

# Une méthodologie automatisée de la logique juridique

Alessandro Caldarone



Volume 31, numéro 1, 1990

URI : <https://id.erudit.org/iderudit/043006ar>  
DOI : <https://doi.org/10.7202/043006ar>

[Aller au sommaire du numéro](#)

Éditeur(s)

Faculté de droit de l'Université Laval

ISSN

0007-974X (imprimé)  
1918-8218 (numérique)

[Découvrir la revue](#)

Citer cet article

Caldarone, A. (1990). Une méthodologie automatisée de la logique juridique. *Les Cahiers de droit*, 31(1), 227–260. <https://doi.org/10.7202/043006ar>

Résumé de l'article

Le droit est une matière qui se prête très bien au traitement informatique. En effet nous pouvons dans cette discipline identifier deux aspects différents et complémentaires de la même réalité, à savoir l'aspect éthique et l'aspect logique.

Dans cette étude, l'auteur se penche sur l'aspect logique du droit. Pour ce faire, il utilise la logique formelle afin de réduire sous forme de règles la matière portant sur le remboursement des améliorations faites sur l'immeuble d'autrui. Le résultat final en est un d'intelligence artificielle avec la réalisation d'un système expert portant le nom de LEX-A. Celui-ci a été réalisé sur ordinateur Macintosh et utilise le langage de programmation PROLOG.

# Une méthodologie automatisée de la logique juridique

---

Alessandro CALDARONE \*

*Le droit est une matière qui se prête très bien au traitement informatique. En effet nous pouvons dans cette discipline identifier deux aspects différents et complémentaires de la même réalité, à savoir l'aspect éthique et l'aspect logique.*

*Dans cette étude, l'auteur se penche sur l'aspect logique du droit. Pour ce faire, il utilise la logique formelle afin de réduire sous forme de règles la matière portant sur le remboursement des améliorations faites sur l'immeuble d'autrui. Le résultat final en est un d'intelligence artificielle avec la réalisation d'un système expert portant le nom de LEX-A. Celui-ci a été réalisé sur ordinateur Macintosh et utilise le langage de programmation PROLOG.*

---

*Law is a subject matter that lends itself well to data processing. Indeed, it is a discipline in which we can identify the two distinctive and complementary aspects of one reality: the ethical and the logical aspects.*

*In this study, as the ethical question has been settled beforehand, the author delves into the logical aspect of law. To do so, he uses formal logic so as to reduce to logical rules the data related to the refunding of improvements made to a third party's building. The final result is one of artificial intelligence with the realization of an expert system bearing the name LEX-A. It was developed on a Macintosh computer and uses the PROLOG programming language.*

---

\* LL.M. La disquette LEX-A est disponible au bureau des *Cahiers de Droit*.

	<i>Pages</i>
<b>1. L'intelligence artificielle.....</b>	<b>230</b>
1.1. Historique et objet de l'intelligence artificielle .....	230
1.2. Les systèmes experts .....	231
1.3. Les juristes <i>versus</i> les systèmes experts .....	233
1.3.1. Le droit artificiel ou <i>simia juris</i> .....	235
1.3.2. Les systèmes experts réalisés en droit .....	237
<b>Conclusion .....</b>	<b>239</b>
<b>2. La représentation des connaissances .....</b>	<b>240</b>
2.1. Typologie des connaissances .....	240
2.2. Transfert des connaissances de l'expert à l'informaticien .....	240
2.3. La logique .....	241
2.3.1. Le calcul des propositions et les tables de vérité .....	241
2.3.2. Le calcul des prédicats .....	245
2.3.3. Le principe de résolution .....	245
2.3.4. Prolog .....	248
<b>Conclusion .....</b>	<b>250</b>
<b>3. Une expérience de système expert dans le droit des biens .....</b>	<b>251</b>
3.1. Les données de l'expérience .....	251
3.2. Représentation des connaissances sur le remboursement des améliorations sur l'immeuble d'autrui .....	252
3.3. Programmer en logique .....	258
3.4. La structure interne de LEX-A .....	258
<b>Conclusion .....</b>	<b>260</b>

Le discours sur l'informatique juridique est établi à partir des disciplines qui étudient les aspects théoriques et non pas substantifs du droit. La jurimétrie est en effet une méthodologie appliquée au droit positif. Or, les problèmes méthodologiques sont par leur nature des problèmes théoriques qui sont étudiés par des disciplines théoriques comme la théorie générale du droit.

La théorie générale du droit nous enseigne que le droit peut être étudié de différentes façons : le droit comme rapport juridique, le droit comme institution, le droit comme norme juridique, etc.

Pour nous, le droit est un ensemble de normes juridiques, vu dans le sens de la théorie normativiste de Hans Kelsen <sup>1</sup> (1881-1973). Malgré les critiques

1. KELSEN, *Teoria generale delle norme*, Turin, Einaudi, 1985.

dont elle a été l'objet et sur lesquelles il n'y a pas lieu de s'arrêter, cette théorie du droit a l'avantage de se baser sur un élément dit premier : le sème du droit, la norme juridique. Or, c'est de la combinaison structurée des normes juridiques que naît un système juridique. La théorie générale du droit peut conséquemment être considérée comme l'ensemble des études qui essaient de déterminer les caractéristiques structurelles des règles juridiques et leurs liens dans un système juridique donné.

Le rôle de la théorie générale du droit étant ainsi délimité, différentes méthodologies pourraient être utilisées pour son étude. Nous souhaitons porter notre attention sur une seule d'entre elles : la méthodologie basée sur la définition générale de l'intelligence artificielle<sup>2</sup>. L'application de cette méthodologie à la théorie générale du droit constitue une nouveauté, nouveauté tenant à ce que sur le plan pratique elle ouvre la porte à la méthodologie connexe des systèmes experts<sup>3</sup>.

Les grands thèmes dont nous traiterons correspondent aux grandes parties de l'article (I. L'intelligence artificielle ; II. La représentation des connaissances ; III. Une expérience de système expert dans le droit des biens) et se présentent dans une séquence conçue pour le lecteur ayant une formation juridique. Aucune connaissance spécifique de l'informatique, des mathématiques ou de la logique ne lui est demandée.

- 
2. L'intelligence artificielle est la « discipline visant à comprendre la nature de l'intelligence en construisant des programmes d'ordinateur imitant l'intelligence Humaine », A. BONNET, *L'intelligence artificielle*, Paris, InterÉditions, 1984, p. 17. Pour d'autres définitions de l'Intelligence artificielle voir aussi : R.C. SHANK, *The Cognitive Computer*, Reading, Mass., Addison-Wesley, 1984, p. 49 ; A. BARR et E.A. FEIGENBAUM, *The Handbook of Artificial Intelligence*, vol. 1, Stanford Cal., Heuristech Press, 1981, p. 3 ; M. BORILLO, « Limites de l'autonomie décisionnelle des machines », dans H.L. Dreyfus, *Intelligence artificielle : mythes et limites*, Paris, Flammarion, 1984, p. 423 à 455 ; L. RASTRIGUINE et P. GRAVÉ, *La cybernétique telle qu'elle est*, Moscou, Mir, 1979, p. 189 ; P.H. WINSTON, *Artificial Intelligence*, Reading, Mass., Addison-Wesley, 1977, p. 1 ; F. KLIX, *Human and Artificial Intelligence*, Amsterdam, 1979 ; P.H. WINSTON et R.H. BROWN, *Artificial Intelligence : An MIT Perspective*, sous la direction de Patrick Henry Winston et Richard Henry Brown, Cambridge, Massachussets, MIT Press, 1979.
  3. Un système expert gère l'information contenue dans une banque de connaissances avec des procédures d'inférences similaires à la logique humaine sous-jacente à toute recherche pour la résolution de problèmes complexes. « A computer program that has built into in the knowledge and capability that will allow it to operate at the expert's level » est la définition donnée par E.A. FEIGENBAUM et P. MCCORDUCK, *The Fifth Generation*, Reading, Mass., Addison-Wesley, 1983, p. 63 ; voir aussi : A. BARR et E.A. FEIGENBAUM, *The Handbook of Artificial Intelligence*, vol. 2, Stanford Cal., Heuristech Press, 1982, p. 79 ; R.H. MICHAELSON, D. MITCHIE et A. BOULANGER, *The Technology of Expert System*, (1985) 10 *Byte*, p. 303 ; H.C. MISHKOFF, *infra*, note 5, p. G-1.

Tout en essayant de rendre le texte accessible et clair, je dois cependant en signaler au moins une difficulté importante : le contraste entre la structure réticulaire du savoir informatique et la structure nécessairement linéaire de son exposition. En d'autres termes, une compréhension réticulaire doit s'ajouter à une lecture linéaire. La difficulté n'est pas telle cependant que la compréhension soit impossible.

Toujours dans le but de faciliter la lecture du texte, la plupart des exemples proposés dans la première et deuxième partie de cet écrit sont de nature générale et non pas juridique. Ce choix évitera au lecteur une gymnastique intellectuelle juridique inutile tout en mettant en relief l'aspect méthodologique et interdisciplinaire de cette étude. Par ailleurs, la troisième partie de cet écrit explique la démarche suivie pour la réalisation de LEX-A, notre système expert sur le remboursement des améliorations faites sur l'immeuble d'autrui.

## **1. L'intelligence artificielle**

### **1.1. Historique et objet de l'intelligence artificielle**

La voie la plus simple pour saisir les principaux points de contacts entre l'intelligence artificielle et le droit consiste à suivre la genèse même de l'intelligence artificielle.

Les recherches dans le domaine de l'intelligence artificielle ont pris naissance avec un écrit de A. Turing en 1947. Dans son livre, Turing affirme que l'on peut parler d'intelligence à propos de machine<sup>4</sup>. Au sens de Turing, le critère d'intelligence est déterminé de façon artificielle. Ce critère est basé sur la réussite d'un test (appelé depuis, test de Turing), qui consiste à avoir dans deux salles différentes un être humain et un ordinateur. Dans une troisième salle, se trouve un deuxième être humain, chargé de déterminer dans quelle salle se situe l'ordinateur. Le deuxième être humain interroge les deux salles au moyen de questions en langue naturelle. Plusieurs êtres humains se succèdent dans la troisième salle, et si le programme de dialogue de l'ordinateur réussit à tromper 50% ou plus de ses interrogateurs humains en leur faisant croire que le dialogue se fait avec un autre être humain, alors cette machine passe le test de Turing et est déclarée « intelligente ».

