

# *Acquérir une connaissance par Abstraction et Explication*

**Engelbert MEPHU NGUIFO**

L.I.R.M.M

UMR 9928 Université Montpellier II / CNRS

161 rue Ada - 34392 Montpellier Cedex 5 - France.

E.mail: [mephu@lirmm.fr](mailto:mephu@lirmm.fr) Fax: (33) 67 41 85 00 Tel: (33) 67 41 85 83 / 85

## **Résumé**

Le but de ce papier est de montrer l'importance de l'explication dans une stratégie de recherche d'abstraction, et de façon duale de l'abstraction pour la construction d'explications. La production d'abstraction s'appuie sur la structure de treillis de Galois et sur des mécanismes de raisonnement empirique et analogique. Nous nous basons sur la notion d'objection pour bâtir les explications compréhensibles par l'utilisateur et qui de plus vont lui permettre d'interagir avec le système de manière très précise pour éventuellement valider ou réfuter l'explication produite, et par conséquent l'abstraction construite.

## **Mots-clés**

Explication, Abstraction, Apprentissage automatique, Acquisition de connaissances, Coopération, Agents, Applications, Biologie Moléculaire.

## 1- Introduction

Les divers travaux sur l'explication [Paris, 1992], [Brézillon, 1992], [Balacheff, 1992] montrent qu'il est généralement reconnu que l'utilisateur d'un système accepte d'autant plus facilement les conclusions données par un système automatique que celui-ci est capable d'expliquer de manière satisfaisante son raisonnement, ses conclusions et ses conseils. De plus, le besoin pour le système de pouvoir expliquer son raisonnement de manière interactive, et dialoguer avec l'utilisateur pour résoudre un problème ou obtenir des informations supplémentaires devient de plus en plus évident.

Ce travail se situe à la confluence de l'acquisition de connaissances et l'apprentissage. En effet, "Peut-on croire, sans aucun doute, à une connaissance produite par une machine?". Nous essayons d'y répondre à travers ce papier. Nous cherchons à proposer un environnement coopératif de modélisation conceptuelle dans lequel deux agents : le système et l'utilisateur, interagissent. Le premier est capable d'extraire des connaissances à partir de données, par une méthode d'apprentissage numérique-symbolique, par exemple LEGAL [Liquière & Mephu, 1990], [Mephu, 1992] ou une de ses variantes LEGAL-E [Mephu, 1994a]. Il utilise ensuite ces connaissances pour reconnaître de nouvelles données et enfin explique ses décisions. Et l'utilisateur est capable de contrôler le processus d'apprentissage en acceptant les résultats et les explications du système, ou en les critiquant.

Notre démarche de modélisation conceptuelle s'organise autour de ces deux agents. L'amélioration de la modélisation dépend fortement de cette coopération entre les deux agents. Deux cas peuvent se produire : soit l'utilisateur est satisfait des résultats et des explications du système, soit il ne l'est pas. Dans le cas d'un désaccord, cela peut être dû à plusieurs facteurs dont les principaux sont : une connaissance apprise erronée qu'il conviendrait de faire évoluer, un mauvais paramétrage du système ou une mauvaise formulation du problème. Comme en pratique, il est difficile de faire évoluer la connaissance d'un système tout en maintenant sa validité et sa pertinence, nous faisons l'hypothèse qu'un problème bien formulé est facilement résoluble. Il est alors nécessaire pour le système de produire des explications plausibles et pertinentes pour aider l'utilisateur à bien reformuler le problème.

Nous présentons dans la première partie de cet article, notre méthode de modélisation conceptuelle qui s'appuie sur le système d'apprentissage LEGAL pour produire une abstraction, en explicitant sur le module d'explication qui est basée sur la notion d'objection. L'objection est une contrainte qui marque la différence entre des objets et des exemples d'un concept. Cette différence est due à des incorrections ou à des incomplétudes de l'expression d'un objet par rapport à l'expression d'un exemple. L'objection permet d'argumenter le raisonnement du système, et facilite la production d'explication. La seconde partie de ce papier est consacrée à la présentation de la méthodologie d'explication.

## 2- Modélisation conceptuelle

Nous posons que l'intelligence d'un agent humain ou machine est déterminée par la dynamique de son interaction avec le monde. Prenons

l'exemple d'un élève, celui-ci est dit intelligent lorsqu'à chaque question du maître, il a presque toujours une réponse juste et qu'en plus il est capable d'argumenter sa réponse, de tenir compte des critiques du maître pour améliorer ses argumentations. Aussi, l'interaction entre l'utilisateur et le système est élaborée comme un cycle au cours duquel chaque agent est capable d'interagir avec l'autre. Nous nous situons dans un cas où ni le système, ni l'utilisateur ne savent et ne dominent. Dans ce contexte d'incertitude, le système ou l'utilisateur dispose chacun de sa propre connaissance mais est susceptible de la faire évoluer. Le système produit des expressions susceptibles d'amener l'utilisateur à reformuler son problème. Nous ne pouvons pas assurer que la progression est effectivement une amélioration de la qualité des décisions, du fait des incertitudes des deux agents, mais tout au moins une amélioration de l'adéquation entre les résultats du système et les croyances de l'utilisateur. Si trop de contradictions venaient à se faire sentir de la part de l'utilisateur, on assisterait à une perte de pertinence ou de validité du système, il deviendrait de plus en plus silencieux ou ferait plus d'erreurs.

Notre objectif vise à concevoir une abstraction à partir d'exemples et d'une interaction avec l'utilisateur. L'abstraction se réfère à la réduction d'un problème au cours de laquelle les données pertinentes sont retenues, et les données non pertinentes sont oubliées. L'utilisation de l'abstraction dans le raisonnement est présente dans plusieurs sous-domaines de l'Intelligence Artificielle, et particulièrement en apprentissage [Knoblock & al., 1991] et en explication [Kassel & al., 1992]. Giunchiglia et Walsh [1992] montrent que la notion d'abstraction est un moyen efficace pour apprendre, pour fournir des explications de haut niveau, pour produire un raisonnement par analogie.

