaines formes. 4) La position des objets est donnée par un humain (et non par un système de vision). 5) Fait comme si les formes pouvaient être déplacées par magie. 6) Divisé en deux parties majeures: - système de planification; - système de modélisation. 7) Inintelligent.

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
SIPE	D.E. Wilkins	(26)	1984	États-Unis	Module de décision (génère un plan).	-	-	LISP	Recher-che	-	-	-	1) En partant d'un plan, le système planifie une démarche à suivre pour atteindre un certain but. 2) Il est fréquent que le système n'atteigne pas le but fixé. 3) Imitation de STRIPS.
SPAN	D.L.S. Berlin	(12)	1985	États-Unis (Univ. Stanford)	Module de décision.	-	-	-	-	-	-	-	
STRIPS	R. Fikes, N. Nilsson.	6, 9, 19, (20), 21	1971	États-Unis (SRI Inter-national)	Module de décision permettant à un robot de se déplacer (gé-nère un plan).	-	-	-	Recher-che	-	-	-	1) Destiné au robot HILARE. 2) Le temps d'exécution pour générer un plan peut être diminué grâce à un MACROP (réduit de 2/3 dans certains cas). 3) Partant d'un plan, le système doit pla-nifier une démarche à suivre pour atteindre un certain but. 4) En profondeur d'abord. 5) Si deux plans semblent identiques, un choix arbitraire est fait.

SYSTÈMES EXPERTS (information incomplète)

Description des systèmes experts

ARGUS

BLAST

CAPS

CHAT-80

CORA

DYNPAT

EPAM

EXTASE

FADES

GREASE

ILOG

IMACS

INET

NUDGE

PARSIFAL

PDS

PERT

POLITICS

SYSTÈMES EXPERTS (information incomplète) (suite)

Description des systèmes experts

RACE

ROBOT/INTELLECT

ROME

SCD

SERF

SESP

SOAR

SOJA

TRANSCELL

WASTE

WRIGHT

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	APPLICATIONS PROBABLES	AUTRES
ARGUS	W.R. Reitman	10	1965	-	-
BLAST	-	(30)	-	Analyse de signaux.	États-Unis (CGI/DEC). Fonctionnel.
CAPS	H.J. Berliner	6	1974	Échecs.	-
CHAT-80	D.H.D. Warren, F.C.N. Pereira.	10	1982	-	-
CORA	-	(30)	-	Relay protection analysis.	États-Unis (Westinghouse).
DYNPAT	J.A. Campbell, F. Gardin.	10	1982	-	-
EPAM	E.A. Feigenbaum	1, 5	1964	-	Moteur PS. Règles de production.
EXTASE	F. Jakob, D. Vernet, M. Lepetit.	(13)	1986	Surveillance de processus industriels.	France (CGE).
FADES	-	(30)	-	Equipment selection.	États-Unis (Purdue). Recherche.
GREASE	-	(30)	-	Cutting fluid selection.	États-Unis (CMU/Gulf). Recherche.
ILOG	-	(30)	-	Sourcing and transportation.	États-Unis (CMU/DEC).
IMACS	-	(30)	-	Flow shop management.	États-Unis (CMU/DEC). Fonctionnel.
INET	-	(30)	-	Knowledge-based simulation.	États-Unis (CMU/DEC). Fonctionnel.
NUDGE	I.P. Goldstein, R.B. Robert.	3	1977	-	-

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	APPLICATIONS PROBABLES	AUTRES
PARSIFAL	M.P. Marcus	6	1980	-	-
PDS	-	(30)	-	Process diagnosis.	États-Unis (Westinghouse). Production.
PERT	-	2	-	Analyse de risques.	-
POLITICS	J.G. Carbonell	3	1979	-	-
RACE	-	(30)	-	Preventative maintenance planning.	États-Unis (Honeywell). Recherche.
ROBOT/ INTELLECT	L. Harris	5, 16	1977	-	-
ROME	-	(30)	-	Long range planning.	États-Unis (CMU/DEC).
SCD	-	(30)	-	Cell control.	États-Unis (Hitachi). Production.
SERF	F. Ingrand	(13)	1985	Permet d'induire la forme géométrique d'un objet à partir de ses spécifications fonctionnelles.	France (IMAG). Recherche.
SESP	-	(13)	-	Embauche de personnel.	France (Data Base Informatica). Détermine votre profil psychologique qu'il compare avec le portrait-robot de l'employé modèle.
SOAR	Groupe HELIX	(32)	1985	System capable of general intelligent behavior.	États-Unis (Univ. Stanford). Recherche. 1) A déjà démontré son habileté à résoudre des problèmes de tic-tac-toe, etc. 2) A été utilisé avec le système-expert R1 et les résultats obtenus sont excellents.

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	APPLICATIONS PROBABLES	AUTRES
SOJA	B. Sauvé	(13)	1985	Ordonnancement d'atelier.	France (CGE). Sélectionne et ordonne des phases en tenant compte des contraintes qui interviennent sur les outils disponibles pour chaque machine, sur la durée d'usinage, etc.
TRANSCELL	-	(30)	-	Cell control.	États-Unis (CMU). Fonctionnel.
WASTE	-	(30)	-	Process diagnosis and simulation.	États-Unis (CMU). Recherche.
WRIGHT	-	(30)	-	Facility layout.	États-Unis (CMU).
-	J. Gros, C. Bernard.	(13)	-	Licenciement de personnel.	France (Univ. Montpellier).
-	D. Michie	(31)	-	Nuclear fuel enhancement.	Écosse (Turing Institute).
-	-	(30)	-	PCB fault diagnosis.	États-Unis (ITT). Fonctionnel.

SYSTÈMES GÉNÉRAUX

Description des systèmes experts

ADVISE (de PLANT/da, PLANT/dc, BABY)

ADVISOR

AGE

AIMDS

ALOUETTE

ALX

APES II

ARGOS II

ARS

ART

CRIQUET (de SYGAL)

EAQUE

EMYCIN (de MYCIN)

ES/P ADVISOR

EXPEROPS5

EXPERT 1

EXPERT (de CASNET)

EXPERT SYSTEM

EXSYS

FLAVORS

SYSTÈMES GÉNÉRAUX (suite)

Description des syetèmes experts

FOL

FORMES

FRL

FUZZY

GENLOG

GOLEM

GOLUX

GOSSEYN (dans COPEST)

HAMEX

HPRL

ID3

INDUCE 1, 2

INSIGHT 1

IPS

IROISE

KAS (de PROSPECTOR)

KEE

KEPE

KES dans (REACTOR)

KL-ONE

SYSTÈMES GÉNÉRAUX (suite)

Description des systèmes experts

KNOWLEDGE CRAFT

KOOL

KRL

KS-300 dans (DRILLING ADVISOR, WAVES)

LOOPS (de KBVLSI)

M1 (de EMYCIN)

MacEXPERT

MERING

METABOL

METALOG

METHODS

MICRO-EXPERT

MICROMYC 1, 2

MICROPLANNER

MICROSYMBOL (MICROMOTOR)

MIME dans (SATIN)

MIRLITHO dans (LITHO)

MP-LRO

MRS

NOAH

SYSTÈMES GÉNÉRAUX (suite)

Description des systèmes experts

OPS (OPS 5+)

ORBIT

OURCIN

OWL

PARSEC

PAS-II

PERSONAL CONSULTANT

PILOTEX

PLANNER

PROLOG

PSC

R.L.L.

RAINBOW

ROSIE dans (TATR)

RULE MASTER

SI

SACSO

SAGE

SERIES PC

SMALLTALK

SYSTÈMES GÉNÉRAUX (suite)

Description des systèmes experts

SMP

SNARK dans (SIMMIAS)

SNePS

SPII-1

SPIRAL

SPROUTER

STROBE

SYPRUC

TANGO

TEIRESIAS

TEK 440A

THOTH

TIGRE-1

TIMM

TOPSI

TROPIC

UNITS

XSYS

ZERO+

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
ADVISE (de PLANT/ds, PLANT/dc, BABY)	R.S. Michalski, A.S. Baskin.	(1)	1983	États-Unis (Univ. Illinois)	Système général. -Médecine -Agriculture	-	-	PASCAL	-	-	-	-	1) Extrait du système-expert PLANT/ds. 2) Utilise des primitives de création dynamique des règles. 3) Utilise le chaînage avant.
ADVISOR	-	(12)	-	États-Unis	Système général.	255	-	ASSEMB-LER	Fonc-tionnel	-	-	-	1) Fonctionne sur Commodore 64, Apple II et Atari 800. 2) 95 $ US. 3) Disponible chez: Ultimate Media Inc., 275 Magnolia Ave., Larkspur, CA 94939, USA, Tél.: (415) 924-3644.
AGE	-	(2)	-	États-Unis (Univ. Stanford)	Système général guidant le développement de systèmes experts.	-	-	-	-	-	-	-	
AIMDS	N.S. Sridharan	(5)	1980	États-Unis (Univ. Rutgers)	Système général.	-	Réseaux sémantiques	-	-	-	-	-	
ALOUETTE	D. Mulet-Marquis, M. Gondran.	1, 9	1984	France (E.D.F. Clamart)	Système général.	-	-	PASCAL	-	-	-	-	1) Dispose de primitives qui agissent directement sur le moteur d'inférence. 2) Le modèle de compatibilité est la semi-unification. 3) Les règles seulement comportent des variables. 4) Adopte une stratégie par tentative. 5) Peut fonctionner de façon non monotone. 6) Structuration de la base de connaissance (partition de la base de faits). 7) Systèmes-experts = Superikon, Simmins.

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
ALX	-	(2)	-	Grande-Bretagne (Intel-ligent Terminals Ltd.)	Aide les experts en diagnostic à coder leur connaissance d'un domaine scientifique, générant ainsi un système capable d'employer cette connaissance à leur place.	-	-	PASCAL	Fonc-tionnel sur Apple II	Chaînage avant	-	-	1) Approche par la théorie des probabilités (approche Boysienne) pour le contrôle du déclenchement des règles.
APES II	Logic Based Systems Ltd.	-	1986	Angleterre (Londres)	Système général.	Varia-ble	Règles de production	LPA PROLOG	Fonc-tionnel	Chaînage arrière	-	-	1) La première version, APES I, a été présentée en 1984. 2) L'ordre dans lequel le système utilise(ra) les règles peut influencer le résultat. 3) Le système peut répondre à des questions sur son raisonnement. 4) Le système peut détecter les contradictions. 5) Les applications sont très variées. 6) Fonctionne sur IBM-PC, IBM-XT, IBM-AT, Sirius 1, Victor 9000, Apricot, DEC Rainbow, Wang PC, RM Nimbus, IBM-PC compatibles.
ARGOS II	H. Farreny, M. Cayrol, B. Fade.	(1), 5	1980	France (L.S.I., E.N.S.E.E.I.H.T.) + (Univ. P. Sabatier)	Système général orienté vers la construction de robots.	-	Règles de production	LISP	-	-	-	-	1) Ordre 1. 2) Distingue des catégories de faits: entre faits établis, faits à établir et des plans. 3) Le modèle de compatibilité est la semi-unification. 4) Offre une gamme d'opérateurs de filtrage, de fonctions de filtrage et de règles spécifiques au filtrage (théorèmes).

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
ARGOS II (suite)													5) Utilise des primitives qui peuvent modifier la base des connaissances. 6) Dispose de primitives qui agissent directement sur le moteur d'inférence. 7) Utilise des métarègles. 8) Les corps des métarègles peuvent faire appel à des primitives d'inhibition et validation de règles. 9) L'inhibition ou la validation peut éventuellement être armée lors du déclenchement de la métarègle sans exécution immédiate. 10) Permet l'introduction de procédures dans les règles ou métarègles. 11) Permet l'utilisation de démons. 12) Les règles seulement comportent des variables. 13) Le moteur fonctionne en chaînage mixte. 14) Développement en profondeur d'abord. 15) Adopte une stratégie par tentative. 16) Peut fonctionner de manière non monotone. 17) Orienté vers le contrôle de robots de troisième génération. 18) Un module pour le suivi de l'exécution des plans.
ARS	R.M. Stallman, G.J. Stussman.	(1)	1979	États-Unis (MIT)	Système général.	-	Logique des prédicats.	-	-	-	-	-	1) Utilise le chaînage avant. 2) Analyse de circuits électriques. 3) Démonstration de théorèmes en géométrie plane. 4) Utilisé pour modéliser la gestion d'une entreprise agricole.

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
ART	-	(9)	-	États-Unis (Inference Corp.)	Système général.	-	Règles Si-alors.	LISP	-	-	-	-	1) Les différentes pièces du système peuvent être achetées séparément. 2) Modes d'inférence par chaînages avant et arrière = chaînage bidirectionnel. 3) Peut représenter des connaissances incertaines. 4) Interface graphique - Fenêtre. 5) Peut être implanté sur: - LISP, - Zeto LISP, - VAX LISP, - LMI Machines, - DEC-VAX.
CRIQUET (de SYGAL)	P. Vignard	1	1984	France (INRIA, Sophia-Antipolis)	Système général. Classification des galaxies.	-	-	-	-	Chaînage bidirec-tionnel	-	-	1) Peut exploiter les connaissances incertaines. 2) Utilise une fonction heuristique intrinsèque au moteur.
EAQUE	C. Roche	(1)	1984	France (Univ. de Savoie, Chambéry)	Système général. Problèmes d'ordon-nancement.	-	-	-	-	-	-	-	1) Un moteur d'inférence paramétré, c'est-à-dire un moteur dans lequel on peut faire varier, selon la classe d'appli-cation considérée, la stratégie de choix de règles à appliquer.
EMYCIN (de MYCIN)	W. Van Melle	1, 2, (5)	1974	États-Unis (SRI)	Système général. -Diagnostic -Contrôle de processus -Agriculture -Conception mécanique	-	Règles de production	LISP	-	Chaînage bidirec-tionnel	-	-	1) Ordre 0. 2) Détecte les contradictions. 3) Évalue un degré de compatibilité entre déclencheurs et règles. 4) Utilise des métarègles. 5) On offre des possibilités de structu-ration des faits et des règles en conglomérats significatifs dans le domaine d'application. 6) Le modèle de compatibilité est la semi-unification.