L'expression intelligence artificielle a vu officiellement le jour en 1956, à l'occasion de la Conférence de Dartmouth. Ce fut John McCarthy qui utilisa

---

4. A. TURING, « Computing Machinery and Intelligence » dans *Computers and Thought*, sous la direction de E. Feigenbaum et J. Feldman, New York, McGraw-Hill, 1963, p. 1-35.

ce terme pour la première fois<sup>5</sup>, dans le but de formaliser un cadre théorique de recherche, apte à résoudre divers problèmes, tel que celui de la traduction automatisée.

Les recherches en intelligence artificielle tendent à rapprocher (jusqu'à peut-être faire correspondre) les procédures de l'ordinateur à celles de l'intelligence humaine. Les systèmes informatiques, qui n'utilisent pas des techniques d'intelligence artificielle, suivent des schémas formels et rigides, des modèles algorithmiques<sup>6</sup>.

Toutefois, l'activité humaine intellectuelle suit des structures beaucoup moins rigoureuses. On parle ici de modèles heuristiques constitués par des conseils de bon sens aptes à résoudre un problème donné, mais de façon moins rigoureuse et certaine qu'un algorithme<sup>7</sup>. L'intelligence artificielle s'occupe de programmer les ordinateurs de façon à ce qu'ils paraissent suivre un modèle heuristique ou intuitif, et non pas un modèle rigide.

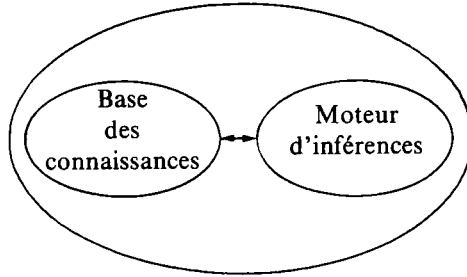
Parmi les divers champs d'application de l'intelligence artificielle, citons par exemple le traitement de la langue naturelle, l'apprentissage par machine, la vision par ordinateur, et bien entendu les systèmes experts, que nous allons maintenant développer.

## 1.2. Les systèmes experts

Comme nous l'avons déjà vu, les systèmes experts sont des programmes conçus de façon à imiter le comportement d'un expert humain. Un système expert contient une base de connaissances et un moteur d'inférences (voir la figure 1) : les connaissances représentent le savoir que l'homme a dans un certain domaine et l'ensemble des règles qui les définissent ; le moteur d'inférences est l'ensemble des procédures qui permettent l'utilisation optimale des connaissances. Il existe, faut-il le préciser, différents moteurs d'inférences

- 
5. E. CHARNIAK et D. McDERMOTT, *Introduction to Artificial Intelligence*, Reading, Mass., Addison-Wesley, 1985, p. 9-11 ; A. TURING, *On Computing Number*, cité par Henry C. MISHKOFF, *Understanding Artificial Intelligence*, Dallas, Tex., Texas Instruments, 1986, p. 2-4.
  6. « Description de la suite d'actions à entreprendre afin d'arriver à une solution » dans A. BONNET, *supra*, note 2, p. 20. Voir aussi, G. IFRAH, *Les chiffres*, Paris, Laffont, 1985, p. 284 ; H.C. MISHKOFF, *supra*, note 5, p. 1-11 et G-1 ; J. ROULIN, G. SCHEPENS et P. VAN ASTELAER, « Introduction aux concepts techniques et glossaire commenté de l'informatique » dans *Le droit des contrats informatiques*, Bruxelles, Larcier, 1983, p. 9-28 ; R.C. SCHANK, *supra*, note 2, p. 5 ; M. WILLIAMSON, *Artificial Intelligence for Micro-computers*, New York, Brady Communications Company, 1986.
  7. R.S. MICHALSKI, J.G. CARBONELL et T.M. MITCHELL, *Machine Learning. An Artificial Intelligence Approach*, vol. 2, Los Altos, Californie, Morgan Kaufmann Publishers Inc., 1986, p. 710.

en fonction de la discipline objet du système expert : parfois il faudra donner des indications sur la qualité de l'objet, parfois ce sont les règles qui prédominent<sup>8</sup>.



**FIGURE 1 : Organisation de base d'un système expert**

Un autre aspect fondamental des systèmes experts est celui de fournir des explications à son utilisateur, et ceci à deux niveaux : d'une part, il doit pouvoir justifier le « pourquoi de ses questions », ce qui lui permet de lui faire savoir ce qu'il est en train d'essayer de faire ; d'autre part, il doit pouvoir justifier comment il est parvenu au résultat final, ou, en termes informatiques, il doit pouvoir « reparcourir son raisonnement ». Ceci est capital pour gagner la confiance des utilisateurs. Ces derniers peuvent ainsi vérifier — et éventuellement critiquer — la bonne marche du programme.

Les utilisateurs d'un système expert peuvent être des novices qui ont besoin d'un conseil dans un domaine pointu. Un exemple peut être celui d'un notaire qui, confronté pour la première fois de sa pratique à un problème de droit international privé, doit déterminer au préalable s'il s'agit d'une question d'état et capacité plutôt que d'une question de régimes matrimoniaux afin de trouver la bonne solution. Les spécialistes aussi pourraient bénéficier d'un système expert. En effet, un système expert contient par définition l'expérience de l'expert ou des experts qui ont travaillé à sa mise au point, expérience qui n'est pas toujours rapportée dans la littérature classique du domaine.

Aujourd'hui les systèmes experts sont le centre d'intérêt tant des chercheurs que des industries. Dans les systèmes experts, l'étude de l'intelligence artificielle se concrétise en produits utilisables dans des activités pratiques. Au début, les programmes conçus pour l'étude de l'intelligence artificielle visaient à faire faire à l'ordinateur la démonstration de théorèmes ou à jouer aux échecs. Il s'agissait donc de jeux intellectuels sans importance pratique immédiate.

8. Marie-Odile CORDIER, « Les systèmes experts », dans *La recherche en intelligence artificielle*, Paris, Seuil, 1987, p. 285-312.

Aujourd'hui, les systèmes experts aident les médecins à établir des diagnostics complexes, effectuent des analyses chimiques (comme le système expert Dendral), des analyses géologiques (Prospector), trouvent la combinaison optimale des composants des ordinateurs VAX (XCON), etc. Pour accomplir ces tâches, les systèmes experts se servent d'ordinateurs traditionnels et de programmes qui œuvrent surtout sur les symboles de logique. Les langages traditionnels, tels que Basic, Pascal, etc., sont peu adaptés au traitement symbolique des données, et des langages expressément conçus pour l'intelligence artificielle sont en fait couramment utilisés en intelligence artificielle ; les plus importants étant Lisp<sup>9</sup> aux États-Unis et Prolog<sup>10</sup> en Europe et au Japon.

### 1.3. Les juristes *versus* les systèmes experts

Il y a déjà longtemps Nietzsche écrivait : « les méthodes, on ne le répétera jamais assez, sont essentielles, et aussi le plus difficile, ce qui se heurte le plus longtemps aux habitudes et à la paresse »<sup>11</sup>.

À titre d'illustration de ce que disait Nietzsche, examinons pour un instant la réaction négative des juristes confrontés au problème méthodologique faisant l'objet de ce travail. Ces réactions ont été recensées dans différents articles juridiques de langue anglaise, française, italienne et espagnole. Nous en ferons une synthèse. D'autres réactions proviennent d'expériences personnelles.

*Interprétation des lois.* En matière d'interprétation des lois, on admet que l'ordinateur peut être utilisé pour des procédés de raisonnement et de déduction dans les cas simples. On refuse cependant d'admettre que la machine puisse en faire autant dans les cas plus complexes. L'intelligence humaine serait la seule à pouvoir s'attaquer à ces cas complexes.

On établit également, en matière d'interprétation des lois, une relation entre le travail de la machine et le style exégétique utilisé au XIX<sup>e</sup> siècle. Comme cette méthode d'analyse des textes de lois a été abandonnée parce que considérée imparfaite et ne se souciait guère des réalités sociales ni des exigences de la justice ou de l'équité<sup>12</sup>, pourquoi le réintroduire par le biais des ordinateurs ?

*Principes généraux du droit.* Lorsque le juge ne trouve pas dans la loi, *lato sensu*, la solution de certains problèmes particuliers, il cherche, par la synthèse de plusieurs dispositions éparées, des règles générales, des principes

9. Pamela MCCORDUCK, *Machines who Think*, San Francisco, W.H. Freeman, 1979, p. 216.

10. Alain COLMERAUER et al., *PROLOG*, Paris, InterÉditions, 1985.

11. NIETZSCHE, *l'Antechrist*, Paris, Gallimard, 1978, par. 59.

12. H. DE PAGE, *Traité élémentaire de droit civil belge*, t. 1, Bruxelles, Établissements Émile Bruylant, 1952, n. 6.



généraux dont les textes constitueraient des applications particulières. La violation de ces principes généraux sera dès lors sanctionnée par le juge au même titre que la violation d'une loi<sup>13</sup>. Si le juge dégage des principes généraux de droit par des procédés logiques, ce n'est cependant pas par des procédés de la logique déductive, mais par des procédés d'analogie et d'induction. En effet, le juge crée une nouvelle règle de droit dont les textes sont censés être dérivés. La machine, qui est purement déductive par définition, ne pourrait élaborer par induction ou analogie une règle de droit qui n'aurait pas été insérée dans ses circuits.

*Les antinomies.* « L'antinomie est la contradiction réelle ou apparente entre deux lois, deux règles de droit »<sup>14</sup>. Un exemple d'antinomie dans le cadre d'un même système de droit peut être le suivant : un navire fait naufrage. À bord de ce navire se trouvent le propriétaire du bateau et son fils, déshérité par lui. Seul le fils se sauve tout en sauvant le bateau. Le fils déshérité par son père ne peut pas lui succéder, mais une autre forme juridique précise qu'au cas de naufrage le navire appartient au survivant l'ayant sauvé. En l'espèce, quelle loi s'appliquera ? Une méthode pour éliminer ce problème d'antinomies est de fonder sa propre solution sur des considérations morales ou philosophiques, extérieures à la loi proprement dite pour ainsi éliminer l'une des dispositions ou pour choisir entre un texte et une règle de droit non écrite<sup>15</sup>. La machine ne pouvant pas faire de considérations d'ordre morale ou philosophique, l'antinomie bloquerait nécessairement la machine et empêcherait son fonctionnement.