Dans notre démarche, le processus de conception d'une abstraction est progressif et nécessite plusieurs niveaux. Le tableau 1 montre une adaptation de la définition d'abstraction de Russell, sous forme d'un enchaînement à 9 niveaux depuis la détermination des exemples jusqu'à la production des preuves et de leur argumentation. A chacun des niveaux nous faisons correspondre un concept de notre modélisation.

| #   | Niveaux d'abstraction   | Concepts                            |
|-----|---|-------------------------------------|
| 1   | déterminer un ensemble d'exemples   | Exemples                            |
| 2   | décrire les exemples  | Fait                                |
| 3   | choisir une relation d'équivalence entre les exemples                             | Régularité                          |
| 4   | construire les classes d'équivalence d'exemples                                   |                                     |
| 5   | étiqueter chaque classe d'équivalence par un énoncé du langage                    | Hypothèse                           |
| 6   | déterminer une fonction d'appartenance pour chaque classe d'équivalence           |                                     |
| 7   | formuler le concept comme une relation qui lie entre-eux les étiquettes de classe | Formulat <sup>o</sup> de conjecture |
| 8.a | déterminer un protocole qui décide si un objet peut être ou non un exemple        | Preuve empirique                    |
| 8.b |   | Preuve analogique                   |
| 9.a | donner une argumentation, pour expliquer le succès ou l'échec de la décision      | Objections générales                |
| 9.b |   | Objections contextualisées          |

**Tableau 1 : Adaptation de la définition d'abstraction de Russell**

La colonne 3 indique les concepts correspondant aux différents niveaux. Au niveau 8, on a le choix entre une preuve empirique (8.a) ou une preuve analogique (8.b). Le niveau (9.a) est liée au niveau (8.a), tandis que (9.b) est liée à (8.b).

Le but de Russell était d'extraire le concept "nombre" à partir de la théorie des ensembles ZF. Pour cela, il a défini les 7 premiers niveaux. Ses

niveaux 8 et 9 étaient spécifiques à la théorie des nombres, et son approche visait à la définition d'un outil théorique qui résume la formation de concept de manière générale. Dans notre démarche, nos niveaux 8 et 9 sont dus à la définition de Lakatos [1984] du contrôle par preuves et réfutations. Nous nous appuyons ensuite sur le petit exemple "Être un bon Général" donné par Russell pour illustrer notre méthodologie et les concepts présents (tableau 2). Nous nous limitons ici à ces 2 tableaux pour illustrer notre propos.

| Phases                  | "être un bon général"  | Concepts                | Interact° Usager  |
|-------------------------|--|-------------------------|---|
| Exemples                | Napoleon, Wellington, ...  | Exemples                |   |
| Langage                 | petit, gros, intelligent, ..., blanc, age  | Fait                    |   |
| Abstraction             | (petit $\wedge$ gros $\wedge$ ...), (intelligent $\wedge$ ...),<br>(petit $\wedge$ blanc ..., (Napoléon, ...)) | Régularité<br>Hypothèse |   |
|                         | ?  | Formulat° conjectur     |   |
| Contrôle<br>et          | X est ~ car (petit, gros, ...)   | Preuve empirique        | ⇒ retour au niveau 6, ex.:<br>changer la fct d'appartenance     |
|                         | X est ~ comme Napoléon.  | Preuve analogique       | ⇒ retour au niveau 7, ex. :<br>modifier la relat° liant classes |
| Révision<br>coopératifs | (age $\geq$ 70), ou ( $\neg$ -blanc)... $\Rightarrow$ jamais ~<br>S. Keita n'est pas ~, car ( $\neg$ -blanc)   | Objecti° générales      | ⇒ retour au niveau 2<br>ex. : supprimer fait (blanc)            |
|                         | A l'opposé de Napoléon, X n'est pas<br>~, car non (petit et gros et ...)                                       | Objecti° contextuées    | ⇒ retour au niveau 1<br>ex.: modifier l'objet X                 |

**Tableau 2 : Coopération Homme-Machine**

Abbréviation : (i) ~ : un bon général (ii) ? : inconnu (iii) Ctual. : conceptual (iv) ° : ion

La coopération entre les 2 agents permet de retourner à un niveau inférieur si l'usager critique les résultats ( $\Rightarrow$ ). Par exemple, Soundiata Keita n'est pas un bon général parce qu'il est objecté par le fait (non blanc). L'usager critique cette objection parce qu'elle est insuffisante pour ne pas être un bon général. L'usager malgré sa critique va alors se rendre compte que tous les exemples initiaux sont blancs (connaissance acquise). Aussi il va modifier le langage de description de manière à ce que ce fait soit invalide ultérieurement. Mais l'usager peut conserver ce fait, en rajoutant dans l'ensemble suffisamment de bons généraux qui ne sont pas blancs. Le système va alors ensuite reformuler le concept (amélioration de la modélisation).

Notre méthodologie<sup>1</sup>, illustrée par le tableau 3, va consister :

- d'abord à construire un ensemble d'énoncés, qui sont des ressemblances entre les exemples; Cette construction est basée sur la structure de treillis de Galois [Wille, 1992], et a donné naissance au système d'apprentissage à partir d'exemples LEGAL [Liquière & Mephu, 1990; Mephu, 1992] dont plusieurs variantes sont présentées et comparées dans [Mephu, 1994a],
- ensuite à utiliser cet ensemble d'énoncés pour construire de nouveaux énoncés appelées objections qui s'opposent à la reconnaissance d'un nouvel exemple.

Pour justifier ou réfuter un objet, le système est capable de produire une preuve empirique à partir des ressemblances, ou une preuve analogique en recherchant des exemples qui lui sont proches [Mephu, 1992]. La proximité entre deux objets est fonction des ressemblances qu'ils vérifient. L'empirisme est utilisé comme un préalable à l'analogie qui va chercher à identifier les exemples proches à un objet donné.

<sup>1</sup> Cette démarche est semblable à celle utilisée par Haugeland [1989] pour concevoir ce qu'il appelle une expression bien formée. Il considère que le premier ensemble est formé d'énoncés simples, et le second d'énoncés complexes et acceptables. De plus, il part du fait qu'une combinaison d'énoncés simples n'est pas forcément un énoncé complexe et acceptable pour montrer qu'il faut des règles qui indiquent lesquels sont acceptables et lesquels ne le sont pas. A la différence de Haugeland, nous nous appuyons en plus sur une interaction avec l'usager pour contrôler et valider les énoncés produits par le système.

