# Opérateurs naturels en Programmation Logique Inductive

**Fabien Torre**

**Céline Rouveirol**

*Laboratoire de Recherche en Informatique,*
*URA 410 du CNRS, Bâtiment 490,*
*Université Paris-Sud,*
*91405 - ORSAY Cedex*

RÉSUMÉ. *L'apprentissage en langage du premier ordre se ramène à un problème de recherche dans des espaces de taille très importante, voir infinie. Nous présentons dans ce papier les* relations naturelles *qui étendent la notion de relation de généralité : ce pré-ordre permet un élagage optimal par rapport à une propriété que doit satisfaire la définition de concept, que ce soit la correction et la complétude par rapport aux exemples, ou bien des restrictions du langage.*

*Ces relations définies, nous discutons l'existence d'opérateurs idéaux pour des espaces ordonnés par nos relations naturelles. Nous adaptons les résultats de [19] sur la non-existence d'opérateurs idéaux à ces relations naturelles ce qui nous permet de constater que les conditions de non-existence ne s'appliquent pas pour certaines relations naturelles. Cela nous amène à exhiber des opérateurs idéaux et, ainsi, dépasser les résultats de non-existence pour les espaces ordonnés par la $\theta$-subsomption.*

MOTS CLÉS : *programmation logique inductive, biais de langage, élagage dynamique, relations de généralité, opérateurs idéaux.*

## 1 Introduction

Un *problème de recherche* est la donnée d'un ensemble d'états (l'espace de recherche), un ensemble d'opérateurs, un état initial et un état final. En pratique, l'état final n'est pas toujours explicitement donné mais caractérisé par un ensemble de propriétés qu'il doit présenter.

La tâche d'apprentissage peut être vue comme un problème de recherche [6]. Ainsi, la *programmation logique inductive* (PLI) peut être formalisée en ces termes à partir d'exemples positifs du concept à apprendre, d'exemples négatifs et d'une éventuelle théorie du domaine. L'espace de recherche est l'ensemble des clauses définies. Les opérateurs de raffinement [14, 10, 18] permettent de modifier une hypothèse pour obtenir de nouvelles hypothèses supposées meilleures : de nouvelles hypothèses plus spécifiques dans le cas d'un opérateur *descendant*, plus générales pour un opérateur *ascendant*. L'état initial est construit à partir d'un exemple positif. D'après [8], l'état final (la solution) doit expliquer les exemples positifs (*complétude*), sans expliquer les négatifs (*correction*).

En PLI, il n'est pas envisageable d'explorer exhaustivement l'espace de recherche. Pour cette raison, plusieurs techniques ont été développées dans le but d'élaguer l'espace de recherche, soit avant la recherche (élagage statique), soit pendant (*élagage dynamique*).

Une méthode bien connue consiste à exploiter la relation de généralité sur l'espace de recherche [6]. Par exemple, si l'on se place dans une recherche ascendante par rapport à cette relation (qui va donc des hypothèses les plus spécifiques aux plus générales), et qu'une hypothèse couvre un exemple négatif, alors il n'est pas nécessaire de développer ses généralisations puisqu'elles présenteront assurément le même défaut.

Une seconde technique est d'utiliser ce que l'on appelle des *biais d'apprentissage* : il s'agit de contraintes supplémentaires sur la définition du concept recherchée. Les biais d'apprentissage ont été largement étudiés [9] et nous nous intéresserons plus particulièrement aux *biais de langage* qui posent des conditions sur la syntaxe de la définition. [7] met en avant la nécessité d'utiliser des biais au cours du processus d'apprentissage, en partie pour rendre le processus efficace mais, surtout, pour obtenir un résultat non trivial (apprentissage par cœur).

Malheureusement, lorsqu'une hypothèse n'est pas satisfaisante par rapport à un biais de langage, il n'est pas immédiat que l'on puisse cesser de développer ses raffinements sans plus de précautions. Si un seul des raffinements de cette hypothèse satisfait le biais, nous sommes contraints de poursuivre son développement. Si bien que l'on en vient à se demander si les biais de langage ont un rôle à jouer en cours d'apprentissage dans un espace ordonné par la relation de généralité et si l'on ne doit pas les réduire à des tests post-apprentissage. Quels sont les cas où l'on peut réellement s'épargner sans risque le développement d'une hypothèse ne satisfaisant pas un biais de langage ?

Dans ce papier, nous nous proposons d'utiliser au mieux toutes les propriétés connues sur le concept à apprendre. Au contraire des systèmes qui n'utilisent qu'une relation de généralité pour explorer et élaguer l'espace de recherche, l'idée est de prendre aussi en compte les biais de langage pour ordonner l'espace de recherche.

Nous allons considérer un ensemble de propriétés que devra posséder la définition du concept : ces propriétés englobent à la fois les biais de langage et les contraintes liées aux exemples. Ainsi, étant donné un ensemble de propriétés $\{P_1, \ldots, P_n\}$ à satisfaire, la spécification de l'état final de notre problème devient : l'hypothèse $H$ est solution si et seulement si $P_1(H) \wedge \ldots \wedge P_n(H)$. Pour traiter ce nouvel objectif, nous allons concevoir de nouveaux pré-ordres, appelés *relations naturelles*, qui autorisent l'élagage dynamique par rapport à des propriétés données : quand une hypothèse ne satisfait pas l'une des propriétés, toutes ses descendantes peuvent être élaguées sans pour autant prendre le risque de perdre une solution.