*Les lacunes du droit.* En cas de lacunes, le juge doit décider selon l'état du droit et est par conséquent obligé, en l'absence de texte directement applicable, soit de rejeter la demande, soit de rechercher un moyen de combler la lacune. Or, on ne peut concevoir une machine qui pourrait elle-même combler ces lacunes et par conséquent élaborer elle-même une règle de droit à partir de la législation en vigueur.

*Philosophie et morale.* Les notions d'ordre public et d'équité sont essentielles. Elles varient d'une manière peut-être insensible mais certaine. Pour éviter de rendre une décision choquante à laquelle il serait amené par les rigueurs de la logique déductive, le juge va remonter aux prémisses et

13. François GENY, *Méthode d'interprétation et sources en droit privé positif : et sa critique*, t. 1, Paris, Librairie Générale de droit et de jurisprudence, 2<sup>e</sup> éd., 1954, p. 305.

14. P. FORIERS, « Les antinomies en droit », communication faite au Centre national de recherches de logique, le 4 mars 1961, publiée dans Perelman, Ch. (éd.), *Les antinomies en droit*, Bruxelles, Bruylant, 1965, p. 20.

15. Luc SILANCE, « Exemples d'antinomies et Essai de classement », communication faite au Centre national de recherches de logique le 27 janvier 1962, publiée dans *Les antinomies et le droit*, *supra*, note 14, p. 63 s.

modifier les bases de son raisonnement en remplaçant les règles de droit normalement applicables par des principes différents, le plus souvent non écrits et puisés dans ses propres conceptions morales ou philosophiques<sup>16</sup>. La subtilité d'un tel raisonnement dépasse, et de loin, les capacités d'une machine. Une machine n'est pas capable de prendre en considération les divergences de vue existant entre les différents groupes qui la constituent et de modifier son interprétation en suivant l'évolution des mœurs et de la philosophie.

*Indifférence.* Souvent les juristes sont indifférents aux disciplines autres que le droit *strictu sensu* et préfèrent s'enfermer dans la tour d'ivoire de leurs solides constructions dogmatiques. Ceci n'empêchera pas certains juristes de prendre *a priori* une attitude négative envers l'automatisation et de voir le droit fait à l'aide de l'ordinateur comme un « droit artificiel »

### 1.3.1. Le droit artificiel ou *simia juris*

Toutes les réactions négatives de la part des juristes par rapport aux systèmes experts ne trouvent point de justifications sérieuses et cachent, derrière des argumentations philosophiques, beaucoup d'ignorance et d'indifférence par rapport aux nouvelles réalités de la société humaine.

Malheureusement il nous est impossible de traiter de façon complète cette problématique étant donné sa nature philosophique. Cependant, certains éclaircissements s'imposent à ce stade de la recherche.

Dans l'histoire de la pensée juridique, on peut remarquer une constante oscillation entre deux pôles d'intérêt. D'un côté, le principe de l'éthique, c'est-à-dire l'ensemble des valeurs de l'être humain, veut que le juriste adapte ses lois aux exigences de la société dont il fait partie. De l'autre côté, le juriste sent le besoin de sauvegarder la rationalité de son comportement juridique.

Le XVII<sup>e</sup> siècle va privilégier la prépondérance de l'application mécaniste du texte de loi par le juge, alors que le XVIII<sup>e</sup> siècle va plutôt consacrer l'approche par « l'esprit populaire » en réaction contre les excès du rationalisme juridique. Pendant les deux guerres mondiales, surtout en Europe, on assiste au retour d'une vision strictement rationnelle, positive et technique du droit. C'est la scission entre la logique et l'éthique. La deuxième période de notre siècle se caractérise, en partie, par un retour du droit naturel, c'est-à-dire le retour à l'éthique. Cependant, ces dernières années, on peut remarquer que la littérature juridique s'intéresse toujours plus aux aspects sémantiques, logiques et technologiques du droit<sup>17</sup>.

16. *Id.*, p. 136.

17. Voir pour tous, Jean GOULET, *La machine à faire le droit*, Québec, P.U.Q., 1987, p. 1 à 14.

Ceci dit, refuser l'aide de l'ordinateur et du droit artificiel est un contresens. En effet, le droit artificiel n'est autre chose qu'une métaphore qui sert à individualiser le moment logique-formel du phénomène droit à côté du moment éthique du droit. Ce sont seulement deux aspects de la même réalité. L'expression droit artificiel est donc arbitraire et son utilisation est possible seulement par opposition au côté éthique du droit.

Dans ce sens, les systèmes experts font du droit artificiel en aidant les juristes à trouver la solution d'un problème juridique donné et le point de passage du juriste aux systèmes experts nous est fourni par la logique<sup>18</sup>.

C'est effectivement grâce à la représentation logiquement impeccable de n'importe quelle problématique juridique que le juriste trouvera dans l'ordinateur un collaborateur très efficace. Le juriste devra entreprendre une œuvre de réduction de la problématique juridique à sa dimension logique afin de pouvoir la soumettre à un procédé de transformation rigoureusement technologique : le résultat sera celui d'un produit de droit artificiel.

C'est l'aspect logique du droit lui-même qui permet d'appliquer au droit une méthodologie qui pourrait rendre le droit sans lacunes, antinomies, etc. Dans la deuxième partie de ce travail, nous allons voir que la logique appliquée au droit ne laisse plus de place pour les antinomies. Celles-ci seront découvertes et éliminées.

Un système expert pourrait aussi combler les lacunes existantes dans notre droit dans la mesure où le droit positif est — ou est censé être — le miroir fidèle de la société où il opère. Les développements subséquents de la logique, qui ont abouti au principe de Résolution<sup>19</sup>, permettent en effet de créer par ordinateur du droit nouveau à partir du droit en vigueur se trouvant dans la mémoire de ces ordinateurs. Le droit positif, comme tous le savent, peut être considéré comme un ensemble de lois, règlements, usages, principes généraux, normes d'ordre morale, etc. Quand nous indiquons que le savoir juridique d'un expert humain est confié à un système expert, nous tenons nécessairement compte de toutes ces facettes du droit. Mais, dès qu'une de ces facettes change, le système expert est évidemment dépassé. Admettons, à titre d'exemple, que ce soit la morale d'une société donnée qui a changé, le comportement des individus qui composent cette société changera lui aussi, parfois de façon immédiate et radicale, parfois de façon insensible mais certaine.

Le juge, étant un humain lui aussi, tiendra compte des changements survenus en société et « pour éviter de prononcer une décision choquante à

---

18. Voir *infra*, section 2.3 s.

19. Voir *infra*, section 2.3.3 s.

laquelle il serait amené par la rigoureuse logique déductive, le juge va remonter aux prémisses et modifier les bases de son raisonnement en remplaçant les règles de droit normalement applicables par des principes différents, le plus souvent non écrits, qu'il puisse dans ses conceptions morales ou philosophiques »<sup>20</sup>. Ces changements de société n'auront aucun impact sur le système expert. Ceci n'exclut pas qu'on puisse parler de morale des systèmes experts, ce sera la morale de son créateur fixée à un moment donné d'une époque donnée.

Comme l'a dit A. Einstein, l'ordinateur ne peut pas avoir la conscience du résultat obtenu<sup>21</sup>, c'est-à-dire que l'ordinateur n'a pas « l'intention » dans le sens plein et humain du terme qui implique toujours une intention de résultat d'origine morale. Il en va de soi que le moment éthique du droit ne pourra pas être confié à un système expert. Même si les ordinateurs un jour arrivaient à faire tout ce que l'homme peut faire, ils ne pourraient qu'avoir la conscience « d'être machine », c'est-à-dire d'être une créature de l'homme, dont la construction, la façon d'œuvrer est faite de manière différente de l'homme. Ils n'auront certainement pas une conscience humaine. Un système expert ne reproduit donc pas la pensée juridique, il l'imité seulement et c'est pour cette raison qu'un expert en droit peut être défini de *simia iuris*.

### 1.3.2. Les systèmes experts réalisés en droit

Dans le domaine du droit, une expérience très significative de système expert est celle de projet TAXMAN (commencée en 1972, elle regroupe à présent TAXMAN I et TAXMAN II) de McCarty<sup>22</sup>. Ce système a une grande importance théorique parce qu'il permet de saisir les mécanismes mêmes de l'argumentation juridique, appliqués en l'espèce au droit fiscal. On

20. L. SILANCE, *supra*, note 15, p. 136.

21. Rapporté par D. THOMSON, « Can a Machine be Conscious », dans *The British Journal for the Philosophy of Science*, XVI, 1966, p. 33.

22. L.T. MCCARTY, « Reflections on TAXMAN: An Experimental in Artificial Intelligence and Legal Reasoning », (1977) 90 *Harvard Law Review* 837 ; « The TAXMAN Project: Towards a Cognitive Theory of Legal Argument », dans *Computer Science and Law: advanced workshop on computer science and law*, Swansea, 1979 ; C. CIAMPI, « Artificial Intelligence and Legal Information Systems », *International Conference on Logic, Informatic, Law*, vol. 1, Florence, Italie, avril 1981 ; A.A. MARTINO, « Deontic Logic, Computational Linguistics and Legal Information Systems », *International Conference on Logic, Informatic, Law*, vol. 2, Florence, Italie, avril 1981 ; voir aussi l'article de R.E. SUSSKIND, « Expert Systems in Law: a Juisprudential Approach to Artificial Intelligence and Legal Reasoning », (1986) 49 *The Modern Law Review*, p. 1, 169-194 et la bibliographie que cet auteur cite.

peut signaler à la même époque le projet JUDITH<sup>23</sup> et les expériences réalisées en Grande Bretagne par Stamper pour la construction du langage LEGOL/NORMA<sup>24</sup>, qui est un langage hautement formalisé grâce auquel il est possible de traduire les règles juridiques afin de vérifier leur cohérence logique.