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
EMYCIN (de MYCIN) (suite)													7) Les règles seulement comportent des variables. 8) Peut exploiter des connaissances incertaines. 9) Le chaînage arrière est combiné avec un développement en profondeur. 10) Adopte une stratégie irrévocable. 11) Incorpore un grand nombre d'utilitaires. Ex.: ARL = compilateur de règles. Teiresias = un système-expert dans l'écriture des systèmes-experts.
ES/P ADVISOR	-	(12)	1984	États-Unis (Expert Systems Intern-ational)	Système général. -Diagnostic -Prescription	400	Règle Si-alors	PROLOG	Fonc-tionnel	Chaînage arrière. Profon-deur d'abord.	-	-	1) Fonctionne sur IBM-PC. 2) 1 895 $ US. 3) Disponible chez: Expert Systems Int'l, 1150 First Ave., King of Prussia, PA 19406, USA, Tél.: (215) 337-2300. 4) Le modèle de compatibilité est le principe de résolution. 5) Retraçage des inférences. 6) Pour des petits systèmes-experts. 7) Peut être implanté sur IBM-PC.
EXPEROPS5	-	(12)	-	États-Unis	Système général.	-	-	EXPER-LISP	Fonc-tionnel	-	-	-	1) Fonctionne sur Macintosh (Apple). 2) 195 $ US. 3) Disponible chez: Expertelligence Inc., 559 San Ysidro Road, Santa Barbara, CA 93108, USA, Tél.: (805) 969-7874.
EXPERT 1	-	(9)	-	- (Mind Soft)	Système général.	-	-	-	-	-	-	-	

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
EXPERT (de CASNET)	S. Weiss, C. Kulikowski.	2, 3, (5)	1979	États-Unis (Univ. Rutgers)	Système général. -Médecine	-	-	FORTRAN	-	Chaînage avant	-	-	1) Les règles sont classées en trois types: - les associations entre manifestations (symptômes et signes); - les associations entre symptômes et hypothèses; - les associations entre hypothèses. 2) Utilise de nombreuses abréviations qui diminuent fortement sa lisibilité. 3) Utilise des coefficients de vraisemblance. 4) Dégagé à partir de CASNET/GLAUCOMA. 5) Appliqué aux affections de la thyroïde en rhumatologie et en neuro-ophtamologie. 6) Peut représenter des connaissances incertaines. 7) Règle Si-alors. 8) Interactif question-réponse. 9) Analyse statistique de sa propre performance. 10) Peut être implanté sur toute une gamme d'ordinateurs (mais par sur ordinateur personnel).
EXPERT SYSTEM	-	(12)	-	États-Unis	Système général.	5 000	-	BASIC	Fonc-tionnel	-	-	-	1) Fonctionne sur IBM-PC. 2) 20 $ US. 3) Disponible chez: PPE Inc., P.O. Box 2027, Gathersburg, MD 20879, USA, Tél.: (301) 997-1489.
EXSYS	-	(12)	-	États-Unis	Système général.	5 000	-	C	Fonc-tionnel	-	-	-	1) Fonctionne sur IBM-PC. 2) 295 $ US. 3) Disponible chez: EXSYS Inc., P.O. Box 75158, Albuquerque, NM 87194, USA, Tél.: (505) 836-6676.

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
FLAVORS	D.A. Moon, D. Weinreb.	(5)	1980	États-Unis (MIT)	Système général.	-	Frames	LISP	-	-	-	-	
FOL	R.W. Weybrauch	5, I	1980	États-Unis (Stanford)	Système général.	-	Logique des prédicats	-	-	-	-	-	
FORMES	P. Cointe, X. Rodet.	(5)	1983	-	Système général.	-	Frames	-	-	-	-	-	
FRL	R.B. Roberts, I.P. Goldstein.	5, 15	1977	États-Unis (MIT)	Système général.	-	Frames	-	-	-	-	-	
FUZZY	Le Faivre	-	1977	France	-	-	-	-	-	-	-	-	1) Motivé par la théorie des ensembles flous. 2) Généralisation de la théorie des ensembles Booleen. 3) Alloue un degré d'appartenance d'un individu à l'ensemble.
GENLOG	-	(1)	-	-	Micro-système général.	-	-	LISP	-	Chaînage arrière	-	-	1) Ordre 0.
GOLEM	-	(9)	-	France (Micro-formatic)	-	-	-	-	-	-	-	-	
GOLUX	Hayes	(I)	-	États-Unis	Système général.	-	-	-	-	-	-	-	1) Usage limité.
GOSSEYN dans (COPEST)	J.M. Fouet	(1)	1983	France (E.N.S. E.I., Cahan)	Système général. -Conception de pièces estampées	-	-	FORTRAN	-	-	-	-	1) Le but premier était la conception d'un système général (indépendant de l'application).

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
HAMEX	Betaille	1, 3	1984	France	Système général. -Médecine	-	-	PASCAL	-	-	-	-	1) Sur IBM-PC. 2) Utilisé pour le développement d'un système destiné à guider patients et médecins pour l'utilisation d'appareils d'hémodialise.
HPRL	S. Rosenberg	(5)	1983	États-Unis (Hewlett Packard)	Système général.	-	Réseaux sémantiques ou Frames	-	-	-	-	-	
ID3	Quinlan	(9)	-	-	Système général.	-	-	-	-	-	-	-	
INDUCE 1, 2	R.S. Michalski, T.G. Diettrich.	5, 9, L	1981	États-Unis	Système général servant à l'apprentissage.	-	-	-	-	-	-	-	
INSIGHT 1	-	(12)	1984	États-Unis (Knowledge System)	Système général. -Diagnostic -Prescription	2 000	Règle Si-alors	TURBO PASCAL	Fonctionnel	Chaînage bidirectionnel	-	-	1) Fonctionne sur IBM-PC. 2) 95 $ US. 3) Disponible chez: Level 5 Research Inc., 4980 S-A1A, Melbourne Beach, FL 32951, USA, Tél.: (305) 729-9046. 4) Disponible aussi: INSIGHT 2. 5) Peut représenter des connaissances incertaines. 6) Retraçage des inférences. 7) Peut être implanté sur: - DEC-Rainbow, - Victor 9000. 8) Interface avec le logiciel D BASE II.
IPS	M.D. Rychener, A. Newell.	(1)	1978	États-Unis	Système général dérivé d'OPS.	-	-	-	-	Chaînage mixte	-	-	

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
IROISE	-	(9)	-	France (CNET)	Système général.	-	-	-	-	-	-	-	
KAS (de PROSPECTOR)	R. Reboh	2, 3, 5	1981	États-Unis (SRI Inter-national)	Système général. -Analyse géologique -Recherche pétrolière	-	Règles de production	LISP	-	-	-	-	1) Fonctionne à partir d'un réseau d'in-férence. 2) La base de connaissance est constituée d'un réseau sémantique qui comprend à la fois les faits qui sont connus par le système (pondérés par un facteur de probabilité) et des arcs d'inférence qui spécifient comment la probabilité d'une insertion affecte celle d'une autre insertion. 3) De plus, il est possible de considérer des antécédents d'une règle comme étant des contextes, c'est-à-dire des états qui doivent être précisément établis pour que la règle se déclenche. 4) Offre un grand nombre de facilités et d'utilitaires pour: - manipuler une base des données; - interagir avec l'utilisateur (langage quasi-naturel); - expliquer son raisonnement.
KEE	T.P. Kehler, C.D. Clemenson.	(5), 13	1984	États-Unis (Intelli Corp.)	Système général.	-	Frames + règles de production	LISP	-	-	-	-	1) Utilisé par la NASA et le département de la défense. 2) Petite merveille. 3) Mode d'inférence défini par l'utilisa-teur. 4) Mode interactif, image graphique de la base de connaissance. 5) Valeurs repères pour le contrôle du comportement du système.

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
KEE (suite)													6) Peut être implanté sur: - LISP Machine, - Zeta/Common LISP, - Xerox 1100S, - Symbolics 3600, - LMI LAMBDA, - TI Explorer.
KEPE	-	(2)	-	États-Unis (Intelli-Genetics Inc.) Palo Alto	Système général.	-	-	-	Fonc-tionnel	-	-	-	
KES dans (REACTOR)	-	(1), 12	1984	États-Unis (Software Architec-ture and Engineer-ing)	Système général orienté vers le dia-gnostic interactif. -Traitement d'acci-dents dans des réacteurs nucléaires	-	Règles Si-alors	IQ LISP	Fonc-tionnel	-	-	-	1) Pourvu d'un environnement évolué pour l'acquisition des connaissances. 2) Fonctionne sur IBM-PC. 3) 4 000 $ US. 4) Disponible chez: Software A and E Inc., 1500 Wilson Blvd., Arlington, VA 22209, USA, Tél.: (703) 276-7910. 5) Contrôle procédural. 6) Utilise la théorie des probabilités pour le choix des règles à déclencher. 7) Retraçage des inférences. 8) Interface possible avec d'autres bases de données. 9) Peut être implanté sur: - A-LISP, - Wisconsin LISP, - Franz LISP, - DEC VAX/VMS, - VAX/UNIX, - CDC CYBER - APOLLO,

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
KES dans (REACTOR) (suite)													- Xerox 1100, - Symbolics 3600, - IMB PC.
KL-ONE	R.J. Brachman	(5), 15	1977	États-Unis (BBN)	Système général.	-	Réseaux sémantiques + Frames	-	-	-	-	-	
KNOWLEDGE CRAFT	-	(9)	-	- (TECSI)	Système général.	-	-	-	-	-	-	-	
KOOL	-	(9)	-	- (BULL)	Système général.	-	-	-	-	-	-	-	
KRL	O. Bobrow, T. Winograd.	(5), 15, B	1977	États-Unis (Xerox Parc)	Système général.	-	Frames	LISP	-	-	-	-	
KS-300 dans (DRILL-ING ADVISOR, WAVES)	-	1, 2	-	États-Unis (Teknow-ledge Inc.) Palo Alto	Système général.	-	-	-	Fonc-tionnel	-	-	EMYCIN	
LOOPS (de KBVLSI)	D.G. Bobrow, M. Stefik.	2, 5	1983	États-Unis (Xerox Parc)	Système général. -Démonstrateur de théorèmes	-	Réseaux sémantiques ou Frames	-	-	-	-	-	1) Mode d'inférence est défini par l'utilisateur. 2) Support graphique pour la modification de la base de connaissance. 3) Le programme permet de montrer différentes portions de la base de connaissance. 4) Peut être implanté sur: - INTERLISP, - Xerox 1100S.

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
M1 (de EMYCIN)	-	(1), 12	1984	États-Unis (Teknow-ledge)	Système général. -Diagnostic -Prescription	Maximum 300	Règles Si-alors	PROLOG	Fonc-tionnel	Chaînage arrière en pro-fondeur d'abord	-	EMYCIN	1) Descendant modernisé et simplifié de EMYCIN. 2) Fonctionne sur IBM-PC. 3) 10 000 $ US (M1); 2 500 $ US (M1A). 4) Disponible chez: TEKNOWLEDGE, 525 University Ave., Palo Alto, CA 94301, USA, Tél.: (415) 327-6606. 5) Peut représenter des connaissances incertaines. 6) Retraçage des inférences. 7) Peut être implanté sur: - IBM PC (192 K).
MacEXPERT	-	(9)	-	- (Mind Soft)	Système général.	-	-	-	-	-	-	-	
MERING	J. Ferber	(5)	1984	France	Système général.	-	Frames	-	-	-	-	-	1) S'inscrit comme une généralisation des langages de type SMALLTALK. 2) Haut degré de modularité. 3) Il communique par envoie de messages. 4) Les mécanismes de filtrage qui permettent de mettre en correspondance des entités de la base en tenant compte des contraintes et des "réflexes". - Réflexes: sorte de procédures attachées, associées aux attributs. 5) Structures de contrôle élaborées telles que agenda.
METABOL	-	1	-	-	Système général (extension de METALOG).	-	Logique des prédicats	-	-	-	-	-	1) Est utilisé pour développer "MANAGER".

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
METALOG	-	1	-	-	Système général (extension de PROLOG).	-	Logique des prédicats	-	-	-	-	-	1) A servi à l'analyse et la synthèse de circuits électroniques passifs (PEACE 1, PEACE 2).
METHODS	-	(12)	-	États-Unis	Système général.	-	-	ASSEM-BLER et BASIC	Fonc-tionnel	-	-	-	1) Fonctionne sur IBM-PC. 2) 250 $ US. 3) Disponible chez: Digitalk Inc., 5200 W. Century Blvd., Los Angeles, CA 90045, USA, Tél.: (213) 645-1082.
MICRO-EXPERT	-	-	-	-	-	-	-	PASCAL	-	-	-	-	1) Peu complexe. 2) Ce logiciel ne peut pas servir au développement de systèmes-experts commerciaux. 3) Fonctionnement par chaînage arrière.
MICROMYC 1, 2	-	(1)	-	-	Système général.	-	-	-	-	-	-	-	
MICROPLANNER	-	5, 8	-	-	Système général.	-	Logique des prédicats	-	-	-	-	-	1) Descendant de PLANNER.
MICROSYMBOL (MICROMOTOR)	-	(1)	-	-	Système général.	-	-	-	-	-	-	-	
MIME dans (SATIN)	S. Brau-Nogue	(1)	1984	France (L.S.I., E.N.S.E.E.I.H.T.) et (Univ. P. Sabatier)	Système général. -Transfert de technologies	-	-	-	-	-	-	-	1) Moteur simple. 2) Un des modules du système multi-expert "SATIN". 3) SATIN est développé pour faire coopérer des expertises juridiques, économiques, technologiques et documentaires en vue de réaliser des transferts de technologie. 4) Le moteur fonctionne en alternance entre chaînage avant et chaînage arrière.

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
MIRLITHO dans (LITHO)	J.G. Ganascia	9, 13	-	France	Module d'interprétation de résultats.	-	-	-	-	-	-	-	
MP-LRO	-	(9)	-	France (CRIL)	Système général.	-	-	-	-	-	-	-	
MRS	-	(2)	-	États-Unis (Univ. Stanford)	Système de haut niveau pour la représentation de la connaissance et le contrôle de résolution de problèmes.	-	-	-	-	-	-	-	
NOAH	E.D. Sacerdoti	1, 5	1977	États-Unis (SRI)	Système général (générations de plans). -Robotique	-	Logique des prédicats	QLISP	-	-	-	-	1) A été appliqué à des exemples d'assemblage de pièces électromécaniques. 2) L'homme exécute des tâches matérielles de montage et démontage sous les conseils d'un robot. 3) Engendre des plans par degré progressif de détails selon les demandes de l'apprenti.
OPS (OPS 5+)	C. Forgy, J. McDermott.	1, 2, 5, 12	1977	États-Unis (Univ. Carnegie Mellon)	Système général. -Contrôle de processus -Planification	Maximum 1 500	Règles de production	C	Fonc-tionnel	Chaînage mixte. Stratégie par tentatives.	-	-	1) Fonctionne sur IMB-PC (OPS 5+). 2) 3 000 $ US. 3) Disponible chez: Artelligence Inc., 1402 Preston Road, Dallas, TX 75240, USA, Tél.: (214) 437-0361. 4) Dispose d'une technique évoluée de compilation de la base de connaissances lui permettant de réduire le temps consacré à déterminer les règles à déclencher. 5) Ordre 1. 6) Est un développement de PS. 7) Le modèle de compatibilité est la semi-unification.