Dans les systèmes d'apprentissage à partir d'exemples, se posent les problèmes liés au choix du langage de description, et à la représentativité des exemples initiaux [Michalski, 1983; Kodratoff, 1993]. Nous posons comme hypothèse que la validité globale des connaissances apprises peut être testée de manière satisfaisante par des méthodes déductives ou statistiques connues, mais que la validité d'une décision prise et la pertinence des connaissances ne peuvent être jugées que par un usager [Mephu & Sallantin, 1993a]. D'où la nécessité de contrôler le raisonnement du système à travers une interaction avec l'utilisateur. Cette interaction sera d'autant plus efficace que sera le dialogue entre les deux agents, le système donnant des explications pertinentes, et l'utilisateur reformulant son problème à travers ces explications.

| Concepts               | Algorithme   |
|------------------------|--|
| Exemples               | $o_1, o_2, o_3, \dots, o_m$  |
| Fait                   | attributs binaires décrivant les exemples  |
| Régularité             | $\bigwedge_j \text{fait}_j, 1 \leq j \leq n$<br>Attribut pertinent: attribut qui apparaît au moins une fois dans une régularité (régularité, ensemble d'exemples)  |
| Hypothèse              | Correspondance de Galois: une hypothèse (O,A) est caractérisée comme suit :<br>(i) A est les sous-ensemble de tous les attributs que vérifient tous les objets de O,<br>(ii) et O est le sous-ensemble de tous les objets qui vérifient tous les attributs de A.<br>Critère de sélection :<br>• Une régularité est valide si elle est vérifiée par "suffisamment" d'exemples<br>• Elle est quasi-cohérente si elle est vérifiée par "peu" de contre-exemples   |
| Formulation de concept | Sup demi-treillis d'hypothèses<br>Les hypothèses sont ordonnées par la relation de sous-hypothèse-surhypothèse $\leq$ ,<br>( $O_1, A_1$ ) est une sous-hypothèse de ( $O_2, A_2$ ) ssi $O_1 \subseteq O_2$ (ou $A_1 \supseteq A_2$ )<br>% (régularité)   |
| Preuve empirique       | Principe de vote majoritaire sur l'ensemble des régularités pertinentes.<br>• Un objet $o_i$ est un exemple s'il vérifie beaucoup de régularités. $o_i$ est dit justifié.<br>• $o_i$ n'est pas un exemple s'il vérifie "peu" de régularités. $o_i$ est dit refuté.<br>• sinon, $o_i$ est ambigu.   |
| Preuve analogique      | L'ensemble des régularités vérifiées par X & un exemple Y, sont presque identiques.<br>Principe de confirmation de décision :<br>• L'objet $o_x$ est un exemple si :<br>- il vérifie "suffisamment" de régularités et est similaire à "beaucoup" d'exemples;<br>- ou il vérifie "peu" de régularités et est similaire à un exemple particulier *.<br>• il n'est pas s'il vérifie "peu" de régularités et n'est pas simil. à un exple particulier;<br>• sinon, cet objet est ambigu.<br>* Un exple particulier est un exple initial non similaire aux autres objets initiaux. |
| Objections Générales   | { $\neg(\forall \text{fait}_k)$ } tel que $\neg(\forall \text{fait}_k) \Rightarrow$ n'est pas un exemple.<br>Une objet <sup>o</sup> gle est la négat <sup>o</sup> d'une disjonct <sup>o</sup> de qq attr. pertinents / si elle est vérifiée par un objet alors celui-ci ne peut que vérifier "peu" de régularités, et est empiriquement refuté.<br>Un objet est objecté s'il vérifie une objection.  |
| Objections Contextuées | $\neg(\bigwedge \text{fait}_k)$ vérifié par X $\Rightarrow$ non similaire à l'exemple Y<br>Soit un ex. initial Y, une objection contextuée sur un objet X est une négat <sup>o</sup> de conjonct <sup>o</sup> de qq attributs pertinents / si elle est vérifiée par X, alors X ne peut pas être similaire à Y.   |

Tableau 3 : Formalisme d'abstraction des données avec LEGAL.