Mentionnons également POLYTEXT/ARBIT<sup>25</sup>, réalisé sous les auspices de l'Institut Suédois de la Défense et de la Recherche Nationale ; le CCLIPS (*Civil Code Legal Information Processing System*)<sup>26</sup> de C.G. DeBessonnet qui a, entre autre, comme tâche de faire une codification scientifique d'une partie du Code civil de la Louisiane, SARA<sup>27</sup> développé au Norwegian Research Center for Computers and Law (NRCCCL) ; le programme TAXADVISOR<sup>28</sup> de Michaelsen réalisé à l'aide d'EMYCIN<sup>29</sup> donne des indications sur la planification en matière d'impôt. D'autres systèmes experts dans le domaine du droit sont AUDITOR, DSCAS, LDS, LEGAL ANALYSIS, LRS, SAL<sup>30</sup>. Le principe est toujours le même : l'illustration des mécanismes

- 
23. W. POPP et B. SCHLINK, « Artificial Intelligence in der Rechtsinformatik. Stationen einer Forschungsreise in Nordamerika », (1975) 4 *Datenverarbeitung im Recht*, mais voir surtout A. JUDITH, « Computer Program to Advise Lawyers in Reasoning a Case », (1975) 4 *Jurimetrics Journal*, p. 303-314.
  24. R.K. STAMPER, *et al.*, « The LEGOL Project since 1976 », *Computers and Law*, n. 23, février 1980, 10-13 ; pour NORMA voir du même auteur « A Non-Classical Logic for Law Based on the Structures of Behaviour », *International Conference on Logic, Informatic, Law*, Florence, Italie, avril 1981, voir aussi notre note 1.
  25. S. LOF, *The POLYTEXT/ARBIT Demonstration System*, Stockholm, FOA, Report C40121-M7, septembre 1980.
  26. C.G. DEBESSONNET, « A Proposal for Developing the Structural Science of Codification », (1980) 8 *Autgers Journal of Computers, Technology and Law*, p. 47-63 ; « An Automated Approach to Scientific Codification », (1982) 9 *Autgers Journal of Computers, Technology and Law*, p. 25-75 ; « An Automated Intelligent System Based on a Model of a Legal System », (1984) 10 *Autgers Journal of Computers, Technology and Law*, p. 31-58.
  27. M. BORCHGREVINK et J. HANSEN, « SARA : A System for the Analysis of Legal Decisions », dans Jon Bing et K.S. Selmer, *A Decade of Computers and Law* (1980) ; J. BING, « Deontic Systems, A Sketchy Introduction », dans Jon Bing et K.S. Selmer, *A Decade of Computers and Law* (1980).
  28. R.H. MICHAELSEN, « An Expert System for Federal Tax Planning », (1984) 1 *Expert Systems*, p. 149-167.
  29. EMYCIN est un outil pour la réalisation des systèmes expert. L'utilisateur n'a qu'à introduire ses connaissances tout en donnant des indications quant aux liens logiques de ses connaissances. Ces genres de programmes ont un prix très élevé, le KDS coûte 1 440 \$, le KES coûte 4 000 \$, le M. 1 coûte 10 000 \$. D'autres programmes existent sur le marché tels l'Expert-Ease, le Exsys, l'Insight dont nous ne connaissons pas le prix comme pour l'EMYCIN.
  30. Il n'y a pas de texte où le lecteur pourra trouver une étude où est exposé le fonctionnement de chacun des systèmes experts existants dans le domaine du droit. Par contre, le lecteur qui désire avoir un aperçu général sur le fonctionnement des systèmes experts existants dans le

mêmes de l'argumentation juridique, appliqués, suivant les cas, à différents secteurs du droit.

Il avait été question, à la première Conférence Internationale sur l'Intelligence Artificielle et le Droit<sup>31</sup>, d'autres systèmes experts réalisés ou en voie de réalisation en droit, tels : *The Datalex Project*, *The LEX Project*, *A PROLOG Model of the Income Tax Act of Canada*, *An Expert System for Screening Employee Pension Plans for the Internal Revenue Service*.

Nous sommes évidemment encore très loin d'avoir en droit un système expert qui regroupe toutes les connaissances du savoir juridique et peut-être n'atteindrons-nous jamais pareil résultat. Pour nous, la chose la plus importante est cependant de continuer dans le sillage des « grands » que nous venons de citer. En effet, un système expert est toute autre chose que la simple application de l'automatisation au droit. Il se présente plutôt comme une méthode de raisonnement et son application au droit n'apporterait rien de vraiment neuf à ne prendre en compte que son aspect d'automatisation. Le système expert est toutefois susceptible de renouveler, et c'est là une partie de son originalité, des conceptions aussi vieilles que le droit lui-même, si on suit son modèle d'organisation des recherches.

## Conclusion

La réputation de l'intelligence artificielle a longtemps souffert de déclarations grandiloquentes de chercheurs enthousiastes sur ce que les ordinateurs arriveraient à faire en peu de temps<sup>32</sup>. Il faut donc combattre le scepticisme des observateurs et des experts incrédules en présentant rapidement un prototype qui fonctionne, même si les résultats atteints sont moins ambitieux que ceux initialement visés.

L'expert sera encouragé à poursuivre s'il voit rapidement une partie de ses capacités de raisonnement représentée dans une machine. L'informaticien, pour sa part, ne doit en aucune manière critiquer le manque de logique de l'expert ; ce manque de logique est souvent inhérent à la nature du domaine ; d'autre part, ce manque de logique n'est souvent qu'apparent car il peut

---

domaine du droit, peut consulter : Elio FAMELI, « Intelligenza artificiale e sistemi esperti nel diritto. Note in tema di apprendimento a ragionamento per analogia », *ID*, X, 1984, p. 165 à 179 ; M.A. BRAMER, *Research and Development in Expert Systems*, Cambridge, Cambridge University Press, 1985.

31. *The First International Conference on Artificial Intelligence and Law*, May 27–29, 1987, Boston, Massachusetts, ACM Press.

32. C'est la déclaration de Simon, rapportée dans L. DREYFUS, *supra*, note 2, p. 21 ; l'auteur critique abondamment l'esprit et le contenu de la déclaration de Simon tout au long de son ouvrage.

masquer des pas de raisonnement implicites que l'expert n'a pas encore dévoilés.

Nous sommes maintenant confronté à l'épineux problèmes de la représentation des connaissances d'un système expert. Ceci sera le but de notre deuxième partie et pour ce faire nous ferons appel à quelques principes de logique.

## 2. La représentation des connaissances

### 2.1. Typologie des connaissances<sup>33</sup>

Les connaissances d'un être humain se présentent de différentes façons. En effet, il y a des connaissances de type heuristique telles « si A et B sont connus, alors C peut être une hypothèse plausible mais certainement pas D ». D'autres connaissances sont de type procédural telles que : « si A est connu, alors s'intéresser à B puis à C, sinon s'intéresser à C puis à B ». Il y a aussi des connaissances factuelles du type « A se compose de B, C et D ».

### 2.2. Transfert des connaissances de l'expert à l'informaticien

Un schéma classique du transfert de l'expertise au « cognitien » est représenté par la figure 2. Ce processus prévoit le transfert du domaine (ED) de l'expert humain au « cognitien », ou informaticien (C) suivi de transfert de cette donnée du « cognitien » vers le futur programme expert<sup>34</sup>.

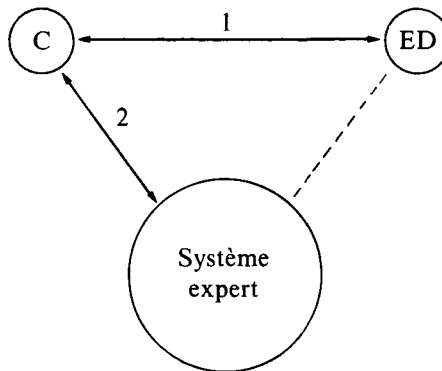


FIGURE 2 : Schéma classique du transfert d'expertise

33. Nous nous sommes inspirés du livre de F. HAYES-ROTH et D.A. WATERMAN, *Building Expert Systems*, Lenat D.B. éd., 1983.

34. A. BONNET, *supra*, note 2, p. 183, figure 5.

Une façon de permettre un examen facile des connaissances consiste à utiliser un formalisme clair. À titre d'exemple de ce que nous venons de dire, considérons comment la connaissance se représente sous forme de règles dans le système MYCIN<sup>35</sup> :

#### Règle 1

Si :           1) La morphologie de l'organisme est connue, et  
              2) la coloration de l'organisme est inconnue

Alors :           Il y a une possibilité (0.6) pour que la coloration de l'organisme soit Gram +.

Le nombre (0.6) est une mesure (entre -1 et 1) de l'opinion médicale experte témoignant de la force probante de la conclusion.

Ceci dit, il faut maintenant transférer les connaissances de l'informaticien au programme informatique qui prendra par la suite le nom de système expert.

### 2.3. La logique<sup>36</sup>

Les phrases utilisées dans le langage naturel, celui que nous utilisons tous les jours, ne sont pas assez rigoureuses pour construire un discours scientifique univoque. « On peut par exemple utiliser la logique pour représenter des propositions et des relations entre propositions, et pour en déduire d'autres propositions »<sup>37</sup>. Cette forme de logique, qui est en une extension du calcul des propositions<sup>38</sup>, est appelée calcul des prédicats.

#### 2.3.1. Le calcul des propositions et les tables de vérité

Le point de départ de cette logique est la proposition considérée comme un tout, d'où le nom de calcul des propositions. Les propositions du langage naturel sont étudiées de façon à vérifier d'une façon formelle si elles sont vraies ou fausses. Le calcul propositionnel s'occupe des propositions telles que : huit est plus petit que dix ; ce train s'arrête à Montréal ; les êtres humains ont deux jambes.

35. Mycin contient une connaissance experte concernant le diagnostic et le traitement des infections bactériennes du sang. La connaissance continue dans MYCIN est exprimée sous forme d'un ensemble de règles. Les quelque 500 règles de MYCIN ont pour forme « Si conditions alors actions ».

36. Voir pour tous, Mario LOSANO, *Informatica per le scienze sociali*, t. 1, Turin, Einaudi, 1985, p. 79 s. Pour nos exemples sur la logique nous nous sommes en partie inspirés du susdit livre.

37. W.F. CLOCKSin et C.S. MELLISH, *Programmer en Prolog*, Paris, Eyrolles, 1985. Par proposition on entend « a statement that has the property of being either true or false », T.H. EDWARDS et J.P. BARBER, « A Computer Method for Drafting Using Propositional Logic », (1975) 53 *Texas Law Review*, p. 965-1004.