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
OPS (OPS 5+) (suite)													8) Les règles seulement comportent des variables. 9) Peut fonctionner de manière non monotone. 10) Règle Si-alors. 11) Peut être implanté sur: - Franz LISP, - Mac LISP, - VAX 11/780. Note: La version commercialisée par Verac Inc. peut être implantée sur: - Zeta LISP, - Symbolics 3600.
ORBIT	L. Steels	(5)	1982	États-Unis (Schlum-berger)	Système général.	-	Réseaux sémantiques ou Frames	-	-	-	-	-	
OURCIN	E. Demonchaux, J. Quinqueton.	(1)	1984	France (I.N.R.I.A. - S.E.M.A.)	Système général. -Médecine -Assurance -Paléontologie -Analyse des données	-	-	-	-	-	-	-	1) Ordre 0. 2) Un filtre ne peut être compatible qu'avec une expression de fait parfaitement identique. 3) Effort particulier consacré à la communication entre l'utilisateur et le système. 4) Système simple. 5) Des applications ont été réalisées en analyse des données et paléontologie alors que d'autres sont envisagées dans le domaine médical et le domaine des assurances.
OWL	P. Szolovits, S.G. Pauker.	(5), 15	1978	États-Unis (MIT)	Système général.	-	Réseaux sémantiques	-	-	-	-	-	

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
PARSEC	-	(9)	-	France (ELF-Aquitaine)	Système général.	-	-	-	-	-	-	-	
PAS-II	D.A. Waterman	(1)	1975	États-Unis (Univ. Carnegie Mellon)	Système général. -Apprentissage	-	-	-	-	Chaînage avant	-	-	1) Utilise des primitives capables de modifier la base de connaissance (métarègles). 2) Le modèle de compatibilité est la semi-unification. 3) Les règles peuvent comporter des variables mais non les faits. 4) A été construit en vue d'évaluer dans quelle mesure les systèmes de production se prêtent à des formes d'apprentissage. 5) Plusieurs jeux de règles ont été définis pour illustrer la résolution de tests d'intelligence du type continuation de séries numériques ou alphabétiques. 6) Primitives de déclaration dynamique de règles (système-expert dit adaptatif).
PERSONAL CONSULTANT	-	(12)	-	États-Unis (Texas Instruments)	Système général. -Diagnostic -Prescription	400	Règle Si-alors	IQ LISP	Fonc-tionnel	Chaînage arrière en pro-fondeur d'abord	-	-	1) Fonctionne sur IBM-PC. 2) 3 000 $ US. 3) Disponible chez: Texas Instruments, P.O. Box 2909, Austin, TX 78769, USA, Tél.: (800) 527-3500. 4) Peut représenter des connaissances incertaines ou imprécises. 5) Éditeur de la base de connaissance. 6) Retraçage des inférences. 7) Peut être implanté sur: - TI Professional et Portable Computers, - IQ LISP.

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
PILOTEX	Husson	(1)	1985	France (Société IIMI) Grenoble	Système général (suivi de procesaus continu).	-	-	-	Recher-che	-	-	-	1) Peut exploiter les connaissances incertaines. 2) Peut fonctionner de manière non monotone.
PLANNER	C. Hewitt	5, 10, 15, 8	1972	États-Unis (MIT)	Système général. -Reconnaissance du langage écrit	-	Logique des prédicats	LISP	-	Chaînage arrière	-	-	1) Pour la représentation de la connais-sance procédurale. 2) Fut implanté sous la version MICRO-PLANNER. 3) Capable de spécifier si les règles logiques peuvent être utilisées en raisonnement avant ou raisonnement arrière. 4) En plus de recommander les règles possiblement utiles, des classes géné-rales de règles peuvent être suggérées par l'utilisation de filtres. 5) Est capable d'appliquer un raisonnement par défaut. 6) Utilise la non-monotonie. 7) Est incapable de considérer la forme générale de la solution du problème. 8) Ne peut raisonner sur son information de contrôle (ne peut retracer son raisonnement). Note: Comme CONNIVER, ce moteur d'infé-rence n'est plus utilisé aujour-d'hui, mais a eu une profonde influence sur le mode de fonction-nement des moteurs d'inférence existant présentement.
PROLOG	P. Roussel	1, 2, 5, 9	1975	France (Univ. Marseilles)	Système général.	-	Logique des prédicats	LISP	-	Chaînage arrière en	-	-	1) Ordre 1. 2) Le modèle de compatibilité est l'unification.

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
PROLOG (suite)										profon-deur d'abord			3) Utilise le mode d'inférence appelé principe de résolution. 4) Stratégie par tentative. 5) Capable de non-monotonie.
PSC	P. Dufresne, M. Ghallab.	(1)	1984	France (L.A.A.S., Toulouse)	Système général (robotique).	-	-	-	-	-	-	-	1) La technique évoluée de compilation de la base de connaissances lui permet de réduire fortement le temps consacré à déterminer les règles à déclencher (indispensable en robotique). 2) Destiné à prendre place dans le système de décision du robot HILARE. 3) Cette technique permet à des systèmes de fonctionner en temps réel.
R.L.L.	R. Greiner, D. Lenat.	-	-	États-Unis (Univ. Stanford)	-	-	-	-	-	-	-	-	1) Il dispose d'une connaissance sur son propre fonctionnement: - sa structure, - ses modes d'inférence. 2) Comprend un grand nombre de composantes: - différents types de schéma de repré-sentation possible, - de nombreuses structures de contrôle: ex.: une structure de contrôle fondée sur la notion d'agenda permet d'organiser des tâches selon certains modes de priorité.
RAINBOW	C.R. Hollander, H.C.	(5)	1979	États-Unis (IBM, Palo Alto)	Système général.	-	Règles de production	-	-	-	-	-	
ROSIE dans (TATR)	-	(2)	-	États-Unis (RAND Cor-poration) Californie	Système général.	-	-	-	-	-	-	-	

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
RULE MASTER	-	(12)	-	États-Unis	Système général.	200	-	C	Fonc-tionnel	-	-	-	1) Fonctionne sur IBM-PC. 2) 5 000 $ US. 3) Disponible chez: Radian, 8501 MO-Pac Blvd., Austin, TX 78766, USA, Tél.: (512) 454-4797.
S1	-	(9)	1984	États-Unis (Teckowl-edge Inc.)	Système général. -Diagnostic -Prescription	-	-	LISP	-	Chaînage arrière en pro-fondeur d'abord	-	-	1) Fonctionne sur XEROX, Symbolics ou Digital Equipment. 2) Peut représenter des connaissances imprécises ou incertaines. 3) Explication sur l'inférence (comment et pourquoi). 4) Interface graphique.
SACSO	P. Debord, P. Dalle, S. Castan.	(1)	1981	France (C.E.R.F. I.A.) et (Univ. P. Sabatier)	Système général (orienté vers l'ana-lyse de scènes).	-	-	-	-	-	-	-	
SAGE	-	(2)	-	- (SPL Inter-national)	Système général.	-	-	-	-	-	-	-	
SERIES PC	-	(12)	-	États-Unis (SRI)	Système général. -Diagnostic -Prescription	300	Règle Si-alors	IQ LISP	Fonc-tionnel	Chaînage arrière en pro-fondeur d'abord	-	-	1) Fonctionne sur IBM-PC. 2) 15 000 $ US. 3) Disponible chez: SRI International, 333 Ravenswood Ave., Menlo Park, CA 94025, USA, Tél.: (415) 859-5889. 4) Retraçage des inférences. 5) Un éditeur de la base de connaissance.
SMALLTALK	A. Kay, A. Goldberg.	(5)	1977	États-Unis (Xerox)	Système général.	-	Frames	-	-	-	-	-	1) Langage-objets.

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
SMP	-	(1)	1984	France (CRIL, LIA Univ. de Savoie Chambery)	Système général.	-	-	-	-	-	-	-	1) Dérivé de EAQUE.
SNARK dans (SIMMIAS)	J.L. Laurière	1, (5), 9	1981	France (G.R. 22 + Univ. Paris VI)	Système général. -Géologie -Archéologie -Enseignement en physique	-	Logique des prédicats	PASCAL PL/1	Fonc-tionnel	Chaînage avant	-	-	1) Ordre 1. 2) Plusieurs versions ont été réalisées. 3) Système français le plus connu et le plus diffusé. 4) Les versions les plus récentes comportent des primitives pour exprimer des métaconnaissances et remettre en cause des hypothèses. 5) A inspiré le développement du moteur ALOUETTE. 6) Le modèle de compatibilité est la semi-unification. 7) Dispose de primitives qui agissent sur le moteur d'inférence. 8) Utilise des métarègles. 9) Le corps des métarègles peut faire appel à des primitives d'inhibition et validation de règles, elles-mêmes désignées associativement. 10) Les règles seulement comportent des variables. 11) Peut exploiter les connaissances incertaines. 12) Utilise une fonction heuristique intrinsèque au moteur. 13) Adopte une stratégie par tentative. 14) Peut fonctionner de manière non monotone. 15) Structuration de la base de connaissance.

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
SNARK dans (SIMMIAS) (suite)													16) Les règles sont caractérisées par un ensemble d'actions qui sont déclenchés si les prémisses de la partie condition peuvent être mises en correspondance avec des faits de la base. 17) Record pour la vitesse des calculs. 18) Faiblesses: - son manque d'environnement et d'assistance à l'utilisateur; - le format des règles et des faits qui doivent être rédigés dans la base est très stricte.
SNePS	J.P. Martin, S.C. Shapiro.	(3)	1979	États-Unis	Système général.	-	-	LISP	-	-	-	-	
SPII-1	R. Martin-Clouaire	(1)	1984	France (L.S.I., E.N.S.E.E.I.H.T.) + (Univ. P. Sabatier)	Système général.	-	-	-	-	Chaînage arrière	-	-	1) Capable de traiter des faits incertains ou imprécis. 2) Met en oeuvre la théorie des possibilités.
SPIRAL	-	(9)	-	- (CEA)	Système général.	-	-	-	-	-	-	-	
SPROUTER	B. Hayes-Roth	(L)	-	États-Unis	Système général.	-	Frames	-	-	-	-	-	1) A été étendu dans l'analyse du langage.
STROBE	-	(11)	-	États-Unis (Schlumberger Doll Research)	Système général.	-	-	-	-	-	-	-	
SYPRUC	-	(9)	-	- (Cimsa)	Système général.	-	-	-	-	-	-	-	

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
TANGO	M.O. Cordier, M.C. Rousset.	(1), 5	1982	France (LRI, Univ. Paris-Sud) Orsay	Système général (utilisation en EAO).	-	Logique des prédicats	LISP	-	Chaînage bidirectionnel	-	-	1) Ordre 1. 2) On ne peut explicitement qu'ajouter des faits (défaut atténué par le fait que le moteur peut, de sa propre autorité, supprimer un fait s'il survient un ajout déjà présent dans la base). 3) Technique évoluée de compilation de la base de connaissances, lui permettant de réduire le temps consacré à déterminer les règles à déclencher. 4) Le temps gagné à la détermination des règles à déclencher risque d'être atténué par une consommation prohibitive d'espace-mémoire. 5) Le modèle de compatibilité est la semi-unification. 6) Permet l'introduction de procédures dans les règles et métarègles. 7) Adopte une stratégie irrévocable. 8) Structuration de la base des règles. 9) Le mode de contrôle permet de guider le raisonnement par un plan. 10) Le système distingue explicitement la notion d'implication de celle d'action sur une base des faits. 11) S'il y a déclenchement d'une action, cette action modifiera la base des connaissances. 12) TANGO construit un plan de résolution qui permet de choisir la règle à déclencher en tenant compte de buts immédiats, définis par un certain nombre de critères à chaque cycle de résolution. 13) Le type de raisonnement appliqué par le système est un chaînage avant guidé par un plan construit en chaînage arrière.

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
TEIRESIAS	R. Davis, D.B. Lenat.	2, 9, (4), 10,15, H	1980	États-Unis (Univ. Stanford)	Système de transfert de connaissance d'un expert humain à un système et guide l'acquisition de nouvelles règles d'inférence.	-	-	-	-	-	-	-	1) Peut organiser les connaissances données en vrac. 2) Capable de détecter, dans la base des connaissances, les lacunes ou les incohérences.
TEK 4404	-	(9)	-	- (Tektronix)	Système général.	-	-	-	-	-	-	-	
THOTH	S.A. Vere	(L)	-	États-Unis	Système général.	-	-	-	-	-	-	-	
TIGRE-1	-	(9)	-	France (Cognitech)	Système général.	-	-	-	-	-	-	-	
TIMM	-	(12)	1983	États-Unis (General Research Corp.)	Système général.	500	-	FORTRAN 77	Fonc-tionnel	-	-	-	1) Fonctionne sur IBM-PC. 2) 9 500 $ US. 3) Disponible chez: General Research Inc., 7655 Old Springhouse Road, McLean, VA 22102, USA, Tél.: (703) 893-5900. 4) Utilisé par Hugues Aircraft pour la simulation des effets des décisions d'un pilote d'hélicoptère lors de batailles. 5) Peut représenter des connaissances incertaines. 6) Chaînage avant. 7) Explication de l'inférence. 8) Peut être implanté sur: - IBM, - DEC, - Prime, - IBM PC XI.

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
TOPS1	-	(12)	-	États-Unis	Système général.	5 000	-	TURBO PASCAL	Fonc-tionnel	-	-	-	1) Fonctionne sur IBM-PC. 2) 75 $ US. 3) Disponible chez: Dinamic Master Systems, P.O. Box 566456, Atlanta, GA 30356, USA, Tél.: (404) 425-7715.
TROPIC	J.C. Latombe	(1)	1977	France (E.N.S.I. M.A.G.) Grenoble	Système général (orienté vers la CAO).	-	-	-	-	Chaînage arrière	-	-	
UNITS	M. Stefik	2, 5	1979	États-Unis (Stanford)	Système général.	-	Frames	-	-	-	-	-	
XSYS	-	(12)	-	États-Unis	Système général.	-	-	IQ LISP	Fonc-tionnel	-	-	-	1) Fonctionne sur IBM-PC. 2) 1 000 $ US. 3) Disponible chez: California Intelli-gence, 912 Powell Street, San Francisco, CA 94108, USA, Tél.: (415) 391-4848.
ZERO+	M.O. Cordier, M.C. Rousset.	9, 14	-	France (LRI, Univ. Paris-Sud) Orsay	Système général.	-	-	-	-	-	-	-	

SYSTÈMES D'OPÉRATION

Description des systèmes experts

STEAMER

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
STEAMER	Hollan, Hutchins, Weitzman.	(15)	1984	-	Opération de centrales thermiques.	-	-	-	-	-	-	-	1) Graphics coupled to mathematical model of plant.

TRADUCTION AUTOMATIQUE

Description des systèmes experts

ALPS

ARIANE 78

SUZY TITIUS

SYSTRAN

TAUM METEO

TITIUS et TITIUS IV

TITRAN

TITRAN EJ

TITRAN JE

TITRAN JF

WEIDNER

YAMATO

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
ALPS	-	(14)	1980	France (Control Data)	Traduction automatique.	-	-	-	Fonctionnel	-	-	-	1) Commercialisé sous le nom de TRANSMATIC.
ARIANE 78	B. Vauquois	(14)	1978	France (Univ. de Grenoble)	Traduction automatique.	-	-	-	-	-	-	-	1) Le dictionnaire et le modèle linguistique utilisés sont séparés de la partie logiciel (contrairement à plusieurs autres systèmes de l'I.A.). 2) Possibilité d'améliorer les grammaires ou le modèle linguistique. 3) La séquence des opérations effectuées se divise en trois grandes étapes: - analyse (décortiquer le texte); - transport (retrouver, dans la langue cible, des représentations équivalentes); - génération (d'un texte intelligible). 4) La cinquième version de Ariane 78 a été mise au point par le GETA (Groupe d'Études pour la Traduction Automatique). 5) Système de deuxième génération.
SUZY TITIUS	-	(14)	1974	- (Univ. de Sarrebruck)	Traduction automatique.	-	-	-	-	-	-	-	
SYSTRAN	-	9, 14	1970	-	Traduction automatique: (Anglais - Français); (Français - Anglais); (Anglais - Italien).	-	-	-	Fonctionnel	-	Traduit 320 pages de texte dactylographié à l'heure (la qualité de la traduction laisse toutefois à désirer).	-	1) Bien qu'il traduise plusieurs centaines de pages par an, toute cette production doit repasser entre les mains d'un correcteur humain avant d'être utilisée. (!!!) 2) Ne possède pas clairement de modèle linguistique. 3) Difficile de prévoir le fonctionnement et d'améliorer la qualité de la traduction.