| Concepts  | Un bon site de jonction donneur  |
|---|--|
| 767 Exemples de 60 nucleotides dont 384 pour apprentissage & 383 pour test  | CCAGCTGCATCACAGGAGGCCAGCGAGCAGGCTCTGTTCACAGGGCCCTCGAGCCAGCTCTG<br>AGACCCGCGGGAGGGGGAGGACCTGCAGGGTGGAGCCCAACCGCCCTCCGTGCCCGCGC<br>.....<br>AAGCTGCATGTGGATCTCTGAGAACTTCAGGGTAAGTCTAGGAGATGTTCTCTTTGTCTCT  |
| 90 Contre-Exemples où 45 pour apprentissage & 45 pour test  | GGCCCCACCTGGTGAAGCCCTCTACCTGGGTGCGGGGAGCGAGGTTTCTCTACGCA<br>GTTCTAATCATTTCCACCATTTTTGTATTCTGTTTTAAACATCTATCTGGAGGCAGGACA<br>.....<br>TCTCGGGGCGGGCGGGCGGGCGGGGAGCGGTCGCCGGCCGGCCCGGACGCTGTGTGTC  |
| Fait  | A, G, C, ou T ou combinaison de nucleotides à la position i. $1 \leq i \leq 60$  |
| Régularité<br>Critères :<br>Validité = 200<br>Quasi-cohérence = 3   | ..... AA<br>GG GT C G .....<br>.....<br>C AAC<br>G GT GCG .....<br>.....   |
| Hypothèse<br>Seuils :<br>Justification = 15%<br>Refutation = 15%<br>Similarité = 37   | (..... AA<br>GG GT C G ..... {o <sub>i</sub> })<br>.....<br>C AAC<br>G GT GCG ..... {o <sub>j</sub> })   |
| Formulation   | Demi-trellis de 2282 régularités où 458 sont pertinentes   |
| Objets apprentissage<br>prédits comme bons<br>sites:  | Empirique Analogique<br>exemples 96.87 % 97.92 %<br>contre-exemples 04.44 % 03.11 %  |
| Preuve empirique  | ≥ 15% [régularités] ⇒ exemple et < 15% [régularités] ⇒ non exemple   |
| Bons sites en test :<br>exemples 95.04%<br>contre-ex 04.44%   | $\alpha_X = "AGCCAGGGCACTCACCAGGCTGCAAGAACAGTGTCTGGGGTAAGAGGGGAGCGGGGATCC"$<br>n'est pas un exemple car il vérifie 4% de régularités. L'usager peut critiquer la décision, et diminuer le seuil de justification à moins de 5%. Il peut poursuivre le processus.<br>Les régularités vérifiées par X et un exemple Y, sont presque identiques.  |
| Preuve analogique   | $\alpha_X$ n'est pas similaire à un exple particulier. Il est similaire au contre-exemple initial :<br>"ACTCTGTATTTTGGCCCTGAAACCCATAGTGGTCCGTCATGGATATGAAGCAGTGAAGGAA"<br>Son plus proche et bon exemple est $\alpha_Y$ , qui vérifie 16% de régularités<br>$\alpha_Y = "GTGGCAAGTGCCAAGCGCAGGCATCAAGGTAAGTCTGGCCCTCCCACTCCCTCCATCT"$  |
| Objections générales<br>$\alpha_X$ objecté par la dernière<br>object°. Si l'usager le<br>refute alors il modifie le<br>langage de description<br>en retournant à 2. | -(..... G ..... ) ⇒ non expl<br>-(..... C ..... ) ⇒ non ex.<br>.....<br>C C<br>G G ..... ) ⇒ non ex.   |
| Objections contextualisées  | $\alpha_X$ n'est pas similaire à $\alpha_Y$ à cause de<br>-(..... A<br>G ..... )<br>En remplaçant 'CA' par 'AG' dans la description de $\alpha_X$ , on obtient deux objets $\alpha_a$ et $\alpha_b$ :<br>$\alpha_a = "AGCCAGGGCACTCACCAGGCTGCAAGAAAGGCTCTGGGGTAAGAGGGGAGCGGGGATCC"$<br>qui vérifie 15% de régularités et est similaire à $\alpha_Y$ à cause des nucleotides soulignés. |

**Tableau 4 : Abstraction de données sur les sites de jonction donneur.**  
Les données ici sont celles de Noorderwiew & al. [1991]. Les résultats empiriques de LEGAL sont meilleures que d'autres méthodes symboliques, et aussi bons que certaines méthodes neuronales [Mephu & Sallantin, 1993b]

Notre méthodologie a été implémentée au travers du système LEGAL et testé sur plusieurs problèmes relatifs à la Biologie Moléculaire [Mephu, 1994b], et à l'archéologie [Lagrange & al., 1993]. Le formalisme d'abstraction de données est illustré par le tableau 3, et un exemple d'application est donné dans le tableau 4. Une présentation détaillée de LEGAL et de quelques unes de ses variantes est faite dans [Mephu, 1992, 1994a, 1994b]. Nous nous attacherons ici à montrer simplement son applicabilité sur le problème décrit dans le tableau 4. L'usager fournit au

système, un ensemble d'exemples (bons sites de jonction donneur d'épissage<sup>2</sup>) et éventuellement un ensemble de contre-exemples (segments de séquences qui ne sont pas de bons sites de jonction d'épissage, mais qui peuvent présenter des similitudes avec les séquences de bons sites). Une séquence d'ADN est une suite de caractères dans un alphabet à 4 lettres ou bases: A, G, C, T. Les exemples et contre-exemples de longueur 60 bases, sont extraits de séquences d'ADN, et choisis de manière à ce que tous aient le consensus  $\mathbf{GT}$  (en gras dans tableau 4). L'utilisateur fournit en outre un moyen de décrire ces séquences, par exemple nous avons considéré les paires de bases apparaissant position après position [Mephu, 1994b]. Ainsi, les 458 régularités pertinentes engendrées par LEGAL sont celles qui sont souvent vérifiées par les exemples (au moins 200 sur 384: tableau 4) et simultanément apparaissent sur peu de contre-exemples (au plus 3 sur 45: tableau 4). Les 2282 régularités sont ordonnées sous la forme d'un sup demi-trellis de Galois [Mephu, 1992]. Un objet  $oj$  vérifie la régularité

".....<sup>AA</sup><sub>GG</sub>GT<sup>A</sup><sub>C</sub>G....." s'il a (A ou G) aux positions 6 et 7, G et T aux positions respectives 9 et 10, (A ou C) à la position 12, et G à la position 13. LEGAL utilise ces régularités pertinentes pour bâtir des preuves empiriques dans la phase de reconnaissance de nouveaux objets. Pour cela, il détermine les seuils de justification et de refutation. Ainsi un objet est reconnu comme un exemple si le pourcentage de régularités qu'il vérifie est supérieur au seuil de justification. Il n'en est pas un si ce pourcentage est inférieur au seuil de refutation. Dans les autres cas, le système reste silencieux. La preuve analogique dans LEGAL revient à considérer un nouvel objet comme exemple s'il vérifie presque les mêmes régularités pertinentes qu'un exemple reconnu comme tel par l'utilisateur (en général, on utilise pour cela les exemples qui ont servi à l'apprentissage). Les résultats obtenus sur cette application (tableau 4) ont été comparés à ceux obtenus avec d'autres méthodes [Mephu & Sallantin, 1993b; Mephu, 1994b], et ont ainsi montré la robustesse des raisonnements empirique et analogique. Les preuves empirique ou analogique peuvent être réfutées par l'utilisateur avec retour arrière sur les étapes antérieures de production de connaissances ou avec recherche d'explication afin d'obtenir les raisons qui favorisent la production d'une telle preuve. Nous allons examiner comment les explications sont produites dans notre système et utilisées pour acquérir de nouvelles connaissances.