38. A. BONNET, *supra*, note 2, 1984, p. 103.



Le calcul des propositions ne s'occupe pas des phrases qui ne peuvent être qualifiées de vraies ou de fausses, telles que : quelle heure est-il ? ; je me demande ce qu'ils font.

Ainsi peut-on délimiter le champ d'action du calcul des propositions. Mais demandons-nous maintenant quelles sont les activités à entreprendre sur les propositions individualisées.

Étant donné que l'on veut construire un langage rigoureux remplaçant le langage moins rigoureux, il faudra établir des règles semblables à celles qui régissent le langage naturel de façon à ce que le nouveau langage soit en mesure de reproduire la structure et non pas les imprécisions du langage naturel.

Le premier pas consiste à construire une grammaire ou une morphologie, en indiquant les composantes élémentaires du langage : l'alphabet pour le langage naturel et un ensemble de symboles pour le calcul des propositions. Comme il n'est pas vrai que toute combinaison des lettres de l'alphabet est un message pour le langage naturel, il n'est pas vrai non plus que toute combinaison de symboles est une proposition valide pour le calcul des propositions. Les propositions peuvent être reliées entre elles ; c'est la tâche de la syntaxe du langage naturel. Même dans le calcul des propositions, les propositions peuvent être reliées entre elles à l'aide des connectives logiques qui empêcheront les liens ambigus entre propositions.

Les connectives logiques sont :

$\neg$	non	(négation)
$\cap$	et	(conjonction)
$\cup$	ou	(disjonction)
$\rightarrow$	si... alors	(implication)
$\leftrightarrow$	si et seulement si	(équivalence)

Nous allons maintenant voir en quoi consiste le discours rigoureux introduit.

Le calcul des propositions nous indique les conditions qui doivent être satisfaites afin qu'une proposition donnée puisse être qualifiée de vraie. Mais il ne nous dit guère si la proposition est effectivement vraie : il nous dit qu'elle peut être vraie mais non pas qu'elle est vraie. Le calcul des propositions reste ainsi confiné aux deux pôles opposés du vrai et du faux. On ne discute pas cependant de la réalité de ce qui est vrai ou faux, puisqu'on ne se prononce pas sur la vérité ou la fausseté intrinsèque des propositions.

Le calcul des propositions étudie seulement les relations entre les symboles : un groupe de propositions sera valide non pas sur la base du contenu des propositions dont il est composé, mais sur la base du degré de vérité de chaque

proposition. Pour déterminer le degré de vérité de plusieurs propositions reliées par des connectives on a recours aux tables de vérité.

Essayons de mettre en pratique ce que nous venons de dire avec un exemple : « le ciel est serein et le soleil brille ». Nous représentons « le ciel est serein » par  $p$  et « le soleil brille » par  $q$ . La représentation en calcul des propositions de notre phrase «  $p$  et  $q$  » est «  $p \cap q$  ».

Se demander en quoi consiste le degré de vérité de cette phrase équivaut à se demander dans quel cas la conjonction de deux propositions est vraie. La réponse à la question se trouve dans la table de vérité de la conjonction :

$p$	$q$	$p \cap q$
V	V	V
V	F	F
F	V	F
F	F	F

Dans les deux premières colonnes, on indique les différentes combinaisons de la véracité (V) ou de la fausseté (F) des deux propositions<sup>39</sup>. Dans la troisième colonne, on indique le degré de vérité de la conjonction.

Il est facile de comprendre intuitivement le sens de cette table de vérité : la conjonction de deux propositions est vraie si les deux sont vraies.

La table de vérité ci-dessous définit les cinq connectives :

$p$	$q$	$p \cap q$	$p \cup q$	$p \rightarrow q$	$p \neg q$	$p \leftrightarrow q$
V	V	V	V	V	F	V
V	F	F	V	F	F	F
F	V	F	V	V	V	F
F	F	F	F	V	V	V

À partir de ces tables de vérité, il est possible de passer à l'analyse du degré de vérité des propositions plus complexes. Nous allons voir immédiatement un exemple : « si 6 divise 12 et si le fait que 6 divise 12 implique que 12 soit divisible par 3, alors 3 divise 12 ». En logique cette proposition peut s'écrire :

$$[6/12 \cap (6/12 \rightarrow 3/12)] \rightarrow 3/12$$

39. Dans l'ensemble, les combinaisons de la vérité ou de la fausseté sont au nombre de 4, puisque nous sommes en présence de  $p$  et  $q$  (variables). En effet, le nombre des combinaisons est donné par  $2n$ , où  $n$  indique le nombre des variables et 2 indique le binôme vrai-faux.

Maintenant nous représentons «6 divise 12» par  $p$  et «12 est divisible par 3» par  $q$ , nous pouvons écrire :

$$[p \cap (p \rightarrow q)] \rightarrow q$$

Pour vérifier si cela est vrai, il faut construire la table de vérité de cette proposition. Comme nous le savons, les deux premières colonnes de gauche donnent les valeurs possibles de  $p$  et de  $q$ . Comme en algèbre, nous devons résoudre en premier les parenthèses les plus intérieures. Dans notre cas, il s'agit de l'implication. Le premier pas dans la table sera :

$p$	$q$	$[p \cap (p \rightarrow q)] \rightarrow q$
V	V	V
V	F	F
F	V	V
F	F	V

Ce premier pas n'est que la simple reproduction de la table de vérité de l'implication. Il faut maintenant résoudre l'expression entre crochets. Le degré de vérité de  $p$  se retrouve dans la première colonne de gauche, tandis que le degré de vérité de l'implication ( $p \rightarrow q$ ) est le fruit de l'opération précédente. En tenant compte de la table de vérité de la conjonction, nous devons confronter le degré de vérité de  $p$  et ( $p \rightarrow q$ ). Voilà comment se présente maintenant la table de vérité :

$p$	$q$	$[p \cap (p \rightarrow q)] \rightarrow q$
V	V	V V
V	F	F F
F	V	F V
F	F	F V

La dernière colonne que nous avons ajoutée (la troisième de gauche) donne le degré de vérité de la conjonction de l'expression entre crochets. Il ne nous reste maintenant qu'à confronter la colonne que nous venons d'ajouter avec celle de  $q$ . Le résultat est représenté par la table de vérité suivante :

$p$	$q$	$[p \cap (p \rightarrow q)] \rightarrow q$
V	V	V V V
V	F	F F V
F	V	F V V
F	F	F V V

Le fait que toutes les valeurs de la dernière colonne soient des vérités signifie que le raisonnement logique exprimé par les symboles est vrai : nous venons de formuler une loi logique.

En résumé, nous pouvons vérifier le degré de vérité de toute formule. Il faut réduire la formule à ses composantes élémentaires en travaillant, suivant une logique donnée et conformément aux tables de vérité, sur chacune de ses composantes.

### 2.3.2. Le calcul des prédicats

L'intérêt principal du calcul des prédicats par rapport au calcul des propositions est l'introduction de la notion de variable<sup>40</sup>. La signification de variable ne peut être expliquée que par l'introduction des quantificateurs.

Il y a deux types de quantificateurs (x est une variable quelconque et P une proposition quelconque) :

- ( $\forall x$ ) P = quantificateur universel (P est vrai quel que soit x)
- ( $\exists x$ ) P = quantificateur existentiel (Il existe quelque chose, représenté par x, pour laquelle P est vrai).

Voici quelques exemples :

$$(\forall x) (\text{CHEVAL}(x) \rightarrow \text{ANIMAL}(x))$$

signifie que, si x est un cheval, alors x est un animal. En langage naturel la phrase correcte est « tout cheval est un animal »

Tandis que la proposition « tout juriste a un Code » se traduit par :

$$(\forall x) (\exists y) (\text{JURISTE}(x) \rightarrow \text{CODE}(y) \cap \text{POSSEDE}(x,y))$$

c'est-à-dire, quel que soit le juriste x, il possède un code « y » qui est possédé par x.

Le prédicat est le nom qu'on donne à une relation, si nous mettons ensemble le verbe et son objet. Ainsi dans l'exemple précédent, « possède un code » est un prédicat.

### 2.3.3. Le principe de Résolution

Durant les années 60, J. Alan Robinson<sup>41</sup> découvrit le principe de Résolution. La Résolution est une règle d'inférence. Elle permet de créer de nouvelles assertions à partir d'assertions (axiomes ou hypothèses) déjà existantes<sup>42</sup>. Cependant, pour utiliser cette règle d'inférence, les faits et les

40. A. BONNET, *supra*, notre 2, p. 103.

41. J.A. ROBINSON, « A Machine-Oriented Logic Based on the Resolution Principle », (1965) *Journal of the ACM*, p. 23-41 ; voir aussi N. NILSSON, *Principles of Artificial Intelligence*, Paolo Alto, California, Tioga Publishing Company, 1980, p. 145 s.

42. A. BONNET, *supra*, note 2, p. 104.

règles doivent être exprimés de façon logique. Il y a en effet deux opérations logiques à parcourir et définir :

1) Considérons l'expression suivante : « Si vous avez un Macintosh, alors vous avez un ordinateur ». Cette phrase ou proposition peut s'exprimer également de la façon suivante : « vous n'avez pas de Macintosh ou vous avez un ordinateur ». Même si les deux phrases semblent différentes, elles sont identiques d'un point de vue logique.

En logique, comme nous le savons déjà, la façon de prouver une loi logique se fait par le biais des tables de vérité. Si les deux propositions sont identiques, comme nous l'avons affirmé dans le texte, alors le patron de vrai (V) ou faux (F) sera le même dans lesdites tables de vérité.

Les deux énoncés dont on doit prouver l'identité sont :

- A)  $X \rightarrow Y$  (« si vous avez un Macintosh, alors vous avez un ordinateur »)  
 B)  $\neg X \cup Y$  (« vous n'avez pas de Macintosh ou vous avez un ordinateur »)

X	Y	NON (X)	Si X, alors Y	NON (X) ou Y
V	F	F	V	V
V	F	F	F	F
F	V	V	V	V
F	F	V	V	V

Tableau de vérité

2) La deuxième opération logique nécessaire pour appliquer le théorème de résolution comme méthode d'inférence s'énonce ainsi : si on a NON (X) ou Y et X ou Z, nous pouvons résoudre ces deux clauses en une seule : Y ou Z car X et NON (X) s'annulent mutuellement (ils donnent une clause vide) lorsque les deux clauses sont combinées.