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
SYSTRAN (suite)													4) Impossible de savoir au départ quelles sont les possibilités réelles du système. 5) Beaucoup de restrictions sur les formes d'expression: nombre de mots limité, sens des mots défini, contraintes sur le style. 6) Fonctionne sur IBM 4341. 7) Système de première génération. 8) "La qualité des traductions" doit beaucoup au travail des Anglais MM. Masterman et V. Lawson.
TAUM METEO	-	(14)	1977	Canada (Univ. de Montréal)	Traduction automatique de bulletins de météo (Anglais - Français).	-	-	-	Fonctionnel	-	Seul système de T.A. fonctionnant 24 heures par jour sans qu'aucune grande modification lui ait été apportée (80% des bulletins sont traduits sans intervention humaine).	-	1) Incapacité à traduire complètement toutes les phrases (quand cette situation se produit cette (ces) phrase(s) est(sont) affichée(s) sur un écran afin qu'un humain s'en charge). 2) Vu le domaine d'application, il est possible de distinguer clairement les phrases que la machine peut ou non traduire. 3) Système rentable. 4) Conçu pour le gouvernement du Canada.
TITIUS et TITIUS IV	-	(14)	1981 (Titius IV)	France (Institut textile de France (ITF))	Traduction automatique (conçu pour le traitement multilingue de bases de données dans les	-	-	-	Fonctionnel	-	Bonnes performances	-	1) Système à "syntaxe contrôlée": n'autorise que l'emploi de formes d'expression obéissant à certains critères linguistiques restreints et prédéterminés!

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
TITIUS et TITIUS IV (suite)					domaines scientifiques et techniques).								2) Les phrases ne doivent contenir que des termes figurant dans un dictionnaire préalablement établi. 3) Doit déterminer quel sens donner à un mot, lorsque celui-ci en comporte plusieurs. 4) Le système traduit simultanément en allemand, anglais, espagnol et français. 5) Devant le succès de TITIUS, l'ITF s'est efforcée d'atténuer la rigueur des règles d'écriture pour conduire à un système de conception nouvelle, TITIUS IV, lancé en 1981.
TITRAN	-	(14)	-	Japon	Traduction automatique (traduit des titres d'articles scientifiques et des références bibliographiques).	-	-	-	Recherche	-	-	-	1) A engendré: TITRAN EJ (1981); TITRAN JE (1982); TITRAN JF (1982).
TITRAN EJ	M. Nagao	(14)	1981	Japon (Univ. Kyoto)	Traduction automatique, anglais-japonais (titres d'articles scientifiques et références bibliographiques).	-	-	-	-	-	Taux de traduction de 90%.	-	1) Le dictionnaire est constitué de mots simples et de vocabulaire technique. 2) Application limitée.
TITRAN JE	M. Nagao	(14)	1982	Japon (Univ. Kyoto)	Traduction automatique (titres d'articles scientifiques et références bibliographiques).	-	-	-	-	-	90% des titres ont été traduits correctement par le système.	-	1) Le dictionnaire est constitué de mots simples et de vocabulaire technique. 2) Application limitée. 3) Étapes de traduction: - découper en mots le texte japonais qui se présente sous la forme d'une suite de caractères;

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
TITRAN JE (suite)													- après identification des groupes syntaxiques, le programme procède à leur transfert.
TITRAN JF	J. Hubert	(14)	1982	-	Traduction automatique (titres d'articles scientifiques et références bibliographiques).	-	-	-	-	-	90% des titres ont été traduits correctement par le système.	-	1) Le dictionnaire est constitué de mots simples et de vocabulaire technique. 2) Application limitée. 3) Étapes de traduction: - découper en mots le texte japonais qui se présente sous la forme d'une suite de caractères; - après identification des groupes syntaxiques, le programme procède à leur transfert.
WEIDNER	-	(14)	1980	France	Traduction automatique.	-	-	-	-	-	-	-	
YAMATO	-	(14)	1959	Japon	Traduction automatique.	-	-	-	-	-	Système de très bas niveau.	-	
-	Wanying Jin	(3)	1986	États-Unis (Univ. du Texas à Austin)	Traduction automatique (chinois - anglais).	-	-	-	Recherche	-	Le système a été testé sur un nombre limité de phrases (8) et six se sont avérées acceptables et deux médiocres.	-	1) La qualité de la traduction laisse souvent à désirer puisqu'il s'agit d'une traduction mot-à-mot. 2) Temps de traduction relativement long.

ZOOLOGIE

Description des systèmes experts

ZOOLOG

NOM	AUTEUR(S)	RÉFÉ-RENCES	ANNÉE	ORIGINE	APPLICATIONS	NOMBRE DE RÈGLES	MODE DE REPRÉ-SENTATION DES CONNAISSANCES	LANGAGE	PHASE ACTUELLE	FONCTION-NEMENT	PERFORMANCES	MOTEURS D'INFÉRENCE DÉJA EXISTANTS	COMMENTAIRES GÉNÉRAUX
ZOOLOG	-	(1)	-	France	Zoologie.	-	-	LISP	-	Chaînage arrière	-	MICROMYC 1	1) Ordre 0. 2) En profondeur d'abord. 3) micro-système-expert. 4) Peut répondre à des questions sur son raisonnement. 5) Fonctionne sur TRS-80.

REFERENCES

1.- Farreny,H.; "Les systèmes experts: principes et exemples.", Collection Techniques Avancées de l'Informatique, Editions Cepadues, Toulouse, 1985, 254 pages.

2.- Feigenbaum, E.A.; McCorduck, P.; "La 5-ième génération: le pari de l'intelligence artificielle à l'aube du 21-ième siècle.", Intereditions, Paris, 1984.

3.- Société Canadienne pour l'Etude de l'Intelligence par Ordinateur; Actes de la sixième conférence canadienne sur l'intelligence artificielle, Ecole Polytechnique de Montréal, 1986, 268 pages.

4.- Fieschi, M.; "Intelligence artificielle en médecine (des systèmes experts).", Editions Masson, Paris, 1984, 207 pages.

5.- Bonnet, A.; "L'Intelligence artificielle: promesses et réalités.", Interéditions, Paris, 1984, 271 pages.

6.- Cohen, P.R.; Feigenbaum, E.A.; "The Handbook of Artificial Intelligence (volume 1), Heuristech Press, Stanford, 1982, 409 pages.

7.- Cohen, P.R.; Feigenbaum, E.A.; "The Handbook of Artificial Intelligence (volume 2), Heuristech Press, Stanford, 1982, 428 pages.

8.- Cohen, P.R.; Feigenbaum, E.A.; "The Handbook of Artificial Intelligence (volume 3), Heuristech Press, Stanford, 1982, 640 pages.

9.- Revue La Recherche, mensuel No.170, Octobre 1985, Paris.

10.-Yazdani, M.; Narayanan, A.; "Artificial intelligence: human effects.", Ellis Harwood Series, Université de Exetes, Angleterre, 1984, 318 pages.

11.-University of California at Los Angeles, Proceedings of the Ninth International Joint Conference on Artificial Intelligence (Volume 1), Los Angeles, 1985.

12.-University of California at Los Angeles, Proceedings of the Ninth International Joint Conference on Artificial Intelligence (Volume 2), Los Angeles, 1985.

13.-Revues Microsystèmes.

14.-Revues La Recherche.

15.-Ecole Polytechnique de Montréal, Journée Augustin Frigon (Intelligence artificielle et systèmes experts), Ecole Polytechnique de Montréal, 1984.

16.-Proceeding of the First International Joint Conférence on Artificial Intelligence, 1977.

17.-Revue Artificial Intelligence, Volume5, 1974.

18.-Proceeding of the Fifth International Joint Conférence on Artificial Intelligence, 1981.

19.-Revue Artificial Intelligence, Volume 2, 1971.

20.-Revue Artificial Intelligence, Volume 3, 1972.

21.-Revue Artificial Intelligence, Volume 4, 1973.

22.-Revue Artificial Intelligence, Volume 6, 1975.

23.-Revue Artificial Intelligence, Volume 8, 1977.

24.-Revue Artificial Intelligence, Volume 10-11, 1978.

25.-Revue Artificial Intelligence, Volume 20-21, 1983.

26.-Revue Artificial Intelligence, Volume 22, 1984.

27.-Revue Artificial Intelligence, Volume 23-24, 1984.

28.-Revue Artificial Intelligence, Volume 25, 1985.

29.-Revue Artificial Intelligence, Volume 26, 1985.

30.-Fox, M.S.; "Industrial applications of expert systems.", Centre de cours intensifs de l'Ecole Polytechnique de Montréal, 1986, 45 pages.

31.-Buchanan, B.G., "Expert systems: working systems and the reseach literature.", Knowledge Systems Laboratory, Stanford University, December 1985, 55 pp.

32.-Knowledge Systems Laboratory; "Knowledge Systems Laboratory, 1985.", Stanford University, 1985, 94 pp.

ANNEXE 2

MATÉRIEL INFORMATIQUE EN INTELLIGENCE ARTIFICIELLE

La présente annexe veut donner un inventaire des entreprises qui fabriquent et commercialisent le matériel informatique spécifique à l'intelligence artificielle et plus particulièrement aux systèmes experts.

Le présent inventaire ne prétend pas être complet, mais peut être considéré comme un point de départ d'une recherche qui pourrait être plus exhaustive.

Résumé: environnement et outils commerciaux

Sur micro-ordinateur	Sur machine LISP
M.1	S.1
EXSYS	ART
OPS5	KEE
PERSONAL CONSULTANT	DUCK
INSIGHT	KNOWLEDGECRAFT
EXPERT-EASE	OPS5
TIMM	EXPLORER
MICRO-EXPERT	LOOPS
KDS	
RULE MASTER	
AION	
KES	
SERIES-PC	
ES/P ADVISOR	

MATÉRIEL INFORMATIQUE EN INTELLIGENCE ARTIFICIELLE

COMPAGNIE	ADRESSE	ÉQUIPEMENT ET LOGICIEL
ACT-Informatique	France	- Le LISP. - Environnements pour systèmes experts.
ACT-Informatique	France	- LISP. - Langage COOL, CEYX (langage objets). - EXPERTKIT: pour systèmes experts.
ALMA	France	- PROLOG-CRISS.
Amaïa	France	- LISLOG.
Applied Expert System	États-Unis	
Apollo Computer	Cambridge, Mass.	- Compilateur COMMON LISP. - "Explorer LISP Machine" de Texas Instrument avec un interface compatible avec ses propres stations. - Divers environnements pour systèmes experts.
Artificial Int. Corp.	États-Unis	
AT/T		- Outils pour la reconnaissance du langage parlé avec différents interlocuteurs. - Reconnaissance des mots. - Traitement grammatical. - Reconnaissance du sens du contexte. - Environnement UNIX.
Bull	États-Unis	- XLOG. - Environnement KOOL pour le développement de systèmes experts.
Cap Sogeti	France	
Cimsa	France	- Environnement SYPRUC pour le développement de systèmes experts.
CISI	France	

MATÉRIEL INFORMATIQUE EN INTELLIGENCE ARTIFICIELLE (suite)

COMPAGNIE	ADRESSE	ÉQUIPEMENT ET LOGICIEL
Computer Thought	États-Unis	
Cognitek	France	- Propose TIGRE-1.
Cognitive System	États-Unis	
CRI (CSK Research Institute) de Computer Service Corp.	Japon	- Environnement KEE pour le développement de systèmes experts.
CRIL	France	- Prolog P, et l'environnement MP-LRO pour systèmes experts.
Data General	Westboro, Mass.	- Compilateur COMMON LISP. - Environnement pour systèmes experts pour terminals IBM-PC/AT et compatibles.
Delphia	France	- D-PROLOG 3.
Digital Equipment Company (DEC)		- Micro VAXII avec compilateur en COMMON LISP et interface graphique avec la station Al VAX STATION (MicroVAXII bientôt disponible pour environ $15,000.00 U.S.). - Compilateur COMMON LISP. - Logiciel OPS5, PROLOG. - Divers environnements pour systèmes experts (KNOWLEGGECRAFT, S1) - PROLOG II.
Expert Software International	États-Unis	- Environnement EXPERT-EASE pour systèmes experts, coût $2,000.00 U.S. - Environnement ES/P ADVISOR pour systèmes experts, coût $1,895.00 U.S.
Expert-System International	États-Unis	

MATÉRIEL INFORMATIQUE EN INTELLIGENCE ARTIFICIELLE (suite)

COMPAGNIE	ADRESSE	ÉQUIPEMENT ET LOGICIEL
Framentec	France	- Matériel Xerox, Symbolics, Digital. - Équipement, outil de développement de systèmes experts sur IBM-PC et compatibles (S1, M1). - PROLOG.
Franz	Alamedo, Californie	- Compilateur LISP, COMMON LISP.
General Research Corporation	États-Unis	- Environnement TIMM pour systèmes experts, coût $9,000.00 U.S. sur IBM-PC XT.
Gold Hill	Cambridge, Mass.	- Compilateur LISP, COMMON LISP.
Hewlett Packard	États-Unis	- Compilateur COMMON LISP. - HP 9000. - Divers environnements pour systèmes experts.
I.B.M.	États-Unis	- Environnement pour systèmes experts. - Environnement / MVS. - ESE/VM. - Interface pour le traitement du langage naturel.
Inference Corp.	États-Unis	- Environnement ART pour systèmes experts, coût $60,000.00 U.S.
Intell. Corp.	États-Unis	- Environnement KEE pour systèmes experts, coût $50,000.00 U.S. - Langage KRL+.
Intelligence Thinking Machine	États-Unis	
Intelligent Software	États-Unis	
ITMI	France	- Environnement PILOTEX pour le contrôle des processus et maintenance.