### 3- Explication

Une explication tente de rendre intelligible, à un interlocuteur, la réponse et la plausibilité de la réponse. Une explication présente un agencement d'éléments qui n'est pas obligatoirement une chaîne déductive. La qualité d'une explication destinée à un usager dépend à la fois d'une part des modes de raisonnement du système - lesquels déterminent la nature de la trace de

<sup>2</sup>La frontière entre un exon et un intron est un site de jonction d'épissage *donneur* appelé *site de jonction 5'*. Un exon est un fragment de l'ADN qui est conservé et traduit en protéine, alors qu'un intron est un fragment de séquences intermédiaires qui est éliminé lors de la maturation de l'ADN [Felsenfeld, 1985].

résolution de problèmes - et d'autre part du raisonnement explicatif réalisé sur cette trace, lequel inclut la modélisation de l'interlocuteur [KBS, 1992]. Le but de l'explication dans notre système est de faire ressortir les raisons de sa décision, tel que l'utilisateur trouvera les réponses plausibles aux questions suivantes :

- i) Pourquoi l'objet  $o_x$  est justifié comme exemple ?
- ii) Pourquoi  $o_x$  est réfuté ou est ambigu ?
- iii) Pourquoi  $o_x$  est similaire ou non à l'objet  $o_y$  ?

Pour intéresser un usager, ces explications devront permettre à celui-ci d'évaluer les connaissances apprises, de mettre en évidence les erreurs et les carences, la validité et la pertinence de celles-ci. L'explication apparaît dans notre système au travers du contrôle du raisonnement. La première preuve du système, appelée trace du raisonnement, constitue un début d'explication pour l'utilisateur. Cette preuve empirique ou analogique est de nouveau utilisée pour construire une explication plus explicite. Nous nous appuyons sur les objections générales dans le cas de la preuve empirique, et sur les objections contextuées pour l'analogie. L'utilisation de la notion d'objections comme principe de contrôle est due à Lakatos [1984], et une axiomatique en logique des propositions a été proposée par Barboux & Sallantin [1990]. Nous allons dans les paragraphes qui suivent décrire la méthode de production des objections et de leur utilisation dans la recherche d'explications.

### 3.1- Objections générales

Une *objection générale* est une expression de ce qu'il est suffisant de reconnaître sur un objet, pour qu'il ne soit pas considéré comme exemple du concept appris (conjecture). Le principe de génération consiste à rechercher des énoncés qui, lorsqu'ils ne sont pas vérifiés par un objet, empêchent celui-ci d'être *prouvé empiriquement* comme étant un exemple. La preuve empirique est faite au travers d'un énoncé de la forme "p% ensemble d'énoncés", où p% représente le pourcentage de justification, et l'ensemble d'énoncés n'est rien d'autre que l'ensemble des régularités pertinentes. Par conséquent, ces paramètres sont donc les données d'entrée de la procédure de construction des objections générales. Il faut générer des énoncés qui s'ils ne sont pas vus sur un objet, empêchent celui-ci de vérifier plus de p% de régularités pertinentes. Ces énoncés sont des ensembles de faits qui s'ils font simultanément défaut sur l'objet étudié, empêchent ce dernier de vérifier plus de p% de régularités pertinentes. Une objection est la négation d'un tel énoncé qui dans notre cas est une disjonction de faits. Une objection est donc de la forme: "non a et non b et .." équivalent à "non (a ou b ou ..)" où a, b et c sont des faits.  $\neg a \wedge \neg b \wedge \neg c \Leftrightarrow \neg(a \vee b \vee c)$

Dans le cas de l'application sur la prédiction des sites de jonction d'épissage (tableau 4), le fait pour un segment nucléotidique de ne pas avoir aux positions n et n+1 les consensus "G" ou "T" pour les jonctions donneur, l'empêche d'être un exemple. Aussi, l'objection peut s'exprimer comme étant:

"pas de G à la position n"  $\implies$  "pas exemple",  
 "pas de T à la position n+1"  $\implies$  "pas exemple",  
 ou encore: "pas de GT à la position n et n+1"  $\implies$  "pas exemple",

Remarque : Si un objet ne vérifie pas un fait  $a_i$ , alors cet objet ne vérifie pas les régularités dans lesquelles  $a_i$  est un terme. Si un objet vérifie une objection alors il est objecté, sinon il vérifie les disjonctions correspondantes aux objections.

La production des objections consiste à rechercher les disjonctions de faits qui doivent être vérifiés par chacun des objets candidat à être un exemple de la conjecture. Les objections générales sont liées par une relation d'ordre de spécialisation-généralisation. Tout couple d'objections possède une borne inférieure qui est l'objection dont la disjonction comprend l'ensemble des attributs utilisés pour construire le treillis. Face au nombre d'objections à gérer, notre *but* n'est pas de construire toutes les objections, mais plutôt de faire ressortir les objections pertinentes qui peuvent ensuite être examinées par l'usager. La recherche de toutes les disjonctions de faits est équivalent à trouver toutes les combinaisons de ces faits. C'est un problème intrinsèquement exponentiel [Garey & Johnson, 1987] et la combinatoire de la génération des objections rend l'utilisation d'heuristiques indispensable, afin de réduire cette complexité. Cette combinatoire est insuffisamment réductible par le seul biais du critère de minimalité. Nous générons seulement les disjonctions qui sont *minimales* et les *plus courtes*. Une disjonction est minimale si un sous-ensemble des termes qui la composent ne constitue pas déjà une disjonction générée. Elle est courte lorsque sa longueur (nombre de termes) est inférieure à un seuil donné. Nous allons adjoindre à ce *critère syntaxique*, un *critère sémantique* tel que une relation de préférence entre les faits.

**Paramètres d'entrée :**

- $P$  : l'ensemble des régularités (descriptions) pertinentes;
- $A^*$  : l'ensemble des attributs (faits) pertinents;
- $\theta_a$  : Seuil de justification;
- $\phi$  : la longueur maximale d'une disjonction;

**Algorithme :**

- (1) Ordonner les éléments de  $A^*$  suivant le nombre d'apparitions dans  $P$ ;
- (2) Sélectionner un sous-ensemble  $T$  des premiers éléments de  $A^*$   
(critère de préférence = nombre élevé d'apparitions dans les régularités);
- (3)  $O_g^c = \emptyset$ ;
- (4) Construire une disjonction  $c_i$  dans  $T$  de longueur inférieure à  $\phi$ ; (critère syntaxique)
- (5) Tester si la non vérification de  $c_i$  empêche la vérification de  $\theta_a\%$  de régularités;
- (6) Si oui, si en plus  $c_i$  est minimale alors rajouter la négation de  $c_i$  à  $O_g^c$ ;
- (7) repeter (4), (5), et (6) jusqu'à épuisement de toutes les disjonctions possibles.