En d'autres termes, le principe de base de cette stratégie de résolution consiste à nier la thèse à prouver (voir le point 1). Puis, si on réussit à engendrer la clause vide [], on a de fait généré une contradiction, ce qui indique que la thèse à prouver était vraie (voir le point 2).

Un exemple ne saurait que mieux clarifier ce que nous venons de dire.

Supposons une base des connaissances contenant les règles suivantes :

- 1-Si Macintosh est un ordinateur, il est du type APPLE
- 2-Si un ordinateur est de type APPLE, il est fait aux États-Unis
- 3-Si un ordinateur est de type APPLE, c'est un ordinateur puissant

et le fait suivant :

4-Macintosh est un ordinateur ;

Supposons encore que nous voulons savoir si les États-Unis fabriquent des ordinateurs puissants. Pour répondre à cette question notre première règle de logique est de réécrire nos règles sous forme clausale, ce qui nous donne :

1-[NON(Macintosh=ordinateur),(ordinateur=type APPLE)]

2-[NON(ordinateur=type APPLE),ordinateur fait aux ETAS-UNIS]

3-[NON(ordinateur=type APPLE),ordinateur puissant]

tandis que le fait donne lieu à la clause 4

4-[NON(Macintosh=ordinateur)]

Selon la procédure indiquée précédemment, on nie la conclusion que l'on veut obtenir de la façon suivante :

5-[NON(ordinateur fait aux ETAS-UNIS)]

6-[NON(ordinateur puissant)]

En résolvant les paires de clauses 1 et 2 on obtient :

7-[NON(Macintosh=ordinateur),ordinateur fait aux ETAS-UNIS]

et en résolvant les paires de clauses 4 et 7 on obtient :

8-[ordinateur fait aux ETAS-UNIS]

et en résolvant les paires de clauses 5 et 8 on obtient :

9-[ ] car [NON(ordinateur fait aux ETAS-UNIS)],ordinateur fait aux ETAS-UNIS] est une contradiction qui indique que les ETAS-UNIS fabriquent des ordinateurs.

En résolvant les paires de clauses 1 et 3 on obtient :

10-[NON(Macintosh=ordinateur),ordinateur puissant]

et en résolvant les paires de clauses 4 et 10 on obtient :

11-[ordinateur puissant]

et en résolvant les paires de clauses 6 et 11 on obtient :

12-[ ] car [NON(ordinateur puissant),ordinateur puissant] est une deuxième contradiction qui indique que le Macintosh est un ordinateur puissant et que par conséquent les ETAS-UNIS fabriquent des ordinateurs puissants.

La Résolution est une règle de déduction de nouvelles connaissances. Pour ce faire, on utilise la forme clausale. La forme clausale obéit à une syntaxe très rigoureuse de la logique des prédicats : la forme clausale requiert

toujours la même syntaxe pour exprimer la même chose<sup>43</sup>. Il existe une sorte de clause, dite de Horn, caractérisée par un ensemble fini<sup>44</sup> de littéraux<sup>45</sup> dont au plus un est positif et relié par la connective logique  $\cup$ .

Par exemple :  $p \cup q \cup \neg r \cup s$  est une clause, mais non pas de Horn parce qu'elle a plusieurs  $(p \ q \ s)$  littéraux positifs, alors que les clauses  $\neg p \cup q \cup \neg r \cup \neg s$  et  $\neg p \cup \neg q \cup \neg r \cup \neg s$  sont des clauses de Horn : la première a seulement un littéral positif ( $q$ ) tandis que la deuxième n'en a aucun.

On peut toujours transformer une expression logique en une clause de Horn avec un procédé logique, que nous ne jugeons pas nécessaire d'expliquer dans les détails. Ainsi  $\text{cheval}(x) \rightarrow \text{animal}(x)$  sera traduit en une clause de Horn en ces termes :  $\neg \text{cheval}(x) \cup \text{animal}(x)$ .

La stratégie de résolution peut être complètement automatisée, ce sera alors à l'ordinateur de faire ce que nous avons fait sur papier. Le langage de programmation PROLOG fonctionne en utilisant la stratégie de résolution sous la forme clausale des clauses de Horn, c'est pour cette raison qu'on peut dire que PROLOG a déjà son moteur d'inférences.

### 2.3.4. Prolog

PROLOG a été réalisé à l'Université de Marseille en 1972 par Alain Colmerauer et son groupe de recherche. L'appellation PROLOG vient de PROgrammation LOGique. En 1981, les japonais ont annoncé leur intention de choisir PROLOG comme langage pour la réalisation de leur projet de « Cinquième génération ».

La programmation en PROLOG consiste à décrire à PROLOG le problème que nous essayons de résoudre et PROLOG utilisera ses procédures pour résoudre le problème pour nous. Les procédures de PROLOG se basent sur les techniques des clauses de Horn. Par exemple si nous disons à PROLOG que

TOUT CHEVAL EST UN ANIMAL  
 ARCHIBALDE EST UN CHEVAL  
 PROLOG en déduira automatiquement que  
 ARCHIBALDE EST UN ANIMAL

43. En logique des prédicats  $p \rightarrow q$  peut s'exprimer par  $(\neg p) \cup q$ , comme d'ailleurs  $p \leftrightarrow q$  peut s'exprimer par  $(p \cap q) \cup (\neg p \cap \neg q)$  ou bien  $(p \rightarrow q) \cap (p \rightarrow q)$ .

44. Nous avons spécifié « fini » parce que dans les mathématiques existe aussi l'infini.

45. Pour l'explication du mot littéral, voir un peu plus loin du même paragraphe.

En effet, un programme écrit en PROLOG se compose de faits et de clauses. Les faits sont la relation logique entre les objets et leurs qualités, et les clauses sont la syntaxe propre à PROLOG.

L'exemple du cheval, que nous venons de citer est une phrase qui se compose d'un verbe (est), d'un sujet (cheval) et d'un objet (animal). Si nous mettons ensemble le verbe et son objet, la grammaire française nous enseigne que ceci prend le nom de prédicat. Ceci est vrai aussi en PROLOG qui utilise la logique des prédicats pour résoudre les problèmes qu'on lui présente.

Dans la syntaxe de PROLOG<sup>46</sup>, il faudra mettre le verbe d'une phrase en premier lieu, suivi d'une parenthèse ouverte, du sujet, d'une virgule, de l'objet, d'une parenthèse fermée et finalement d'un point virgule. Le symbole  $\rightarrow$  de PROLOG représente le « si » condition.

Dire que « tout cheval est un animal » ne signifie pas autre chose que « tout cheval est un animal *si* cet animal est un cheval ». Étant donné que nous ne connaissons pas tous les chevaux de ce monde, nous pouvons utiliser la notion de variable pour exprimer ce concept et la variable que nous avons choisi pour notre exemple est la lettre « x ».

Nous allons maintenant programmer ce que nous venons de dire à propos du cheval en une construction grammaticale compréhensible à PROLOG :

EST (x,animal)  $\rightarrow$  EST (x,cheval);  
EST (Archibalde,cheval);

Maintenant nous demandons à PROLOG la question : est-ce que Archibalde est un animal ? PROLOG, en utilisant ces procédures logiques, répondra « oui »<sup>47</sup> sans lui avoir dit que Archibalde est un animal. En effet, PROLOG ne fait que suivre les lois de logique dont nous avons traitées dans les pages précédentes.

Nous pouvons conclure ce chapitre sur PROLOG en disant quelques mots sur la différence qui existe entre PROLOG et les autres langages de programmation. Le premier langage de programmation fut le langage binaire de la machine : le code-machine. Vint ensuite l'assembleur, symbolisation du précédent. Apparurent enfin les langages dits de haut niveau comme FORTRAN, COBOL, BASIC, PASCAL. Tous ces langages de programmation

46. Tous les PROLOG fonctionnent de la même façon, mais les différentes versions de ce langage font que la syntaxe peut être différente d'un langage à l'autre. Dans nos exemples nous allons utiliser les règles de syntaxe de PROLOG II.

47. En PROLOG II la question s'écrira « EST (Archibalde,animal) » et la réponse « oui » est représentée par [] tandis que Micro-Prolog sur IBM nous aurait répondu *yes*.



ont un point commun : le programmeur doit y décrire avec précision **comment** le résultat doit être calculé plutôt que de décrire **ce qui** doit être calculé.

En BASIC, par exemple, un programme consiste en une suite d'instructions, chacune d'entre elles décrivant une action que devra accomplir l'ordinateur, et donc, si nous voulons faire un système expert en BASIC, nous devons construire notre moteur d'inférence<sup>48</sup> et connaître non seulement le fonctionnement théorique de la logique, mais aussi son utilisation concrète.

PROLOG est un langage de programmation logique qui contient *in se* son propre moteur d'inférences.

### Conclusion

W.G. Leibniz a démontré dans une analyse célèbre que tout type de message, quelle que soit sa complexité, pouvait toujours être ramené d'une quelconque façon à une série de questions par « oui » ou par « non » exprimables par une série de 0 à 1. C'est la numération binaire exprimée par l'expression un suffit à tirer l'univers du néant<sup>49</sup>.

Déterminer si les propositions sont vraies ou fausses et établir lequel des opérateurs logiques les caractérisent nous amène à découvrir le squelette logique du discours naturel.

Cette structure logique n'est pas tout le discours, comme le squelette n'est pas tout le corps. En effet, le langage de la logique des prédicats a été conçu pour mettre en relief les structures permanentes du raisonnement humain, non pas les aspects reliés aux sentiments. C'est un langage artificiel, scientifique, non pas un langage naturel littéraire. Il en va de soi que la diversité du langage naturel est rendue uniforme par l'utilisation des symboles. Que la chose soit condamnable ou non est une question de point de vue.

Cette cristallisation du langage naturel n'est pas un défaut, mais une caractéristique du langage logique : c'est justement pour cela qu'il a été conçu.

48. Voir *supra*, sections 1.2 et 2.3.4.

49. G.W. LEIBNIZ, « Nova methodus pro maximis et minimis, itemque tangetibus, quae nec irrationales quantitates moratur, et singulare pro illis calculi genus », dans *Acta Euroditorum*, oct. 1684, p. 467-473, réimprimé dans *G.W. Leibniz Mathematische schriften*, par C.I. Gerhardt, Hildesheim (DEU), Georg Olms Verlagsbuchhandlung, 1962, T.V., p. 220-226. Pour plus de détails voir Émile RAVIER, *Bibliographie des œuvres de Leibniz*, Hildesheim, Georg Olms Verlagsbuchhandlung, 1966, p. 50, § 90.