MATÉRIEL INFORMATIQUE EN INTELLIGENCE ARTIFICIELLE (suite)

COMPAGNIE	ADRESSE	ÉQUIPEMENT ET LOGICIEL
Knowledge System	États-Unis	- INSIGHT: environnement pour systèmes experts, coût $95.00 U.S.
LMI		- LM-2. - PICON systèmes experts pour le contrôle de processus. - LM-PROLOG, Flavors (langages).
Lucid	Palo Alto, Californie	- Compilateur LISP, COMMON LISP.
Machine Intelligence	États-Unis	
Micro-Informatique	France	- QLISP, OPS-5.
Mind Soft.	France	- EXPERT I, MacEXPERT.
Mitsubishie Electric	Japon	- À venir: machine à inférence séquentielle (architecture hautement parallèle) (langage PROLOG). - Projets: -Gestion intelligente de base de données. -Traitement du langage naturel: notamment un système pour la traduction du japonais à l'anglais.
NCR	États-Unis	- Compilateur COMMON LISP. - Pour bientôt: revente de divers environnements pour systèmes experts.
PRIME	États-Unis	- Pour bientôt: vente du compilateur COMMON LISP.
PROLOG 1A	France	- PROLOG II (MacIntosh, VAX, HP).
S.R.I. International	États-Unis	- SeRIES-PC: environnement pour systèmes experts, coût $5,000.00 U.S.
SEMA	France	

MATÉRIEL INFORMATIQUE EN INTELLIGENCE ARTIFICIELLE (suite)

COMPAGNIE	ADRESSE	ÉQUIPEMENT ET LOGICIEL
Sperry	États-Unis	- "Explorer LISP Machine" de Texas Instrument. - Environnement KEE pour le développement de systèmes experts.
Sun Microsystems	France	- PROLOG, LISP, C. - Série station 68010, 68020. - Station UNIX.
Sun Micro-System	Mountain View, California	- "Explorer LISP Machine" de Texas Instrument, un interface compatible avec ses propres stations.
Symbolics	Concord, Mass.	- 3610 AE, LISP Machine, coût $39,600.00 U.S. - PROLOG. - Environnement KEE pour systèmes experts.
Syntelligence	États-Unis	
Tecknowledge	États-Unis	- PROLOG sur IBM-PC, coût $2,000.00 U.S. - Environnement S.1 pour systèmes experts, coût $50,000, $80,000.00 U.S.
TECSI	France	- Divers environnements pour systèmes experts. - KNOWLEDGECRAFT.
Tektronix	États-Unis	- SMALLTALK. - TEK 4404. - Stations de travail SUN et APOLLO.
Texas Instrument	Dallas, Texas	- Production d'un micro-processeur LISP à l'automne 1986: "Explorer Mega Chip". - Projet de construction d'un "Compact LISP Machine" (contrat avec la DARPA) qui ne prendrait pas plus d'espace qu'une boîte à soulier. - Explorer LISP Machine. - Environnement "Personal Consultant Plus" pour le développement de systèmes experts. Coût: $1,000.00 U.S. et plus; sur IBM-PC ou compatible. - PROLOG.

MATÉRIEL INFORMATIQUE EN INTELLIGENCE ARTIFICIELLE (suite)

COMPAGNIE	ADRESSE	ÉQUIPEMENT ET LOGICIEL
Verac Inc.		- OPS5: sur ZetoLISP et Symbolics 3600, coût $3,000.00 à $10,000.00 U.S. dépendant de l'environnement.
Xerox	Pasadena, Californie	- LISP Machine: Station 1185, coût $10,000.00 U.S. Peut être mis en interface avec IBM-PC. - Série 1100 machine LISP (INTERLISP). - Bientôt un compilateur COMMON LISP. - Ont développé les langages SMALLTALK et LOOPS (langages objets) (LOOPS coût $300.00 U.S.).

ANNEXE 3

LE LANGAGE PROLOG

TABLE DES MATIÈRES

1.0 INTRODUCTION

Prolog est un langage qui s'écarte radicalement des voies suivies à partir du principe de la programmation impérative, qui oblige l'informaticien à indiquer pas à pas à l'ordinateur ce qu'il doit faire pour calculer le résultat qu'il veut obtenir, il prend plutôt profit de la programmation déclarative qui consiste à représenter dans un formalisme adéquat des connaissances sur un sujet donné, à partir desquelles seront déduites les réponses aux questions de l'utilisateur.

La langage Prolog a été conçu et amélioré durant les années 1970, par A. Colmerauer et R.A. Kowalski, pour des travaux de recherche concernant la compréhension du langage naturel et la programmation en logique. Il a été de plus utilisé pour des recherches sur l'interrogation et la description des bases de données, la conception assistée par ordinateur et la réalisation de systèmes experts.

Le langage se distingue des autres langages en ce sens qu'il ne possède pas de distinction entre programmes et données, ou même vis-à-vis la structure de contrôle. Un programme écrit en Prolog ne fait qu'énoncer un ensemble de faits et de règles sur un domaine. Ces faits et ces règles sont par la suite exploités par un système de déduction, capable de répondre à toute question dont la réponse est logiquement déductible des connaissances fournies au préalable. Ceci est obtenu au cours d'une exploration de l'ensemble des cheminements logiques permettant de remonter de la question aux faits exprimés dans la base de connaissance.

La mécanique de base du langage Prolog a pour fonction de résoudre des équations, celles-ci ne portant pas sur des nombres, mais sur des arbres. Les arbres sont des structures permettant de représenter des informations complexes constituées de données élémentaires organisées hiérarchiquement. Prolog prend les arbres comme structures de données. Il a la dimension de permettre la définition d'une infinité d'arbres, à l'aide d'une collection finie de règles. Au cours de son travail de déduction, Prolog est amené à explorer un espace de recherche qui a lui-même une forme d'arbre.

2.0 UN PROGRAMME EN PROLOG

Prolog est un langage de programmation fait pour représenter et utiliser les connaissances que l'on a sur un certain domaine. Plus exactement le domaine est un ensemble d'objets et la connaissance est matérialisée par un ensemble de relations qui décriront à la fois les propriétés de ces objets et leurs interactions. Un ensemble de règles décrivant ces objets et ces relations constitue un programme Prolog.

Une règle en prolog est de la forme:

A si B et C et... et Z.

Elle possède une conclusion unique soit A, appelé tête de la règle, et zéro, une ou plusieurs conditions (ou prémisses) B, ..., Z appelé queue de la règle. Une règle sans prémisse s'appelle une assertion (ou fait) qui est une formule toujours vraie sinon c'est une règle. La conclusion est considérée comme vraie si l'ensemble des prémisses est vrai. Ce type particulier de prédicat logique est appelé "clause de Horn".

Si on définit une relation entre des objets, l'ordre dans lequel les objets sont donnés est significatif. On appelle "identificateurs" les mots qui servent aussi bien à nommer les objets que les relations qui les lient. Le nom de l'identificateur d'une relation est appelé "prédicat". Les objets sur lesquels portent la relation sont les arguments.

Le fait ou l'assertion suivante: Paul est le père de Jean, peut s'écrire en Prolog par:

père (Paul, Jean) ⟶;
ou "père" est l'identificateur de la relation et est le prédicat, "Paul" et "Jean" étant les arguments.

Voici un exemple de programme Prolog appelé "le programme":

```
"le programme"
père (Paul, Jean) ⟶;
père (Pierre, Mélanie) ⟶;
père (Jean, Yvan) ⟶;
mère (Mélanie, Robert) ⟶;
mère (Chantal, Mélanie) ⟶;

homme (Paul) ⟶;
homme (Jean) ⟶;
homme (Pierre) ⟶;
homme (Yvan) ⟶;
homme (Robert) ⟶;
femme (Mélanie) ⟶;
femme (Chantal) ⟶;

fils (x, y) ⟶ père (y, x) homme (x);
fils (x, y) ⟶ mère (y, x) homme (x);
fille (x, y) ⟶ père (y, x) femme (x);
fille (x, y) ⟶ mère (y, x) femme (x);
```

Ceci est un programme Prolog, il est composé d'un commentaire "le programme" et d'un ensemble de règles. Les commentaires sont placés entre guillemmets. Les règles définissent des relations qui introduisent à la fois les objets et leur classification.

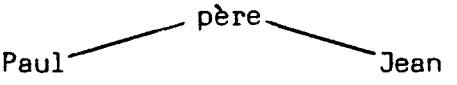
Une question telle que "de qui Paul est le père" sur l'ensemble précédent d'assertions se traduira, par exemple:

```
> père (Paul, e);
Prolog répond:  e = Jean
```

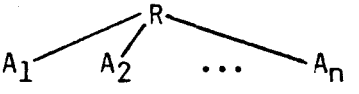
Une question ne comporte pas la flèche "⟶". La question précédente amène Prolog à répondre sur l'ensemble des objets que peut désigner la variable e pour que l'assertion soit vérifiée.

3.0 LA STRUCTURE D'ARBRE

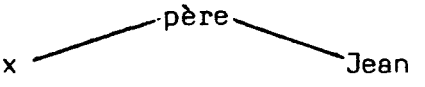
La structure de base des objets que Prolog sait manipuler est la structure d'arbres. Les relations et leurs arguments sont des arbres. C'est une structure suffisamment riche pour représenter des informations complexes, organisées hiérarchiquement, mais aussi assez simple à manipuler, aussi bien du point de vue algébrique que du point de vue informatique. Par exemple, l'assertion "père (Paul, Jean)" peut être représentée par l'arbre suivant:



Plus généralement, un arbre A est constitué d'un noeud R appelé racine, et d'un ensemble ordonné d'éléments A_1, ..., A_n, qui sont eux-mêmes des arbres et qu'on appelle fils de R: A_1, ..., A_n, sont des sous-arbres de A et tout sous-arbre A_i est aussi un sous-arbre de A. Les racines de A_1, ..., A_n, sont les descendants de R. Tout noeud sans descendant est dit terminal ou feuille:



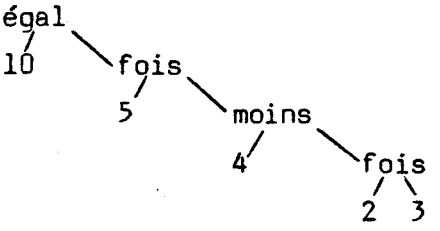
Souvent les structures d'arbre ne sont pas complètement connues; elles comportent des trous représentés par des variables. Les variables peuvent recevoir une valeur au cours de l'exécution du programme, cette valeur étant une structure arborescente quelconque:



en est un exemple.

Une information atomique est la forme d'arbre la plus élémentaire; l'objet associé est une constante, identificateur, chaîne de caractères ou un entier.

Les arbres sont représentés par des formules que nous appelons "termes". Un terme est construit à l'aide des informations atomiques (constantes, nom de fonctions, nom de relations, ...), des variables, des parenthèses, des virgules. Un terme permet de traduire la structure hiérarchisée de l'arbre:



Par exemple, l'arbre ci-dessus est représenté par le terme:

égal (10, fois (5, moins (4, fois (2, 3)))).

4.0 L'UNIFICATION

Un des mécanismes de base de toutes les implantations du langage, consiste à décider si deux arbres peuvent être rendus égaux ou non. Si oui, l'égalité est réalisée en affectant des valeurs aux variables figurant dans l'une ou l'autre des arbres donnés.

D'un point de vue plus général, on peut parler d'équations entre arbres ou même de systèmes d'équations entre arbres. La solution d'un tel système, si elle existe, sera un ensemble de valeurs à donner aux variables figurant dans le système d'équation pour que chacune de celle-ci soit satisfaite.

La notion d'inéquations entre arbres est un des concepts originaux de Prolog II. Les systèmes d'inéquations expriment que tels arbres doivent être différents entre eux.

On a tendance à regrouper systèmes d'équations et d'inéquations sous le terme de "contraintes".

L'interprétation de Prolog utilise un algorithme de réduction qui procède par simplifications successives de la contrainte associée. L'algorithme de réduction des contraintes d'égalité est appelé "unification".

L'unification repose sur les principes suivants: étant donné deux arbres a et a', on cherche si a = a' possède une solution dans le domaine des arbres. Pour cela, on décompose a et a' en leurs sous-arbres respectifs. Si les racines de a et a' sont identifques et ont le même nombre de descendants, alors la contrainte initiale $C_0 = (a = a')$ est remplacée par la contrainte $C_1 = (a_1 = a_1', \ldots, a_n = a_n')$ où a_i et a_i' sont des sous-arbres de a et a' respectivement. On procède de la même façon pour chacune des nouvelles contraintes obtenus en C_1. L'algorithme s'arrête lorsque la contrainte courante est réduite:

1) lorsque la contrainte courante contient une égalité entre deux arbres qui n'ont pas la même racine ou dont les racines n'ont pas le même nombre de descendants, dans ce cas le système n'a pas de solution;

2) lorsque la contrainte courante ne contient pas des égalités dont les membres gauches sont des variables toutes distinctes, auquel cas la contrainte courante représente la solution du système initial.

Dans le cas où la contrainte à satisfaire, C, contient des inéquations, c'est de la forme:

$C = E \cup (a_1 \neq a_1'; a_g \neq a_g'; \ldots; a_n \neq a_n')$ où E ne contient que des équations.

On a les propriétés suivantes:

a) $E \cup (a_1 \neq a_1'; a_g \neq a_g'; \ldots; a_n \neq a_n')$ a une solution si et seulement si chacune des contraintes $E \cup (a_1 \neq a_1') \ldots E \cup (a_n \neq a_n')$ a une solution.

b) Supposons que E ait une solution. Alors:

b1) si $(a_i \neq a_i')$ n'a pas de solution, $E \cup (a_i \neq a_i')$ en a une. Dans ce cas, l'inéquation $a_i \neq a_i'$ peut être supprimée et S est aussi solution de $E \cup (a_i \neq a_i)$;

b2) si E et $E \cup (a_i = a_i')$ ont la même solution, alors $E \cup (a_i \neq a_i)$ n'a pas de solution;

b3) si E et $E \cup (a_i = a_i')$ n'ont pas la même solution, $E \cup (a_i = a_i)$ a pour solution $S \cup (x_1 = v_1; \ldots; x_m = u_m)$ où les x_i sont des variables distinctes entre elles et distinctes de celles apparaissant en membre gauche de S. Alors $E \cup (a_i \neq a_i')$ est soluble et sa solution est déterminée par S et les inégalités $(x_i \neq v_i)$.

5.0 LE PRINCIPE D'EFFACEMENT DES BUTS

Programmer en Prolog consiste donc à formuler les faits sous forme d'assertions et à définir les règles logiques qui relient ces faits à d'autres. De même, exécuter un programme revient à demander la preuve d'une expression. Prolog se comporte comme un "démonstrateur de théorème". Il part du but (l'expression à prouver) et cherche à effacer toutes les conditions qui le compose. Il comporte donc un moteur d'inférence qui réalise cette résolution.

Pour répondre à une question, c'est-à-dire satisfaire une conjonction de relations représentée par une suite de termes, Prolog va tenter d'effacer ces buts un à un, dans l'ordre où ils se présentent. Le principe d'effacement permet d'interpréter questions et règles comme suit:

- une question comme: $q_1\ q_2\ \ldots\ q_n$ s'interprète par l'ordre, c'est-à-dire effacer q_1, puis effacer q_2, ..., puis effacer q_n;

- une règle comme: $p \longrightarrow q_1\ q_2\ \ldots\ q_n$ s'interprète par: pour effacer p, effacer q_1, puis effacer q_2, ..., puis effacer q_n;

- une règle comme: $P \longrightarrow$; s'interprète par: p s'efface.

À chaque pas, nous avons une suite courante de buts à effacer, B, et une contrainte courante, C. L'algorithme démarre en prenant pour valeur initiale de B, la suite initiale des buts (la question posée) et pour valeur initiale de C, la contrainte vide. Le moteur répète ce qui suit:

1) choix du but à effacer: si $B = b_1\ b_2\ \ldots\ b_n$, on choisit d'effacer b_1;

2) choix de la règle à appliquer: soit $P \longrightarrow q_1\ q_2\ \ldots\ q_m$, $(m >= o)$, la première règle du programme telle que (après avoir renommé ses variables) la contrainte $C \cup (P = b_1)$ ait une solution, soit C' sa forme réduite. On pose $B = q_1\ q_2\ \ldots\ q_m\ b_2\ b_3\ \ldots\ b_n$ et $C = C'$. La queue

de la règle a remplacé le but b, effacé, et la contrainte courante a été augmentée des conditions qui ont permis l'effacement de b_1;

3) on recommence en (1) sauf si:
   - la suite des buts B est vide. Cet état caractérise un succès et la réponse est la contrainte courante C;
   - on bloque sur un but b qu'on ne peut effacer parce qu'il n'y a pas de règle qui permette de le faire. C'est un échec.