L'algorithme peut utiliser seulement les faits présents dans les régularités pour construire les objections. Ces heuristiques font que l'ensemble des objections ne peut pas toujours rejeter tous les contre-exemples. Par conséquent, le fait de ne pas être objecté est une **condition nécessaire**, mais

pas **suffisante** pour être un exemple. L'ensemble  $O_g^C$  est constitué des objections les plus pertinentes, au sens où elles sont constituées de faits les plus courants de **P**.

Etant donné un objet  $o_x$ , le système vérifie que cet objet n'est pas objecté.

- Dans l'affirmative, l'objet  $o_x$  peut prétendre être un exemple si le système trouve une preuve empirique (vérifier que  $o_x$  vérifie suffisamment de régularités pertinentes).

- Dans le cas contraire, l'objection du système peut être réfutée ou acceptée comme valide par l'utilisateur. L'explication ici revient à justifier la réfutation de l'objet par l'absence de certains faits.

*Remarques:*

• La réfutation d'une objection générale du système par l'utilisateur va entraîner une révision des connaissances apprises. La validation ou la réfutation d'une objection doit être fonction de sa capacité à rejeter ou non les exemples et les contre-exemples du concept étudié.

• Si les objections générales permettent de savoir *pourquoi un objet n'est pas un exemple*, il ne peuvent fournir la réponse à la question "comment faire pour que cet objet soit un exemple". Aussi, nous introduisons l'*objection contextualisée* qui va chercher à identifier sur un objet, les modifications susceptibles de le transformer en un exemple. Elle permet aussi d'examiner les questions du genre "Pourquoi cet objet ne ressemble-t-il pas à celui-ci ?".

• De plus, l'utilisateur peut juger insuffisante l'explication fournie par le système dans cette phase.

### 3.2- Objections contextualisées

Dans la définition des objections contextualisées, intervient la notion de contexte qui est un objet de la conjecture. Une *objection contextualisée* sur un objet est ce qu'il est suffisant de modifier sur l'objet pour que l'agent le considère comme un exemple. L'objection contextualisée est la différence entre un objet et un exemple  $o_e$ . Construire les objections contextualisées à  $o_x$  relativement à l'exemple  $o_e$  revient à déterminer la différence minimale entre  $o_x$  et  $o_e$ , qui va caractériser le fait que  $o_x$  ne ressemble à  $o_e$ . Cette différence ou objection montre les modifications à effectuer sur  $o_x$  pour le considérer comme proche de  $o_e$ . Le nouvel objet,  $o_r$ , est l'**objet rectifié** de  $o_x$ . Le *principe* de construction consiste à générer des énoncés qui lorsqu'ils ne sont vérifiés pas par un objet, empêche celui-ci d'être *prouvé analogiquement* comme étant un exemple. L'objection contextualisée est la négation de cet énoncé. LEGAL les soumet à l'appréciation de l'utilisateur qui peut les valider ou les réfuter. Ces objections montrent les modifications à effectuer sur un objet pour le considérer comme proche de l'exemple du concept. Les faits décrivant l'exemple  $e$  sont utilisés pour produire les objections contextualisées.

L'ensemble  $C$  des objections contextualisées est constitué de faits qui s'ils sont rajoutés à l'objet  $o_x$ , le rendent similaire à l'exemple  $o_e$ . L'objet  $o_r$  qui est le rectifié de  $o_x$ , est construit en additionnant les faits décrivant  $o_x$  à ceux de l'ensemble  $C$ . Cette addition doit respecter la *contrainte sémantique*, à

savoir un fait et sa négation ne peuvent apparaître simultanément dans la description de  $o_r$ .

De même que pour les objections générales, la construction des objections contextuées est un problème dont la complexité est exponentielle. Les *heuristiques* utilisées rendent ce problème linéaire en fonction du nombre d'attributs pertinents, soit les éléments de  $A^*$ . Les heuristiques sont basées sur des critères syntaxiques (par exemple recherche longueur des objections) et sémantiques (par exemple, relation de préférence entre les attributs de  $A^*$ ).

**Paramètres d'entrée :**

- $P$  : l'ensemble des descriptions pertinentes;
- $A^*$  : l'ensemble des attributs (faits) pertinents;
- $f_e$  : l'ensemble des faits décrivant l'exemple  $o_e$ ;
- $f_x$  : l'ensemble des faits décrivant l'objet  $o_x$ ;
- $\delta$  : le seuil de similarité;

**Algorithme :**

- (1) Déterminer l'ensemble  $Z$  des faits de  $o_e$  n'appartenant pas à  $o_x$ ,  $Z = f_e - f_x$ ;
- (2) Ordonner les faits de  $Z$  suivant le nombre d'apparitions dans  $P$ ;
- (3)  $C = \emptyset$ ;
- (4) Extraire un fait de  $Z$  suivant l'ordre de réordonnement;
- (5) Ajouter ce fait à  $C$ ;
- (6) Ajouter les faits de  $C$  à la description initiale de  $o_x$ , pour construire  $o_r$ ;
- (7) Si  $o_r$  est similaire à  $o_e$ , alors aller à (10);
- (8) Sinon, si  $Z$  n'est pas vide alors extraire un autre fait de  $Z$ , et aller à (5);
- (9) Fin, je ne trouve aucune objection contextuée sur l'objet  $o_x$ .
- (10) Fin, la conjonction des faits recherchés est constituée d'éléments de  $C$ .

• Si l'objet  $o_x$  est analogiquement justifié, alors le système s'appuie sur l'exemple  $o_e$  analogue à  $o_x$  pour construire son explication. Celle-ci revient simplement à utiliser les faits communs aux deux objets,  $o_e$  et  $o_x$ , pour en déduire la conjonction minimale sur laquelle s'appuie cette ressemblance.

Mais le système peut aussi rechercher le contre-exemple,  $o_n$ , le plus proche de  $o_x$ . Ensuite il va chercher les faits présents sur  $o_x$  qu'il faudrait retirer, pour qu'il soit similaire à  $o_n$ . L'explication revient à justifier sa décision par la présence de ces faits dans la description de l'objet  $o_x$ .