Un ouvrage récent précise la place de cette œuvre dans les *operae leibnitianae*, vide André ROBINET, *Architectonique disjointive automates systémiques et idéalité transcendentale dans l'œuvre de Leibniz*, Paris, Vrin, 1986, p. 299 s.

### **3. Une expérience de système expert dans le droit des biens**

#### **3.1. Les données de l'expérience**

Dans les deux premières parties, nous avons appris les notions fondamentales concernant les systèmes experts. Il s'agit maintenant d'expliquer notre expérience de juriste confrontée aux nouvelles méthodologies que la technologie moderne met à notre disposition.

Comme il eût été impossible pour une seule personne de traiter le droit dans sa totalité, nous avons délimité notre champ d'analyse à un secteur spécifique du droit, à savoir : le remboursement des améliorations sur l'immeuble d'autrui.

Le choix du remboursement des améliorations sur l'immeuble d'autrui s'explique sur la base de deux principes : le premier est que les dispositions sur le remboursement des améliorations sur l'immeuble d'autrui sont complexes (elles font appel à de nombreux concepts tels la bonne ou la mauvaise foi, l'état de nécessité ou la situation de simple utilité, la règle de l'accession, etc.) et dispersées dans le Code. Il s'agit donc d'une matière qui n'est pas taillée sur mesure pour les fins de l'écriture en logique formelle. Le second est que la difficulté de la matière n'est pas telle qu'elle puisse mettre en péril la réalisation du projet. Effectivement, dans le cas du remboursement des améliorations sur l'immeuble d'autrui, le champ d'analyse est suffisamment restreint pour permettre une application normale de système expert dans le cadre limité d'un mémoire de maîtrise.

Ceci dit, cette étude ne vise point une synthèse juridique sur le problème du remboursement des améliorations apportées sur l'immeuble d'autrui, mais plutôt la formalisation des connaissances acquises en un langage plus rigoureux que celui adopté dans nos textes de lois, et ce afin d'arriver à une nouvelle méthodologie automatisée de la logique juridique. Nous n'avons donc pas l'intention d'ouvrir un débat sur ce qu'est la bonne ou la mauvaise foi, sur ce qu'est le droit d'accession et le fait que l'accession joue de façon naturelle ou artificielle. Notre démarche consiste à utiliser les règles généralement admises par les tribunaux et la doctrine majoritaire afin de les appliquer à notre expérience de système expert. En effet, la revanche en droit portant sur le remboursement des améliorations sur l'immeuble d'autrui est incluse et traitée dans le système expert LEX-A.

Dans cette troisième et dernière partie, le droit concernant le remboursement des améliorations sur l'immeuble d'autrui sera analysé en fonction de sa possible automatisation et il y aura donc une écriture en des termes qui

rendront son automatisation possible. Cette procédure est illustrée par les règles de 1 à 31<sup>50</sup>.

Ces mêmes règles ont servi pour la programmation en PROLOG et à la réalisation de LEX-A.

Pour mieux expliquer le rapport entre LEX-A et son utilisateur nous allons décrire deux cas concrets de consultation de LEX-A.

### **3.2. Représentation des connaissances sur le remboursement des améliorations sur l'immeuble d'autrui**

Les problèmes posés par le remboursement des améliorations sur l'immeuble d'autrui sont de deux ordres : d'un côté, il faut donner une solution au conflit de propriété existant entre le maître de l'héritage et l'auteur des impenses. Le législateur pourvoit à la solution de ce problème par l'article 408 C.C.B.-C. qui codifie un principe découlant du droit romain connu sous le nom d'accession. Cet article accorde la propriété des choses qui viennent naturellement ou artificiellement se greffer d'une façon accessoire à un principal au propriétaire du principal. De l'autre côté, une fois l'accession ayant joué, il y a le problème du règlement probable des indemnités entre le maître de l'héritage et l'auteur des améliorations, et ce parce qu'il y a perte de propriété de l'accessoire au profit du principal. « Le règlement de ces indemnités entre les parties fait l'objet, au Code et ailleurs, d'une série de dispositions particulières dérivées de l'ébauche de système général prévu à l'art. 417 C.c. »<sup>51</sup>.

Le juriste qui s'engage dans la rédaction d'un travail portant sur le remboursement des améliorations sur l'immeuble d'autrui peut s'attendre à ce que son lecteur puisse suivre son raisonnement, étant donné que ce même lecteur est juriste ou tout au moins a des connaissances provenant de son expérience de vie et liées au fait même qu'il est humain. C'est dire que le lecteur a un sens commun. Rarement il sentira donc la nécessité d'aller au-delà du sujet traité ; le contraire pouvant même être considéré comme un défaut.

Dans le domaine du remboursement des améliorations sur l'immeuble d'autrui, beaucoup d'importance est accordée, pour les fins du calcul des indemnités, à l'existence ou à l'absence d'un lien de droit entre le maître de l'héritage et l'auteur des améliorations. Ce lien de droit peut être de nature

---

50. Voir *infra*, section 3.2 s.

51. F. FRENETTE, « Des améliorations à l'immeuble d'autrui », [1980] *C.P. du N.* 1, p. 16, deuxième colonne.

réelle ou personnelle et représenter une institution juridique telle l'emphytéose, l'usufruit, le louage, etc. Le juriste qui écrit un texte sur le remboursement des améliorations sur l'immeuble d'autrui n'a certainement pas besoin d'expliquer à son lecteur ce qu'est un usufruitier, un emphytéote ou un locataire, etc. Il peut se permettre de penser que son lecteur connaît leur définition et que, au surplus, il peut toujours trouver ailleurs l'information nécessaire.

Une toute autre réalité se présente au juriste qui communique son savoir à un ordinateur. Ce dernier ne sait absolument rien, il faut tout lui dire car il n'est sûrement pas en mesure d'aller se renseigner ailleurs. C'est un travail de patience où la logique fait fuir toute tentative poétique.

Nous allons maintenant présenter les règles que nous avons dégagées de notre étude en droit sur le remboursement des améliorations sur l'immeuble d'autrui. Étant donné que LEX-A est destiné à résoudre les problèmes découlant du remboursement des améliorations sur l'immeuble d'autrui, certaines distinctions juridiques, qui n'ont pas d'impact financier direct sur la réglementation du remboursement, ont été omises. À titre d'exemple, nous ne faisons pas, à la règle 12, la distinction entre la détention légale pour soi ou pour un autre, étant donné que cette distinction n'a aucun impact sur la réglementation du remboursement, le législateur ne se préoccupant point de la question du remboursement de la personne qui possède pour soi.

- Règle 1** (doctrine)  
 Si : 1) main d'œuvre, et/ou  
 2) addition de matériaux  
 Alors : amélioration
- Règle 2** (ex. art. 417 al. 2 C.C.B.-C.)  
 Si : 1) amélioration, et  
 2) éviter une perte, et/ou  
 3) une moins-value  
 Alors : amélioration nécessaire
- Règle 3** (doctrine)  
 Si : 1) grosse réparation  
 Alors : amélioration nécessaire
- Règle 4** (ex. art. 417 C.C.B.-C.)  
 Si : 1) amélioration, et  
 2) plus value  
 Alors : amélioration non nécessaire

- Règle 5** (ex. art. 443 C.C.B.-C.)  
 Si :  
 1) lien de droit, et  
 2) nature réelle, et  
 3) *usus*, et  
 4) *fructus*  
 Alors : usufruit
- Règle 6** (ex. art. 487 C.C.B.-C.)  
 Si :  
 1) lien de droit, et  
 2) nature réelle, et  
 3) *fructus* limité  
 4) *usus* illimité à l'usage  
 Alors : usage
- Règle 7** (ex. art. 443, 487, 496 C.C.B.-C.)  
 Si :  
 1) lien de droit, et  
 2) nature réelle, et  
 3) *usus* limité à l'habitation  
 Alors : habitation
- Règle 8** (ex. art. 567, 568, 578 C.C.B.-C.)  
 Si :  
 1) lien de droit, et  
 2) nature réelle, et  
 3) *usus*, et  
 4) *fructus*, et  
 5) *abusus* quasi complet  
 Alors : emphytéose
- Règle 9** (ex. art. 1600s. C.C.B.-C.)  
 Si :  
 1) lien de droit, et  
 2) de nature personnelle, et  
 3) paiement d'un loyer  
 Alors : locataire
- Règle 10** (ex. art. 2193 C.C.B.-C.)  
 Si :  
 1) aucun lien de droit  
 2) *corpus*, et  
 3) *animus*  
 Alors : possesseur
- Règle 11** (ex. art. 2193 C.C.B.-C.)  
 Si :  
 1) appréhension matérielle de la chose, et  
 2) continu, et  
 3) paisible, et

- 4) publique, et
- 5) non équivoque

Alors : *corpus*

**Règle 12** (ex. art. 2193 C.C.B.-C.)

- Si :
- 1) la personne croit être le propriétaire de la chose, ou
  - 2) la personne ne reconnaît le domaine supérieur d'aucune autre personne sur l'objet possédé.

Alors : *animus*

**Règle 13** (ex. art. 412 C.C.B.-C.)

- Si :
- 1) possesseur, et
  - 2) avec titre, et
  - 3) ignorance des vices du titre.

Alors : possesseur de bonne foi

**Règle 14** (ex. art. 412 C.C.B.-C.)

- Si :
- 1) possesseur, et
  - 2) absence de titre, ou
  - 3) connaissance des vices dont est entaché son titre

Alors : possesseur de mauvaise foi

**Règle 15** (ex. art. 492, 462 al. 2, 468 C.C.B.-C.)

- Si :
- 1) amélioration nécessaire, et
  - 2) usage

Alors : L'usager n'a pas le droit d'être remboursé des améliorations nécessaires réalisées sur l'immeubles d'autrui, étant donné que ces dernières ne sont pas du domaine de sa compétence.

**Règle 16** (ex. art. 492, 462 al. 2, 468 C.C.B.-C.)

- Si :
- 1) amélioration nécessaire, et
  - 2) habitation

Alors : Le titulaire d'un droit d'habitation n'a pas le droit d'être remboursé des améliorations nécessaires réalisées sur l'immeuble d'autrui, étant donné que ces dernières ne sont pas du domaine de sa compétence.