Considérons à un instant donné t_e que nous sommes parmi un des deux cas du niveau (3), et soit b_1 b_2 ..., b_e la suite des buts restant à effacer. Posons C_{t_e} l'ensemble des choix des règles applicables mis en attente lors de l'effacement du but précédent, alors C_{t_e} représente la suite des choix de règles mis en attente dans l'ordre du plus ancien au plus récent.

Au niveau (3) du fonctionnement du moteur dans les deux cas décrits, le moteur ne s'arrête pas là: dans ces deux cas, le moteur effectue un "retour en arrière" ou "back tracking". Dans le but d'exploiter toutes les possibilités qui restent ouvertes parmi le choix de règles en attente, le moteur reconsidère dans l'ordre C_{t_e} ... C_{t_1} la règle P^1 qui suit la règle P ⟶ ... dans le programme. On procède ainsi tant qu'il reste des choix en attente. Il est important de remarquer que lorsqu'on effectue un nouveau choix pour tenter d'effacer un but b on "défait" toutes les affectations (les instanciations) des variables qui ont eu lieu entre le précédent choix pour effacer b et l'instant présent.

On montre en figure 1 le fonctionnement du moteur de Prolog.

On remarque que le fonctionnement du moteur d'inférence de Prolog est le "chaînage arrière" (ou raisonnement guidé par le but) et la stratégie de développement est la "profondeur d'abord". On remplace le but à prouver par des sous-buts résolus (primitifs) directement par les assertions soutenues dans la base. Le langage Prolog est strictement séquentiel, il tente de prouver les sous-buts dans l'ordre où ils ont été écrits. Les

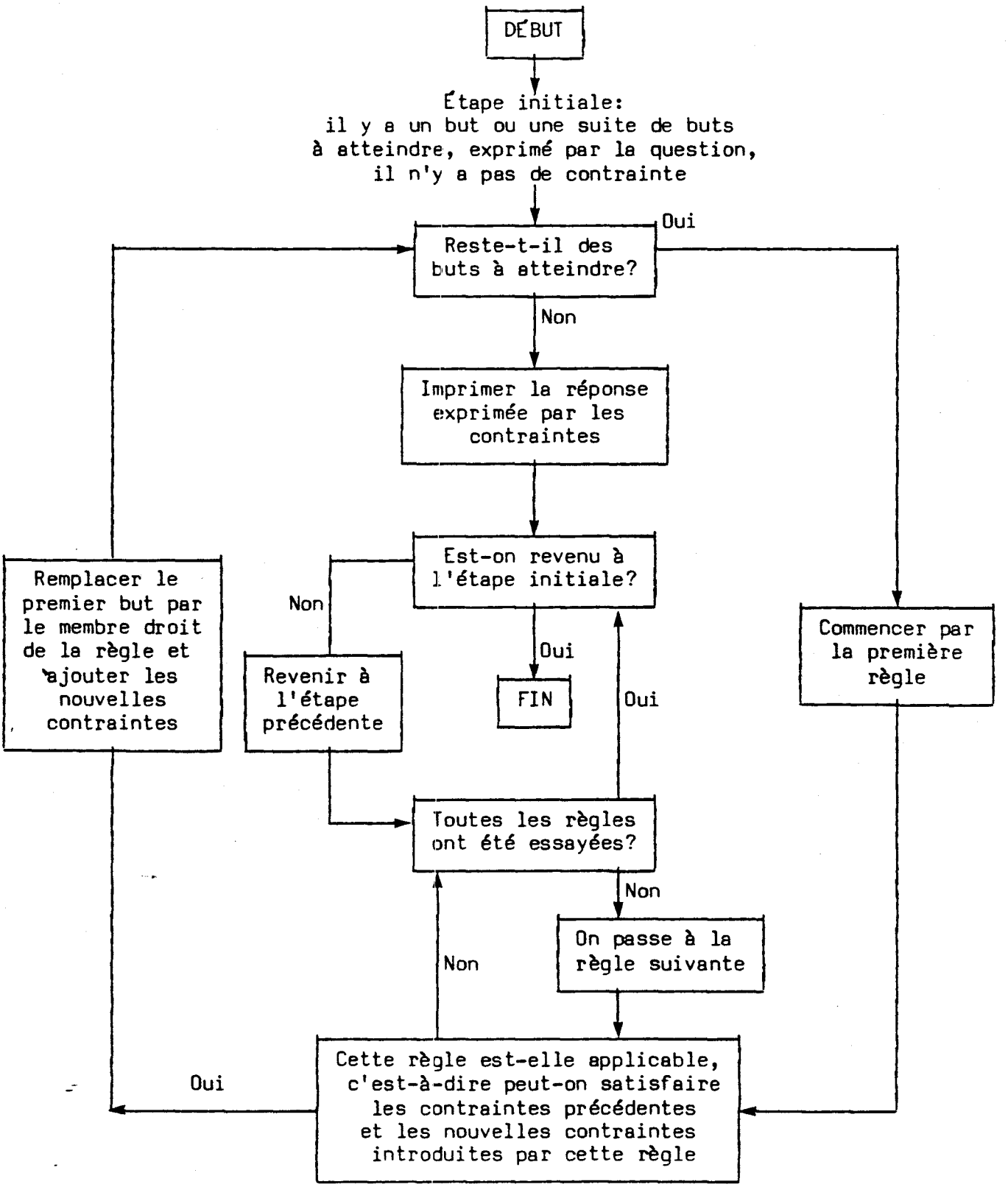


FIGURE 1

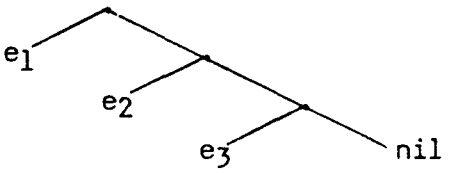
Fonctionnement du moteur d'inférence de PROLOG.

règles de Prolog donc, ne doivent pas être écrites dans n'importe quel ordre; par exemple, il faudra très souvent traiter les cas particuliers en premier. Si Prolog échoue dans la résolution d'un sous-but, il effectue un retour arrière. L'opération retour arrière est effectuée soit en cas d'échec ou soit pour trouver une autre solution, si l'arbre des essais possibles n'a pas été entièrement parcouru.

Les faits et les règles en Prolog admettent des variables. Lorsque ces variables prennent une valeur au cours de la résolution du problème, on dit que ces variables sont instanciées. Le mécanisme de base qui permet d'égaler un terme à un autre dans le but d'instancier les variables présentes dans chacun des deux termes s'appelle l'unification. L'unification est une procédure de base de l'interpréteur Prolog. Lorsqu'il essaie d'effacer un sous-but, il essaie d'unifier celui-ci avec une assertion existante dans la base ou avec une expression de tête de règle.

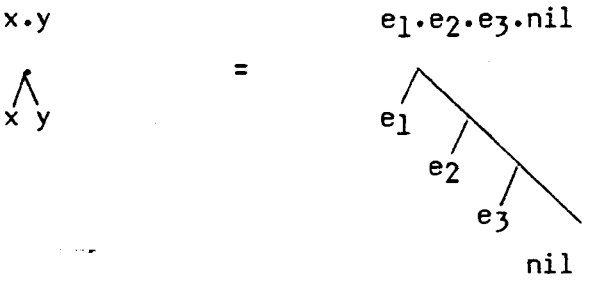
6.0 LES LISTES

À une suite ordonnée ou non, nous associons la liste de ses éléments. À la suite (e_1, e_2, ...) correspond la liste (e_1.(e_2.(...))). La liste vide se note (nil) et sert de marque de fin de liste. Cependant, le codage non parenthèsé est admis. Par exemple, la suite (e_1, e_2, e_3) s'écrit e_1.e_2.e_3.nil, et sa représentation sous forme d'arbre est:



Ces arbres, et par extension les listes qu'ils représentent, sont parfois appelés des peignes.

Si on veut résoudre l'équation x.y = e_1.e_2.e_3.nil ou bien sous forme d'arbre l'égalité suivante:



on a par identification la solution:

x = e_1 , y = e_2.e_3.nil.

La notaton x . y représente une liste dont la tête, c'est-à-dire le premier élément, est x et la queue (le reste de la liste) est y. Cette propriété est à la base de l'utilisation de listes dans les programmes Prolog.

L'accès aux éléments d'une liste se fait à l'aide des deux règles suivantes:

R_1 élément de (x, x.y) $\longrightarrow$;
R_2 élément de (c, u.y) $\longrightarrow$; élément de (x, y).

On interprète ces deux règles de la façon suivante:

R_1: x est élément de toute liste qui commence par x.
R_2: x est élément de toute liste dont la queue est y, si y est élément de y.

Le second membre de R_2 provoque un nouvel appel de R_1 et R_2. C'est ce que l'on nomme la "récursivité". La plupart des problèmes sur les listes sont résolus par des règles mettant en jeu la récursivité. Lorsque nous utilisons le programme pour vérifier que e_2 par exemple, par la question >élément de ($e_2.e_1.e_2.e_3$.nil) e_2 par exemple, appartient à la liste $e_1.e_2.e_3$.nil, la relance par R_2 avec une liste plus courte a pour effet d'éliminer un à un les éléments de tête, jusqu'au moment où l'élément recherché se trouve en tête de liste. R_1 peut alors s'appliquer et bloque le déroulement. Dans la remontée, R_2 se déclenche à nouveau, et continue jusqu'à l'effacement du terme élément de (x, nil) qui conduit à un système insoluble. Le déroulement s'arrête, il y a <u>arrêt par défaut</u>. Dans le cas de l'énumération des éléments d'une liste, par exemple par la question > élément de (x, $e_1.e_2.e_3$.nil), R_1 s'applique à chaque tour et stoppe la boucle. Le même processus se répète dans la remontée jusqu'à l'arrêt par défaut.

Ainsi notre examen met en évidence dans un programme récursif deux types de règles et deux types d'arrêts:

- une ou plusieurs règles provoquent la récursivité, généralement avec des données "plus simples" et assurent le déroulement de la boucle. Dans notre cas c'est le rôle de R_2;

- une ou plusieurs règles stoppent la boucle. Ici c'est R_1 qui s'en charge;

- un arrêt explicite. Par exemple dans R_1, l'identité entre l'élément cherché et la tête de la liste fournie, ce qu'exprime la notation x et x.y;

- un arrêt implicite. En particulier par la rencontre d'un terme impossible à effacer.

Voici maintenant quelques exemples de manipulation de listes:

1) construction d'une liste en prenant les éléments impairs d'une autre liste:

R_1 éléments impairs (nil, nil) ⟶;
R_2 éléments impairs (u.V.y, u.m) ⟶
éléments impairs (y, m);

2) construction d'une liste en prenant les éléments pairs d'une autre liste:

R_1 éléments pairs (nil, nil) ⟶;
R_2 éléments pairs (u.V.y, u.m) ⟶
éléments pairs (y, m);

3) inversion de l'ordre d'une liste d au moyen d'une liste auxiliaire:

R_1 renverser (nil, e, l) ⟶;
R_2 renverser (x.y, l, r) ⟶
renverser (y, x.l);
où r est la liste d inversée

4) la concaténation de deux listes:

R_1 concatener (nil, l, l) ⟶;

R_2 concatener (x.y, l, x.z) ⟶
concatener (y, l, z);

soit deux listes l_1 et l_2, posons l_3 la concatenateur de l_1 et l_2. L'opération de concatenation se ramène au transfert des éléments de la liste l_1 et la durée de l'exécution est proportionnelle au nombre d'éléments de cette liste. Le transfert de l_2 vers l_3 se fait globalement, par passage final de paramètre.

7.0 UN EXEMPLE D'APPLICATION SUR LE CAS DES DÉVERSEMENTS ACCIDENTELS DE CONTAMINANTS

Une maquette de système expert fut développée à partir du document intitulé "Données de base pour le développement d'un système expert concernant les techniques d'intervention lors d'un déversement accidentel de contaminants."

Ce document fut préparé par André Bériault et Jean-Claude Tessier, tous deux du Service de la recherche de la direction Environnement d'Hydro-Québec.

Vous retrouverez en fin de l'annexe le listing du programme de la maquette. Le programme est écrit en turbo-prolog, la version la plus récente de Prolog (sur le marché depuis le début de l'été 1986). Cette version est disponible sur micro-ordinateur P.C. et compatibles.

Vous remarquerez que la syntaxe pour l'écriture des faits et des règles est un peu différente de celle utilisée jusqu'ici pour montrer quelques exemples de clauses Prolog. La syntaxe que nous avons utilisée auparavant est la Prolog II. La syntaxe de Prolog II est quelque peu différente de Turbo-Prolog. Cependant, le fonctionnement du moteur d'inférence à quelques particularités près est le même que Prolog II. Le lecteur pourra s'en convaincre en consultant "Turbo-Prolog: The Natural Language of Artificial Intelligence", édité par Borland International Inc., 4585, Scotts Valley Drive, Scotts Valley, CA., U.S.A.

L'usager peut consulter le système expert après une mise en mémoire du programme, et en donnant comme but à satisfaire celui de "intervention". Le système demandera alors de définir l'état du déversement sur le site par une série de questions-réponses.

Après que l'état du ~~document~~ déversement est bien défini, le système suggère la ou les (s'il y a en a plus d'une possible) intervention(s) appropriée(s).

Chacune des interventions suggérées est divisée en deux parties: intervention de confinement et intervention de récupération. Pour chacune des parties de l'intervention, on donne à titre d'exemple sur un tableau, un coût fictif de l'opération, un facteur de réussite, un coût pondéré en tenant compte du facteur de réussite.