• Si  $o_x$  est analogiquement refuté ou ambigu, alors le système s'appuie sur le contre-exemple  $o_n$  analogue à  $o_x$  pour construire son explication. Celle-ci revient simplement à utiliser les faits communs à  $o_n$  et  $o_x$ , pour en déduire la conjonction minimale sur laquelle s'appuie cette ressemblance.

Il est aussi possible de rechercher l'exemple,  $o_e$ , le plus proche de  $o_x$ , et de construire l'objection contextuée qui empêche l'objet  $o_x$  d'être proche de l'exemple  $o_e$  comme le montre le tableau 4.

Dans l'application de prédiction des sites de jonctions d'épissage, le contrôle par objections contextuées a été utilisé en deuxième passe, après le contrôle par analogie du raisonnement empirique. Dans ce deuxième niveau de contrôle, LEGAL essaie de construire une argumentation à sa décision,

qu'il soumet à la critique de l'usager. Par exemple, supposons que l'usager cherche à avoir une preuve du système sur sa décision à réfuter l'objet  $o_x$  ci-dessous, qui est par contre un exemple de site de jonction donneur. LEGAL, parce qu'il ne trouve pas d'exemple analogue à  $o_x$ , va donc construire une objection contextuée pour justifier sa décision.

```

 $o_x$  = "AGCCAGGGCA CTCACCAGGC TGCAAGAACA GTGCTGGGGT AAGAGGGGAG CGGGGG..."
 $o_r$  = "AGCCAGGGCA CTCACCAGGC TGCAAGAAAG GTGCTGGGGT AAGAGGGGAG CGGGGG..."
 $o_e$  = "GTGGGCAAGT GCCGAAGCGC AGGCATCAAG GTACTGGCCT CCCATCCTCC CCTCCA..."
  
```

L'objet  $o_x$  vérifie 4% de régularités, a un pourcentage de dispersion égal à 6%, et n'est proche d'aucun exemple prototypique. L'exemple prototypique  $o_e$  que nous avons choisi vérifie 16% de régularités et a un pourcentage de dispersion égal à 9%. LEGAL poursuit son explication relativement à  $o_e$ , en signalant que l'absence d'une purine (A ou G) à la position 29 et d'une guanine (G) à la position 30 est la source du rejet de l'objet  $o_x$ . L'objet rectifié  $o_r$  est obtenu en remplaçant dans la description de  $o_x$  les nucléotides "CA" dans les positions respectives 29 et 30 par les nucléotides "AG" ou "GG", on obtient 2 objets rectifiés  $o_r$ . Chacun de ces objets rectifiés  $o_r$  vérifie 15% des régularités et peut être prouvé empiriquement comme un exemple de la conjecture. LEGAL peut également essayer de poursuivre la recherche de l'objet rectifié de manière à ce qu'il soit fort ressemblant à l'exemple prototypique  $o_e$ .

Sur la base d'une telle explication, l'usager pourra donc valider ou critiquer l'argumentation de LEGAL. S'il maintient le fait que malgré tout il n'est pas d'accord et estime que l'objet est un bon exemple, le système devra alors essayer de modifier ses connaissances de manière à tenir compte de la remarque de l'usager. L'objet  $o_x$  est un exemple de site de jonction donneur. Donc bien que l'argumentation de LEGAL soit pertinente, l'usager peut la réfuter parce qu'il maintient sa croyance sur le statut de cet objet. Il peut également revenir sur son hypothèse et déclarer contre-exemple, un objet qu'il prenait pour un exemple, ou bien forcer au contraire le système à modifier son modèle afin de reconnaître un objet qu'il refuse de déclarer exemple. Le succès ou l'échec de la prédiction provoque nécessairement une acquisition de connaissances chez l'usager. L'échec de la prédiction provient alors :

- soit du fait de la non représentativité des objets de ce type dans l'ensemble des objets d'apprentissage. Dans notre exemple, il semble plus que la non représentativité de la classe d'objets proches de  $o_x$  soit la cause de l'échec de la prédiction de LEGAL.

- soit de l'incomplétude du langage de description.

- soit des paramètres d'induction; en effet les régularités générées peuvent être très générales, il est alors nécessaire de baisser les critères d'apprentissage.

L'utilisation des objections dans LEGAL permet l'introduction d'expressions déductives dans un contexte semi-empirique. Dans la conclusion de leur article, Michalski & Kodratoff [1990] pense que les futurs systèmes d'apprentissage doivent être capables d'expliquer ce qu'ils ont appris et ce qu'ils leur restent à apprendre. L'utilisation des objections est un moyen dans LEGAL d'avoir cette faculté.

#### 4- Conclusion

Nous avons montré comment en interagissant avec l'utilisateur, le système est capable de produire une succession d'explications (preuve empirique, objections générales, preuve analogique, objections contextualisées) de plus en plus explicites, plausibles et pertinentes (tableau 4). L'explication est donnée par la présence de faits ou ensembles de faits - minimaux et pertinents - qui justifient la décision de l'agent. L'objection joue un rôle explicatif prépondérant dans l'échec de la preuve. L'utilisateur peut confirmer la pertinence de l'argumentation, et dans ce cas il faut prévoir une gestion des objections pertinentes du point de vue de l'utilisateur. La validation ou la réfutation des arguments du système vont permettre de mettre à jour la connaissance apprise.

Il nous a semblé important de concevoir un système qui constitue sa connaissance à l'aide d'une double interaction : la recherche de l'agrément de l'utilisateur, et une meilleure prédiction-vérification sur les données. Dans notre démarche de conception, il ressort que le système est intelligent ou le devient si l'utilisateur lui donne les moyens de l'être. Nous pensons que les bonnes méthodes seront celles qui gagneront la confiance de l'utilisateur.

**Remerciement :** L'auteur a bénéficié du soutien financier de l'Association Française de lutte contre la Myopathie. Ce travail rentre dans le cadre du programme de recherche "GDR Informatique et Génômes". L'auteur remercie les relecteurs de versions antérieures de ce papier pour leurs commentaires.