**Règle 17** (ex. art. 462 al. 2, 468 C.C.B.-C.)

- Si :
- 1) amélioration nécessaire, et
  - 2) usufruit

Alors : L'usufruitier n'a pas le droit d'être remboursé des améliorations nécessaires réalisées sur l'immeuble d'autrui, étant donné que ces dernières ne sont pas du domaine de sa compétence.

**Règle 18** (ex. art. 1604, 1612, 1653.4 C.C.B.-C.)

Si :  
 1) amélioration nécessaire, et  
 2) locataire

Alors : Le locataire peut être remboursé pour les grosses réparations qu'il a effectuées si le propriétaire ne lui procure pas une jouissance adéquate.

**Règle 19** (ex. art. 417 al. 2 C.C.B.-C.)

Si :  
 1) amélioration nécessaire, et  
 2) possesseur de bonne foi

Alors : Le possesseur de bonne foi qui a effectué de grosses réparations sur l'immeuble d'autrui a droit à un remboursement, le montant se fixe au coût réel à l'époque de la contribution en matériaux et/ou en main d'œuvre. Il a, par ailleurs, un droit de rétention pour assurer le remboursement qui lui est dû.

**Règle 20** (ex. art. 417 al. 2 C.C.B.-C.)

Si :  
 1) amélioration nécessaire, et  
 2) possesseur de mauvaise foi

Alors : Le possesseur de mauvaise foi qui a effectué de grosses réparations sur l'immeuble d'autrui a droit à un remboursement dont le montant est fixé au coût réel à l'époque de la contribution en matériaux et/ou en main d'œuvre. Il a, par ailleurs, un droit de rétention pour assurer le remboursement qui lui est dû.

**Règle 21** (ex. art. 577 C.C.B.-C.)

Si :  
 1) amélioration nécessaire, et  
 2) emphytéose

Alors : L'emphytéote qui a effectué de grosses réparations ne sera pas remboursable. En effet, il est de la nature de l'emphytéose que l'emphytéote effectue de grosses réparations.

**Règle 22** (ex. art. 492, 462, al. 2 C.C.B.-C.)

Si :  
 1) amélioration non nécessaire, et  
 2) usage

Alors : Le titulaire d'un droit d'usage qui a effectué des améliorations non nécessaires ne sera pas remboursable.

**Règle 23** (ex. art. 492, 462 al. 2 C.C.B.-C.)

Si :  
 1) amélioration non nécessaire, et  
 2) habitation

Alors : Le titulaire d'un droit d'habitation qui a effectué des améliorations non nécessaires ne sera pas remboursable.

**Règle 24** (ex. art. 462 al. 2 C.C.B.-C.)

Si : 1) amélioration non nécessaire, et  
2) usufruit

Alors : Le titulaire d'un droit d'usufruit qui a effectué des améliorations non nécessaires ne sera pas remboursable.

**Règle 25** (ex. art. 1624 C.C.B.-C.)

Si : 1) amélioration non nécessaire, et  
2) locataire

Alors : Le locataire qui a effectué des réparations locatives ne sera pas remboursable.

**Règle 26** (ex. art. 417 al. 4 C.C.B.-C.)

Si : 1) amélioration non nécessaire, et  
2) possesseur de mauvaise foi

Alors : Le possesseur de mauvaise foi ne sera remboursé des améliorations non nécessaires que si le propriétaire décide de les retenir.

**Règle 27** (ex. art. 577 C.C.B.-C.)

Si : 1) amélioration non nécessaire, et  
2) emphytéose

Alors : L'emphytéote qui a effectué des améliorations non nécessaires ne sera pas remboursable.

**Règle 28** (ex. art. 417 al. 3 C.C.B.-C.)

Si : 1) amélioration non nécessaire, et  
2) possesseur de bonne foi

Alors : Le possesseur de bonne foi qui a effectué des améliorations non nécessaires a droit à une indemnité.

**Règle 29**

Si : 1) non possesseur, et  
2) non détenteur en vertu d'un lien de droit

Alors : Occupant par tolérance

**Règle 30**

Si : 1) amélioration non nécessaire, et  
2) occupant par tolérance

Alors : Absence de remboursement.



**Règle 31**

Si :           1) amélioration nécessaire et;  
                2) occupant par tolérance  
 Alors :           Absence de remboursement.

**3.3. Programmer en logique**

La représentation de nos connaissances sous forme de règles permet de passer directement à la programmation en PROLOG<sup>52</sup>. En effet il est possible d'écrire notre programme sans avoir recours à d'autres artifices.

La règle numéro 1 de notre base de connaissance, comme nous le savons, nous fournit les éléments nécessaires de la possession, à savoir : aucun lien de droit, le *corpus* et l'*animus* et s'écrit en PROLOG de la façon suivante :

est-possesseur(aucun-lien-de-droit,corpus,animus)->;

Le lecteur se rappellera que le nom de la relation « est-possesseur » est appelé prédicat, tandis que les objets sur lesquels porte la relation sont les arguments.

Cette clause peut aussi s'exprimer en une forme qui ressemble un peu plus au langage naturel, à savoir :

règle 1 (aucun-lien-de-droit.corpus.animus.nil,possesseur)->;

Cette façon d'écrire les clauses de PROLOG est pratiquement la transposition à l'horizontale de ce que nous avons écrit à la verticale lors de la rédaction de la règle 1<sup>53</sup>.

Nous privilégions cette dernière écriture pour la conception et la réalisation de LEX-A.

**3.4. La structure interne de LEX-A**

Le système expert LEX-A se compose d'une base de connaissances de 30 règles dont chacune possède un ensemble de prémisses décrivant une certaine situation et une solution si toutes les prémisses sont satisfaites. Comme nous l'avons déjà vu, chaque règle prend la forme :

Si P<sub>1</sub> et P<sub>2</sub> et . . . . et S alors a

Les P<sub>n</sub> sont les prémisses, le « a » est la solution au problème.

52. PROLOG II est la version que nous avons choisie, elle fonctionne sur ordinateur Macintosh.

53. Voir *supra*, section 3.2 s.

L'ensemble de ces règles constitue la « base de connaissances » spécifiques au domaine du remboursement des améliorations sur l'immeuble d'autrui, tandis que la connaissance générale concernant la façon de résoudre un problème spécifique s'appelle le « moteur d'inférences ». PROLOG possède son propre moteur d'inférences étant donné que ce langage de programmation utilise les techniques des clauses de Horn<sup>54</sup>.

Avec ces deux éléments (la base de connaissances et le moteur d'inférences) il serait possible d'interroger la base de connaissances pour arriver à la solution du problème concret en posant nos questions en PROLOG. Il faut munir le système expert d'un programme de consultation qui permette à l'utilisateur d'avoir une certaine conversation avec l'ordinateur en langage naturel et non pas en PROLOG.

À ces éléments de base s'en ajoute un troisième, qui est essentiel lorsque l'on veut résoudre un problème particulier. En effet, il faut en plus une mémoire de travail où sont insérées les valeurs particulières reliées au problème spécifique que l'utilisateur veut faire résoudre. En d'autres mots, la base de connaissances constitue la partie statique des connaissances concernant un domaine d'expertise, tandis que les données particulières introduites par l'utilisateur avec ses réponses permettent d'apporter au système une connaissance dynamique par instantiation<sup>55</sup> des faits et règles. En d'autres termes, LEX-A prend des notes sur le problème à résoudre de la même façon qu'un juriste serait porté à le faire. LEX-A écrit ses notes sur la disquette à la fin du programme.

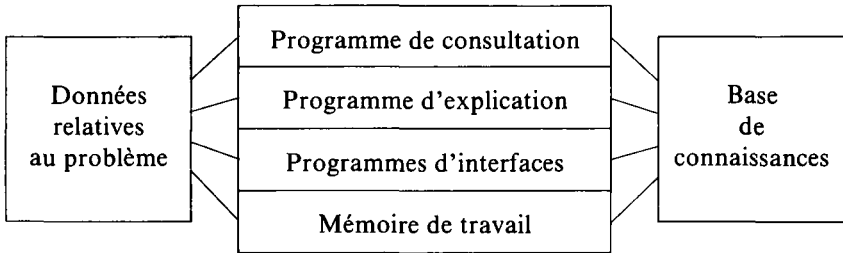
Le système LEX-A, à cause d'une certaine « métaconnaissance » ou connaissance de sa connaissance peut expliquer le raisonnement qu'il a suivi pour atteindre une conclusion et proposer une solution à un problème donné. Il possède une connaissance historique de la séquence des déductions faites lors de la consultation.

LEX-A est muni de trois programmes d'interface permettant une certaine conversation en langage naturel avec l'utilisateur dans la formulation des questions, dans l'explication ou justification de son raisonnement, et dans la représentation de ses connaissances relatives à la solution donnée au problème.

---

54. Voir *supra*, section 2.3.4.

55. L'instanciation est le processus consistant à remplacer une variable par une donnée. Voir pour tous le glossaire de Bonnet, *supra*, note 2, p. 267.



**Schéma interne de LEX-A**

### **Conclusion**

Nous sommes encore très loin d'appliquer d'une façon concrète et sur grande échelle les techniques de l'intelligence artificielle pour la construction de systèmes experts en droit. Tout dépendra probablement de leur rentabilité économique et de la réaction de la communauté juridique à l'intelligence artificielle.

Malgré cela, il faut néanmoins reconnaître l'indéniable intérêt théorique de ces recherches. En effet, l'intelligence juridique artificielle non seulement nous suggère des approches méthodologiques particulières au droit, mais nous aide aussi à comprendre les mécanismes, pour la plupart inconscients, qui sont à la base de toute interprétation juridique en proposant des modèles qui pourront par la suite être vérifiés de façon empirique.

Nous avons voulu ici nous situer dans le sillage de ce genre de recherche pour enfin aboutir à une méthodologie automatisée de la logique juridique. C'est l'aspect rationnel même du droit qui nous a permis de réaliser LEX-A. Apprendre à programmer en PROLOG a été une tâche assez difficile mais nécessaire afin de vérifier nos modèles théoriques ou règles.

Nous croyons pouvoir affirmer, en toute objectivité, que l'être humain et la machine ne sont pas en opposition l'un par rapport à l'autre. L'idée donc que la machine remplace l'homme, dans sa signification absolue, n'a pas de sens. La machine n'est que le prolongement de l'homme et c'est par elle qu'il élargit son emprise sur le cosmos.