Un exemple de sortie d'une consultation est donné juste après le listing du programme. Cet exemple titré "exemple de consultation" correspond à l'état du déversement sur le site qui est le suivant:

1) la tache est répandue sur le sol;
2) la tache est répandue dans une aire d'entreposage;
3) la tache atteint un regard d'égout;
4) la tache est de forme allongée;
5) il y a un risque élevé de contamination de la nappe phréatique;
6) le sol est non gelé;
7) il y a eu peu de temps écoulé depuis le déversement;
8) la quantité répandue est importante;
9) le produit répandu est de l'essence.

```
          /* deverse.pro */
code=4096
  domains
      integerlist = integer*
      file=deversement
  database
      xoui(symbol,integer)
  predicates
     intervention
     intervention1
     intervention2
     verification(string)
     premiere_mesure(string)
     confinement(real,real,real,string)
     recuperation(real,real,real,string)
     oui(symbol,integer)
     enlever_fait
     question(symbol,integer)
     membre(integer,integerlist)
     sortie(integer)

  clauses
     intervention:-
          openwrite(deversement,"a:deverse.dat"),
          verification(A),
          write(A),nl,
          writedevice(deversement),
          write(A),nl,
          writedevice(printer),
          write(A),nl,nl,
          writedevice(screen),
          premiere_mesure(B),
          write(B),nl,
          writedevice(deversement),
          write(B),nl,
          writedevice(printer),
          write(B),nl,nl,
          writedevice(screen),
          intervention1.

     intervention1:-
          intervention2.
     intervention1:-
          write("\nIncapable de determiner d'autres"),
          write(" interventions appropriees. \n\n"),
          writedevice(printer),
          write("-----------------------------------------------------"),nl,
          writedevice(screen),
          closefile(deversement),
          nl,nl,enlever_fait.


     intervention2:-
          confinement(Cc,Prc,Ccp,D),
          recuperation(Cr,Prr,Crp,E),
          write(D),nl,
          write(E),nl,
          write("                    cout    % reussite    cout pondere"),nl,nl,
          writef("%15%10%10%17 \n","confinement:",Cc,Prc,Ccp),nl,
          writef("%15%10%10%17 \n","recuperation:",Cr,Prr,Crp),nl,
```

```
      write("                                          ----------"),nl,
      Ctp=Ccp + Crp ,
      write("                                             ",Ctp),nl,nl,
      writedevice(deversement),
      write(D),nl,
      write(E),nl,
      write("                    cout    % reussite    cout pondere"),nl,nl,
      writef("%15%10%10%17 \n","confinement:",Cc,Prc,Ccp),nl,
      writef("%15%10%10%17 \n","recuperation:",Cr,Prr,Crp),nl,
      write("                                          ----------"),nl,
      Ctp=Ccp + Crp ,
      write("                                             ",Ctp),nl,nl,
      writedevice(printer),
      write(D),nl,
      write(E),nl,
      write("                    cout    % reussite    cout pondere"),nl,nl,
      writef("%15%10%10%17 \n","confinement:",Cc,Prc,Ccp),nl,
      writef("%15%10%10%17 \n","recuperation:",Cr,Prr,Crp),nl,
      write("                                          ----------"),nl,
      Ctp=Ccp + Crp ,
      write("                                             ",Ctp),nl,nl,
      writedevice(screen),
      nl,nl,nl,fail.


enlever_fait:-
      retract(xoui(_,_)),fail.

enlever_fait:-
      write("\n\nPlease press the space bar to Exit"),
      readchar(_).

membre(Int,[Int|_]).
membre(Int,[_|Lint]):- membre(Int,Lint).

oui(X,Y) if xoui(X,Y),!.
oui(X,Y) if
      not(xoui(X,_)),
      question(X,Z),
      Y = Z .

sortie(Z):-
      writedevice(deversement),
      write(Z),nl,nl,nl,
      writedevice(printer),
      write(Z),nl,nl,nl,
      writedevice(screen).


question(reseau,Z):-
      write("Est-ce que la tache:"),nl,
      write("     1-est sur le sol"),nl,
      write("     2-est sur l'eau"),nl,
      write("     3-est sur le plancher"),nl,
      write("       dans un batiment","\n"),
      writedevice(deversement),
      write("Est-ce que la tache:"),nl,
      write("     1-est sur le sol"),nl,
      write("     2-est sur l'eau"),nl,
      write("     3-est sur le plancher"),nl,
```

```
      write("      dans un batiment","\n"),
      writedevice(printer),
      write("Est-ce que la tache:"),nl,
      write("     1-est sur le sol"),nl,
      write("     2-est sur l'eau"),nl,
      write("     3-est sur le plancher"),nl,
      write("       dans un batiment","\n"),
      writedevice(screen),
      readint(Z),
      sortie(Z),
      asserta(xoui(reseau,Z)).

question(interieur,Z):-
      write("Est-ce que la tache a ete repandue:"),nl,
      write("     1-dans un garage,"),nl,
      write("       dans un atelier,"),nl,
      write("       autres batiments"),nl,
      write("     2-dans un batiment"),nl,
      write("       d'un poste"),nl,
      write("     3-dans une centrale","\n"),
      writedevice(deversement),
      write("Est-ce que la tache a ete repandue:"),nl,
      write("     1-dans un garage,"),nl,
      write("       dans un atelier,"),nl,
      write("       autres batiments"),nl,
      write("     2-dans un batiment"),nl,
      write("       d'un poste"),nl,
      write("     3-dans une centrale","\n"),
      writedevice(printer),
      write("Est-ce que la tache a ete repandue:"),nl,
      write("     1-dans un garage,"),nl,
      write("       dans un atelier,"),nl,
      write("       autres batiments"),nl,
      write("     2-dans un batiment"),nl,
      write("       d'un poste"),nl,
      write("     3-dans une centrale","\n"),
      writedevice(screen),
      readint(Z),
      sortie(Z),
      asserta(xoui(interieur,Z)).

question(exterieur,Z):-
      write("Est-ce que la tache a ete repandue:"),nl,
      write("     1-dans une aire"),nl,
      write("       d'entreposage"),nl,
      write("       ou autres aires"),nl,
      write("     2-a l'exterieur,mais a"),nl,
      write("       l'interieur deslimites"),nl,
      write("       de l'installation d'un poste"),nl,
      write("     3-a l'exterieur des limites"),nl,
      write("       d'une installation","\n"),
      writedevice(deversement),
      write("Est-ce que la tache a ete repandue:"),nl,
      write("     1-dans une aire"),nl,
      write("       d'entreposage"),nl,
      write("       ou autres aires"),nl,
      write("     2-a l'exterieur,mais a"),nl,
      write("       l'interieur deslimites"),nl,
      write("       de l'installation d'un poste"),nl,
      write("     3-a l'exterieur des limites"),nl,
```

```
      write("      d'une installation","\n"),
      writedevice(printer),
      write("Est-ce que la tache a ete repandue:"),nl,
      write("    1-dans une aire"),nl,
      write("      d'entreposage"),nl,
      write("      ou autres aires"),nl,
      write("    2-a l'exterieur,mais a"),nl,
      write("      l'interieur deslimites"),nl,
      write("      de l'installation d'un poste"),nl,
      write("    3-a l'exterieur des limites"),nl,
      write("      d'une installation","\n"),
      writedevice(screen),
      readint(Z),
      sortie(Z),
      asserta(xoui(exterieur,Z)).

question(tache1,Z):-
      write("Est-ce que la tache atteint:"),nl,
      write("    1-un regard d'egout"),nl,
      write("    2-un puit d'acces aux cables et au reseau de drainage"),nl,
      write("    3-des conduites de cables"),nl,
      write("    4-des drains de surfaces"),nl,
      write("      ou souterrains"),nl,
      write("    5-une station de pompage"),nl,
      write("    6-aucunes de ces possibilites","\n"),
      writedevice(deversement),
      write("Est-ce que la tache atteint:"),nl,
      write("    1-un regard d'egout"),nl,
      write("    2-un puit d'acces aux cables et au reseau de drainage"),nl,
      write("    3-des conduites de cables"),nl,
      write("    4-des drains de surfaces"),nl,
      write("      ou souterrains"),nl,
      write("    5-une station de pompage"),nl,
      write("    6-aucunes de ces possibilites","\n"),
      writedevice(printer),
      write("Est-ce que la tache atteint:"),nl,
      write("    1-un regard d'egout"),nl,
      write("    2-un puit d'acces aux cables et au reseau de drainage"),nl,
      write("    3-des conduites de cables"),nl,
      write("    4-des drains de surfaces"),nl,
      write("      ou souterrains"),nl,
      write("    5-une station de pompage"),nl,
      write("    6-aucunes de ces possibilites","\n"),
      writedevice(screen),
      readint(Z),
      sortie(Z),
      asserta(xoui(tache1,Z)).

question(tache2,Z):-
      write("Est-ce que la tache atteint:"),nl,
      write("    1-un fosse sec"),nl,
      write("    2-un fosse humide"),nl,
      write("    3-un fosse humide etroit"),nl,
      write("    4-un ponceau"),nl,
      write("    5-aucunes de ces possibilites","\n"),
      writedevice(deversement),
      write("Est-ce que la tache atteint:"),nl,
      write("    1-un fosse sec"),nl,
      write("    2-un fosse humide"),nl,
      write("    3-un fosse humide etroit"),nl,
```

```
        write("     4-un ponceau"),nl,
        write("     5-aucunes de ces possibilites","\n"),
        writedevice(printer),
        write("Est-ce que la tache atteint:"),nl,
        write("     1-un fosse sec"),nl,
        write("     2-un fosse humide"),nl,
        write("     3-un fosse humide etroit"),nl,
        write("     4-un ponceau"),nl,
        write("     5-aucunes de ces possibilites","\n"),
        writedevice(screen),
        readint(Z),
        sortie(Z),
        asserta(xoui(tache2,Z)).

question(tache3,Z):-
        write("Est-ce que la tache atteint:"),nl,
        write("     1-une station de pompage","\n"),
        write("     2-un ruisseau"),nl,
        write("     3-une riviere"),nl,
        write("     4-un lac"),nl,
        write("     5-le fleuve"),nl,
        write("     6-aucunes de ces possibilites","\n"),
        writedevice(deversement),
        write("Est-ce que la tache atteint:"),nl,
        write("     1-une station de pompage","\n"),
        write("     2-un ruisseau"),nl,
        write("     3-une riviere"),nl,
        write("     4-un lac"),nl,
        write("     5-le fleuve"),nl,
        write("     6-aucunes de ces possibilites","\n"),
        writedevice(printer),
        write("Est-ce que la tache atteint:"),nl,
        write("     1-une station de pompage","\n"),
        write("     2-un ruisseau"),nl,
        write("     3-une riviere"),nl,
        write("     4-un lac"),nl,
        write("     5-le fleuve"),nl,
        write("     6-aucunes de ces possibilites","\n"),
        writedevice(screen),
        readint(Z),
        sortie(Z),
        asserta(xoui(tache3,Z)).

question(polluant,Z):-
        write("Est-ce que le polluant est:"),nl,
        write("     1-de l'essence"),nl,
        write("     2-de l'huile ou autres produits","\n"),
        writedevice(deversement),
        write("Est-ce que le polluant est:"),nl,
        write("     1-de l'essence"),nl,
        write("     2-de l'huile ou autres produits","\n"),
        writedevice(printer),
        write("Est-ce que le polluant est:"),nl,
        write("     1-de l'essence"),nl,
        write("     2-de l'huile ou autres produits","\n"),
        writedevice(screen),
        readint(Z),
        sortie(Z),
        asserta(xoui(polluant,Z)).
```

```
question(quantite,Z):-
      write("Est-ce que la quantite est:"),nl,
      write("     1-faible"),nl,
      write("     2-moyenne"),nl,
      write("     3-importante","\n"),
      writedevice(deversement),
      write("Est-ce que la quantite est:"),nl,
      write("     1-faible"),nl,
      write("     2-moyenne"),nl,
      write("     3-importante","\n"),
      writedevice(printer),
      write("Est-ce que la quantite est:"),nl,
      write("     1-faible"),nl,
      write("     2-moyenne"),nl,
      write("     3-importante","\n"),
      writedevice(screen),
      readint(Z),
      sortie(Z),
      asserta(xoui(quantite,Z)).

question(forme_tache,Z):-
      write("Est-ce que la tache est:"),nl,
      write("     1-circulaire"),nl,
      write("     2-allonge"),nl,
      write("     3-quelquonque","\n"),
      writedevice(deversement),
      write("Est-ce que la tache est:"),nl,
      write("     1-circulaire"),nl,
      write("     2-allonge"),nl,
      write("     3-quelquonque","\n"),
      writedevice(printer),
      write("Est-ce que la tache est:"),nl,
      write("     1-circulaire"),nl,
      write("     2-allonge"),nl,
      write("     3-quelquonque","\n"),
      writedevice(screen),
      readint(Z),
      sortie(Z),
      asserta(xoui(forme_tache,Z)).

question(condition_climatl,Z):-
      write("La condition au sol est:"),nl,
      write("     1-sol non-gele"),nl,
      write("     2-sol gele"),nl,
      write("     3-couvert de neige"),nl,
      write("     4-fonte de neige","\n"),
      writedevice(deversement),
      write("La condition au sol est:"),nl,
      write("     1-sol non-gele"),nl,
      write("     2-sol gele"),nl,
      write("     3-couvert de neige"),nl,
      write("     4-fonte de neige","\n"),
      writedevice(printer),
      write("La condition au sol est:"),nl,
      write("     1-sol non-gele"),nl,
      write("     2-sol gele"),nl,
      write("     3-couvert de neige"),nl,
      write("     4-fonte de neige","\n"),
      writedevice(screen),
      readint(Z),
```

```
      sortie(Z),
      asserta(xoui(condition_climat1,Z)).

question(condition_climat2,Z):-
      write("La condition sur l'eau est:"),nl,
      write("     1-couvert de glace"),nl,
      write("     2-couvert de glace par endroit"),nl,
      write("     3-sans couvert de glace","\n"),
      writedevice(deversement),
      write("La condition sur l'eau est:"),nl,
      write("     1-couvert de glace"),nl,
      write("     2-couvert de glace par endroit"),nl,
      write("     3-sans couvert de glace","\n"),
      writedevice(printer),
      write("La condition sur l'eau est:"),nl,
      write("     1-couvert de glace"),nl,
      write("     2-couvert de glace par endroit"),nl,
      write("     3-sans couvert de glace","\n"),
      writedevice(screen),
      readint(Z),
      sortie(Z),
      asserta(xoui(condition_climat2,Z)).

question(temps_ecoule,Z):-
      write("Est-ce que le temps ecoule est:"),nl,
      write("     1-peu"),nl,
      write("     2-important","\n"),
      writedevice(deversement),
      write("Est-ce que le temps ecoule est:"),nl,
      write("     1-peu"),nl,
      write("     2-important","\n"),
      writedevice(printer),
      write("Est-ce que le temps ecoule est:"),nl,
      write("     1-peu"),nl,
      write("     2-important","\n"),
      writedevice(screen),
      readint(Z),
      sortie(Z),
      asserta(xoui(temps_ecoule,Z)).

question(risque,Z):-
      write("Est-ce que le risque de contamination de la nappe phreatique est:"),nl,
      write("     1-eleve"),nl,
      write("     2-faible","\n"),
      writedevice(deversement),
      write("Est-ce que le risque de contamination de la nappe phreatique est:"),nl,
      write("     1-eleve"),nl,
      write("     2-faible","\n"),
      writedevice(printer),
      write("Est-ce que le risque de contamination de la nappe phreatique est:"),nl,
      write("     1-eleve"),nl,
      write("     2-faible","\n"),
      writedevice(screen),
      readint(Z),
      sortie(Z),
      asserta(xoui(risque,Z)).

verification("consulter le plan de drainage de l'installation,verifier la presence de contaminant dans le reseau"):-
      oui(reseau,1),
      oui(exterieur,E),
```

```
      membre(E,[1,2]).
verification("consulter le plan de drainage de l'installation,verifier la presence de contaminant dans le reseau"):-
      oui(reseau,3),
      oui(interieur,1),
      membre(1,[2,3]).
verification(" ").

premiere_mesure("blocage du regard d'egout"):-
      oui(reseau,1),
      oui(exterieur,E),
      membre(E,[1,2]),
      oui(tache1,1).
premiere_mesure("blocage du puit d'acces aux cables"):-
      oui(reseau,1),
      oui(exterieur,E),
      membre(E,[1,2]),
      oui(tache1,2).
premiere_mesure("referer a une methode de blocage des drains de surface ou souterrains"):-
      oui(reseau,1),
      oui(exterieur,E),
      membre(E,[1,2]),
      oui(tache1,4).
premiere_mesure("mesure d'urgence speciale depandant du type de cables,cas extreme:sonnette d'alarme"):-
      oui(reseau,1),
      oui(exterieur,E),
      membre(E,[1,2]),
      oui(tache1,3).
premiere_mesure("arreter la station de pompage"):-
      oui(reseau,1),
      oui(exterieur,E),
      membre(E,[1,2]),
      oui(tache1,5).

premiere_mesure("blocage du regard d'egout"):-
      oui(reseau,3),
      oui(interieur,E),
      membre(E,[2,3]),
      oui(tache1,1).
premiere_mesure("blocage du puit d'acces aux cables"):-
      oui(reseau,3),
      oui(interieur,E),
      membre(E,[2,3]),
      oui(tache1,2).
premiere_mesure("referer a une methode de blocage des drains de surface ou souterrains"):-
      oui(reseau,3),
      oui(interieur,E),
      membre(E,[2,3]),
      oui(tache1,4).
premiere_mesure("mesure d'urgence speciale depandant du type de cables,cas extreme:sonnette d'alarme"):-
      oui(reseau,3),
      oui(interieur,E),
      membre(E,[2,3]),
      oui(tache1,3).
premiere_mesure("arreter la station de pompage"):-
      oui(reseau,3),
      oui(interieur,E),
      membre(E,[2,3]),
      oui(tache1,5).
premiere_mesure(" ").
```

```
confinement(12000,75,16000,"tranchee drainante,enlever le sol contamine"):-
        oui(reseau,1),
        oui(forme_tache,F),
        membre(F,[2,3]),
        oui(risque,1),
        oui(condition_climat1,1),
        oui(temps_ecoule,1).
confinement(15000,60,25000,"tranchee drainante,et puit de pompage"):-
        oui(reseau,1),
        oui(forme_tache,F),
        membre(F,[2,3]),
        oui(risque,1),
        oui(condition_climat1,1),
        oui(temps_ecoule,2).
confinement(10000,80,12500,"digue en terre"):-
        oui(reseau,1),
        oui(risque,2),
        oui(condition_climat1,1),
        oui(quantite,Q),
        membre(Q,[2,3]).
confinement(4000,80,5000,"enlever le sol souille"):-
        oui(reseau,1),
        oui(risque,2),
        oui(condition_climat1,1),
        oui(quantite,1).
confinement(21000,75,28000,"puit de recuperation avec rigoles"):-
        oui(reseau,1),
        oui(forme_tache,1),
        oui(risque,1),
        oui(condition_climat1,1),
        oui(quantite,Q),
        membre(Q,[1,2,3]),
        oui(temps_ecoule,1).
confinement(15000,60,25000,"puit de recuperation"):-
        oui(reseau,1),
        oui(forme_tache,1),
        oui(risque,1),
        oui(condition_climat1,1),
        oui(quantite,Q),
        membre(Q,[1,2,3]),
        oui(temps_ecoule,2).
confinement(18000,60,30000,"fosse d'interception avec rigoles"):-
        oui(reseau,1),
        oui(forme_tache,2),
        oui(risque,1),
        oui(condition_climat1,1),
        oui(quantite,Q),
        membre(Q,[2,3]),
        oui(temps_ecoule,1).
confinement(2000,50,4000,"melange d'absorbants avec le sol,arroser la tache pourrecuperer le contaminant a la surface"):-
        oui(reseau,1),
        oui(forme_tache,F),
        membre(F,[2,3]),
        oui(risque,2),
        oui(condition_climat1,1),
        oui(quantite,Q),
        membre(Q,[2,3]),
        oui(temps_ecoule,1).
confinement(5000,50,10000,"digue en neige"):-
        oui(reseau,1),
```

```
        oui(condition_climat1,3),
        oui(quantite,Q),
        membre(Q,[2,3]),
        oui(temps_ecoule,1).
confinement(1000,80,1250,"blocage du ponceau"):-
        oui(reseau,R),
        membre(R,[1,2]),
        oui(tache2,4).
confinement(3000,60,5000,"digue en neige"):-
        oui(reseau,R),
        membre(R,[1,2]),
        oui(tache2,1),
        oui(condition_climat1,3).
confinement(10000,80,12500,"digue en terre et en tout venant"):-
        oui(reseau,R),
        membre(R,[1,2]),
        oui(tache2,1),
        oui(condition_climat1,1),
        oui(quantite,Q),
        membre(Q,[1,2]).
confinement(12000,60,20000,"bille de bois"):-
        oui(reseau,2),
        oui(tache3,T),
        membre(T,[3,5]),
        oui(condition_climat2,3),
        oui(quantite,1).
confinement(6000,80,7500,"boudins absorbants"):-
        oui(reseau,2),
        oui(tache2,T),
        membre(T,[2,3]),
        oui(condition_climat2,3),
        oui(quantite,1).
confinement(6000,80,7500,"boudins absorbants"):-
        oui(reseau,2),
        oui(tache3,2),
        oui(condition_climat2,3),
        oui(quantite,1).
confinement(5000,50,10000,"barriere ou cloture"):-
        oui(reseau,2),
        oui(tache3,3),
        oui(condition_climat2,3),
        oui(quantite,1).
confinement(5000,50,10000,"bille de bois:rapide mais non efficace"):-
        oui(reseau,2),
        oui(tache3,T),
        membre(T,[3,5]),
        oui(condition_climat2,3),
        oui(quantite,Q),
        membre(Q,[2,3]).
confinement(1000,80,1250,"blocage du ponceau"):-
        oui(reseau,2),
        oui(tache2,T),
        membre(T,[2,3]),
        oui(condition_climat2,3),
        oui(quantite,Q),
        membre(Q,[2,3]).
confinement(1000,80,1250,"blocage du ponceau"):-
        oui(reseau,2),
        oui(tache3,2),
        oui(condition_climat2,3),
```

```
        oui(quantite,Q),
        membre(Q,[2,3]).
confinement(30000,80,37500,"digue en terre percee de conduite"):-
        oui(reseau,2),
        oui(tache2,T),
        membre(T,[2,3]),
        oui(condition_climat2,3),
        oui(quantite,Q),
        membre(Q,[2,3]).
confinement(30000,80,37500,"digue en terre percee de conduite"):-
        oui(reseau,2),
        oui(tache3,T),
        membre(T,[2,3]),
        oui(condition_climat2,3),
        oui(quantite,Q),
        membre(Q,[2,3]).
confinement(12000,75,16000,"deversoir en bois"):-
        oui(reseau,2),
        oui(tache2,T),
        membre(T,[2,3]),
        oui(condition_climat2,3),
        oui(quantite,Q),
        membre(Q,[2,3]).
confinement(60000,80,75000,"barrage en bois percee de conduites"):-
        oui(reseau,2),
        oui(tache2,T),
        membre(T,[2,3]),
        oui(condition_climat2,3),
        oui(quantite,Q),
        membre(Q,[2,3]).
confinement(60000,80,75000,"barrage en bois percee de conduites"):-
        oui(reseau,2),
        oui(tache3,2),
        oui(condition_climat2,3),
        oui(quantite,Q),
        membre(Q,[2,3]).
confinement(40000,80,50000,"digue avec conduite munie d'une valve de debit"):-
        oui(reseau,2),
        oui(tache3,2),
        oui(condition_climat2,3),
        oui(quantite,Q),
        membre(Q,[2,3]).
confinement(40000,80,50000,"digue avec conduite munie d'une valve de debit"):-
        oui(reseau,2),
        oui(tache2,T),
        membre(T,[2,3]),
        oui(condition_climat2,3),
        oui(quantite,Q),
        membre(Q,[2,3]).
confinement(5000,50,10000,"cloture ou barriere"):-
        oui(reseau,2),
        oui(tache3,3),
        oui(condition_climat2,3),
        oui(quantite,Q),
        membre(Q,[2,3]).
confinement(6000,80,9000,"boudins absorbants"):-
        oui(reseau,2),
        oui(tache2,T),
        membre(T,[2,3]),
        oui(condition_climat2,3),
```

```
        oui(quantite,Q),
        membre(Q,[2,3]).
confinement(6000,80,9000,"boudins absorbants"):-
        oui(reseau,2),
        oui(tache3,2),
        oui(condition_climat2,3),
        oui(quantite,Q),
        membre(Q,[2,3]).
confinement(8000,80,10000,"tranchee dans la glace"):-
        oui(reseau,2),
        oui(tache3,T),
        membre(T,[3,4]),
        oui(condition_climat2,1).
confinement(7000,70,10000,"barriere de contreplaque dans la glace"):-
        oui(reseau,2),
        oui(tache3,T),
        membre(T,[3,4]),
        oui(condition_climat2,1).
confinement(12000,60,20000,"bille de bois"):-
        oui(reseau,2),
        oui(tache3,T),
        membre(T,[3,5]),
        oui(condition_climat2,2).
confinement(8000,75,10000,"barriere en bois avec barils"):-
        oui(reseau,2),
        oui(tache3,T),
        membre(T,[3,5]),
        oui(condition_climat2,2).
confinement(6000,80,7500,"tapis de matiere absorbante enroulee dans la broche a poule"):-
        oui(reseau,2),
        oui(tache3,T),
        membre(T,[3,4]),
        oui(condition_climat2,2).
confinement(30000,80,37500,"digue en terre percee de conduites"):-
        oui(reseau,2),
        oui(tache3,T),
        membre(T,[2,3]),
        oui(condition_climat2,2).
confinement(30000,80,37500,"digue en terre percee de conduites"):-
        oui(reseau,2),
        oui(tache2,T),
        membre(T,[2,3]),
        oui(condition_climat2,2).
confinement(5000,50,10000,"barriere ou cloture"):-
        oui(reseau,2),
        oui(tache3,3),
        oui(condition_climat2,2).
confinement(60000,80,75000,"barrage en bois perce de conduites"):-
        oui(reseau,2),
        oui(tache2,T),
        membre(T,[2,3]),
        oui(condition_climat2,2).
confinement(60000,80,75000,"barrage en bois perce de conduites"):-
        oui(reseau,2),
        oui(tache3,2),
        oui(condition_climat2,2).
confinement(40000,80,50000,"digue avec conduite munie d'une valve de debit"):-
        oui(reseau,2),
        oui(tache2,T),
        membre(T,[2,3]),
```

```
      oui(condition_climat2,2).
confinement(40000,80,50000,"digue avec conduite munie d'une valve de debit"):-
      oui(reseau,2),
      oui(tache3,2),
      oui(condition_climat2,2).
confinement(30000,60,50000,"barrieres commerciales"):-
      oui(reseau,2),
      oui(tache3,3),
      oui(condition_climat2,2).
confinement(3000,75,4000,"enlever le sol contaminer et la neige souillee"):-
      oui(reseau,1),
      oui(condition_climat1,C),
      membre(C,[2,3]).
confinement(8000,80,10000,"cloture avec membrane impermeable,enlever le sol contaminer et la neige souillee"):-
      oui(reseau,1),
      oui(condition_climat1,4).
confinement(100,100,100,"arreter la station de pompage"):-
      oui(reseau,2),
      oui(tache3,1).
confinement(100,100,100,"blocage des drains"):-
      oui(reseau,3),
      oui(interieur,1).
confinement(100,100,100,"  "):-
      oui(reseau,3),
      oui(interieur,I),
      membre(I,[2,3]).

recuperation(2000,80,2500,"absorbants en feuilles,coussins,tapis,etc"):-
      oui(reseau,2),
      oui(quantite,1),
      oui(condition_climat2,C),
      membre(C,[2,3]).
recuperation(4000,80,5000,"absorbants en vrac"):-
      oui(reseau,1),
      oui(quantite,1),
      oui(temps_ecoule,1).
recuperation(4000,80,5000,"absorbants en vrac"):-
      oui(reseau,2),
      oui(quantite,1),
      oui(condition_climat2,1).
recuperation(12000,75,16000,"pompe anti-deflagrante et utilisation d'un explosimetre,ou un camion vacuum,ou un ecumoire"):-
      oui(reseau,2),
      oui(quantite,Q),
      membre(Q,[2,3]),
      oui(polluant,1).
recuperation(12000,75,16000,"pompe anti-deflagrante et utilisation d'un explosimetre,ou un camion vacuum"):-
      oui(reseau,R),
      membre(R,[1,3]),
      oui(quantite,Q),
      membre(Q,[2,3]),
      oui(polluant,1).
recuperation(6000,75,8000,"pompe et accessoires,ou un camion vacuum,ou un ecumoire"):-
      oui(reseau,2),
      oui(quantite,Q),
      membre(Q,[2,3]),
      oui(polluant,2).
recuperation(6000,75,8000,"pompe et accessoires,ou un camion vacuum"):-
      oui(reseau,R),
      membre(R,[1,3]),
      oui(quantite,Q),
```

```
        membre(0,[2,3]),
        oui(polluant,2).
recuperation(4000,80,5000,"absorbants en vrac"):-
        oui(reseau,3),
        oui(quantite,1).
```