#### Références

- BALACHEFF N., 1992, "Modélisation de l'utilisateur dans l'explication", *Actes 2èmes Journées Explication*, 17-19 Juin, PRC-GDR-IA du CNRS, Sophia Antipolis, p.235-236.
- BARBOUX C., SALLANTIN J., 1990, "Contrôle par objection d'une théorie semi-empirique", *Revue d'Intelligence Artificielle*, 4(2):29-41.
- BREZILLON P., 1992, "Intervention de l'utilisateur dans les explications", *Actes 2èmes Journées Explication*, 17-19 Juin, PRC-GDR-IA du CNRS, Sophia Antipolis, p105-129.
- CARBONELL J.G., 1989, *Artificial Intelligence*, Special volume on Machine Learning, 40(1-3), September, North-Holland.
- CHANDRASEKARAN B., SWARTOUT W., 1991, "Explanations in knowledge Systems: The Role of Explicit Representation of Design Knowledge", *IEEE Expert*, 6(3):47-50, June.
- CLANCEY W.J., 1983, "The epistemology of a rule-based expert system: a framework for explanation", *Artificial Intelligence*, 20(3):215-251.
- FELSENFELD G., 1985, "L'ADN", *Pour la Science*, Décembre, 98:30-40.
- GAREY M.R., JOHNSON D.S., 1987, *Computers and Intractability: a Guide to the Theory of NP-Completeness*, W.H. Freeman, San Francisco (CA).
- GIUNCHIGLIA F., WALSH T., 1992, "A theory of abstraction", *Artificial Intelligence*, 57:323-389.

- GANASCIA J.G., 1993, "TDis: an Algebraic Formalization", *Proc. of Intl. Joint Conference on Artificial Intelligence*, 2:1008-1013, Chambéry, France.
- HAUGELAND J., 1989, *L'esprit dans la Machine - Fondements de l'IA.*, Ed. Odile Jacob.
- KASSEL G., GREBOVAL M.H., GREBOVAL C., BOUCHER F., 1992, "Raisonnement au bon niveau d'abstraction pour produire des meilleures explications: une étude de cas", *Actes 2èmes Journées Explication*, 17-19 Juin, p147-161.
- KBS, 1992, *Improving the Use of Knowledge Base Systems with Explanations*, Août, sponsored by ECAI, organized by P.Brézillon, A.Cawsey, N.Filer, E.Nicolosi, C.Paris, W.Clansey, B.Chandrasekaran.
- KNOBLOCK C.A., MINTON S., ETZIONI O., 1991, "Integrating Abstraction and Explanation-Based Learning in PRODIGY", *Proc. of the A.A.A.I conference*, 2:541-546.
- KODRATOFF Y., 1993, "Recent Advances in Machine Learning", *The Intl. Journal of Pattern Recognition and Artificial Intelligence*, vol.7, p469-511.
- LAGRANGE M.S., RENAUD M., MEPHU NGUIFO E., SALLANTIN J., 1993, "Apprentissage automatique et typologie. PLATA: une expérience d'acquisition de connaissances dans le domaine de la céramique archéologique", Rapport Recherche LIRMM n°93-0105, Novembre.
- LAKATOS I., 1984, *Preuves et Réfutations*, Hermann Ed.
- LIQUIERE M., MEPHU NGUIFO E., 1990, "LEGAL:LEarning with Galois Lattice", *Actes des 5èmes Journées Françaises sur l'Apprentissage*, p93-113, Lannion, Avril.
- MEPHU NGUIFO E., 1992, "Improvement and Control of Similarity-Based Decision for Knowledge Acquisition", *Proc. of the first African Conference on Research in Computer Science*, Yaoundé(Cameroon), October 14-20, 1:173-184, Ed. M. Tchuenté, INRIA.
- MEPHU NGUIFO E., 1994a, "Une nouvelle approche basée sur le treillis de Galois, pour l'apprentissage de concepts", *Actes du 9ème congrès RFIA*. Paris, 11-14 Janvier, 2:35-46. Aussi dans *Revue Mathématiques, Informatique et Sciences Humaines*. n°124, Février.
- MEPHU NGUIFO E., 1994b, "Séquences Biologiques et Apprentissage Symbolique Automatique: Description, Caractérisation, et Applications", Rapport de recherche LIRMM n°94-0010, Janvier.
- MEPHU NGUIFO E., SALLANTIN J., 1993a, "Interactive Control in Symbolic-Numeric Modelling", *Proc. of 3rd European-Japanese Seminar on Information Modelling and Knowledge Bases*, Budapest (Hongrie), 31 Mai - 3 Juin, 2:389-407.
- MEPHU NGUIFO E., SALLANTIN J., 1993b, "Prediction of primate splice junction gene sequences with a cooperative knowledge acquisition system", *Proc. of the 1st Intl. Conference on Intelligent Systems for Molecular Biology*, Washington DC, July 7-9, Eds. L. Hunter, & al., AAAI/MIT Press, p292-300.
- MEPHU NGUIFO E., SALLANTIN J., 1993c, "A user-driven multi-strategy learning system applied to prediction of primate splice junction gene sequences", *Internal Report n°93-098*, October, LIRMM. Submitted to Machine Learning journal.
- MICHALSKI R.S., 1983, "A theory and methodology of inductive learning", *Machine Learning: an AI Approach*, T.M.Mitchell & al. eds, San Mateo, CA.
- MICHALSKI R.S., KODRATOFF Y., 1990, "Research in M.L. : Recent progress, classification of methods, and future directions", *Machine Learning: an AI Approach*, Kodratoff & Michalski eds, M.Kaufman, CA, p1-30.
- NOORDEWIJER M.O., TOWELL G.G., SHAVLIK J.W., 1991, "Training Knowledge-Based Neural Networks to Recognize Genes in DNA sequences", In *Advances in Neural Information Processing Systems*, vol.3, M. Kaufmann.

PARIS C.L., 1992, "Systèmes Experts Explicatifs", *Actes 2èmes Journées Explication*, 17-19 Juin, p3-23.

RUSSELL B., 1956, *The Principles of Mathematics*, Allen Unwin Ed., London.

WILLE R., 1992, "Concept Lattices & Conceptual Knowledge Systems", *Comp. Math. App.*, 23(6-9):493-515.