Exemple de consultation

Est-ce que la tache:
    1-est sur le sol
    2-est sur l'eau
    3-est sur le plancher
      dans un batiment
1

Est-ce que la tache a ete repandue:
    1-dans une aire
      d'entreposage
      ou autres aires
    2-a l'exterieur,mais a
      l'interieur deslimites
      de l'installation d'un poste
    3-a l'exterieur des limites
      d'une installation
1

consulter le plan de drainage de l'installation,verifier la presence de contaminant dans le reseau

Est-ce que la tache atteint:
    1-un regard d'egout
    2-un puit d'acces aux cables et au reseau de drainage
    3-des conduites de cables
    4-des drains de surfaces
      ou souterrains
    5-une station de pompage
    6-aucunes de ces possibilites
1

blocage du regard d'egout

Est-ce que la tache est:
    1-circulaire
    2-allonge
    3-quelquonque
2

Est-ce que le risque de contamination de la nappe phreatique est:
    1-eleve
    2-faible
1

La condition au sol est:
    1-sol non-gele
    2-sol gele
    3-couvert de neige
    4-fonte de neige
1

Est-ce que le temps ecoule est:
     1-peu
     2-important
1

Est-ce que la quantite est:
     1-faible
     2-moyenne
     3-importante
3

Est-ce que le polluant est:
     1-de l'essence
     2-de l'huile ou autres produits
1

tranchee drainante,enlever le sol contamine
pompe anti-deflagrante et utilisation d'un explosimetre,ou un camion vacuum

	cout	% reussite	cout pondere
confinement:	12000	75	16000
recuperation:	12000	75	16000
			32000

fosse d'interception avec rigoles
pompe anti-deflagrante et utilisation d'un explosimetre,ou un camion vacuum

	cout	% reussite	cout pondere
confinement:	18000	60	30000
recuperation:	12000	75	16000
			46000

Est-ce que la tache atteint:
     1-un fosse sec
     2-un fosse humide
     3-un fosse humide etroit
     4-un pōnceau
     5-aucunes de ces possibilites
5