D'un point de vue plus théorique, des travaux en ILP ont étudiés différentes relations de généralité : la $\theta$-subsomption [12], la subsomption généralisée [1], la T-implication [5] et l'implication logique elle-même [11]. Pour travailler sur ces relations, plusieurs types d'opérateurs ont été caractérisés et, parmi eux, les opérateurs idéaux. Comme nous le verrons les exigences de ces opérateurs semblent très raisonnables. Malgré cela, il a été prouvé que de tels opérateurs ne pouvaient exister dans des espaces ordonnés par la $\theta$-subsomption [19] ou l'implication logique [20]. Il s'agira donc pour nous de savoir si les relations naturelles permettent l'apparition de tels opérateurs.

Le papier est organisé comme suit. Dans la section 2, nous introduisons la notion de *relation naturelle* d'une propriété, ce qui nous permet ensuite d'exprimer une condition nécessaire et suffisante pour que l'élagage dynamique soit possible. Nous nous interrogeons ensuite sur l'existence d'opérateurs idéaux (définis dans [19]) dans des espaces ordonnés par des relations naturelles. Dans ce cadre, nous reformulons à la section 3, les conditions de non-existence d'opérateurs idéaux [19, 20], et caractérisons les propriétés sur le concept qui permettront de dépasser ces conditions. Nous présentons brièvement, à la section 4, un opérateur idéal par rapport à nos relations naturelles (c'est-à-dire idéal et qui permet l'élagage). Nous concluons en resituant notre travail par rapport à des approches déjà étudiées en PLI. Une version plus formelle de ce papier pourra être trouvée dans [16], tandis que l'ensemble des résultats sont regroupés dans [17].

# 2 Élagage dynamique

Le but est ici de considérer le nombre minimal d'hypothèses sans pour autant prendre le risque de perdre une solution. Pour cela, nous nous autorisons à utiliser toutes les propriétés connues sur le concept à apprendre. Une hypothèse doit éventuellement être correcte, complète, à champ restreint, connectée [13] ou réduite [12] ; elle peut aussi respecter une limitation, inférieure ou supérieure, des quantités suivantes : nombre de variables existentielles, nombre de littéraux (taille de la clause), profondeur ou encore degré.

Nous devons donc garantir l'absence de risque et c'est cette notion que nous allons maintenant formaliser.

## 2.1 Propriétés privées

Considérons une hypothèse $H$ qui ne vérifie pas une propriété attendue. On n'arrêtera de raffiner $H$ que si l'on a la certitude qu'aucun descendant de $H$ ne satisfait cette propriété. Si ça n'est pas le cas, on doit poursuivre le raffinement de $H$ et donc considérer des hypothèses qui ne satisfont pas les propriétés requises. Il est alors clair qu'aucun élagage ne sera effectué par rapport à cette propriété et l'on a à parcourir tout l'espace de recherche.

Cette intuition est illustrée par la figure 1 ; nous faisons aussi apparaître sur cette figure que les opérateurs de raffinement peuvent être vus comme des relations sur

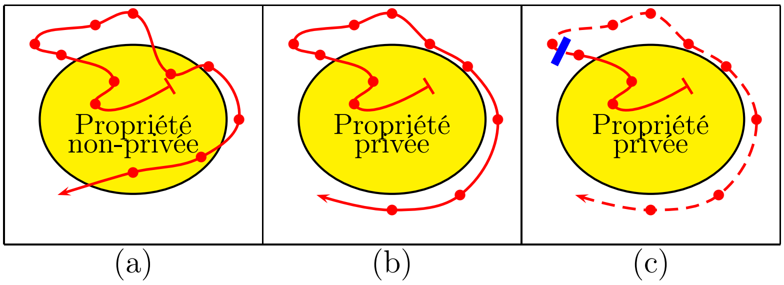



FIGURE 1 – Propriétés privée et non-privée. La figure (a) démontre que l'élagage sûr n'est pas possible par rapport à une propriété non-privée. Dans le cas (b), l'élagage est sûr et (c) exhibe sa réalisation pratique.

l'espace de recherche, et c'est cette vue que nous adopterons pour la suite. Nous sommes finalement amenés à la définition suivante.

**Définition 1 (propriété privée)** *On dira qu'une propriété est* privée *par rapport à une relation* $\mathcal{R}$ *si et seulement si*

$$\forall H, H' : \forall \left( H \mathcal{R} H' \wedge \overline{P(H)} \Rightarrow \overline{P(H')} \right)$$

**Exemple 1** *Considérons la propriété qui fixe une borne sur la taille des clauses*[1]. *Cette propriété est privée pour un opérateur qui ajoute des littéraux : si une clause dépasse la borne fixée par la propriété, lui ajouter des littéraux ne fournira que des clauses de plus grande taille, qui clairement ne satisferont plus jamais la propriété. À noter que cela est indépendant de la borne : c'est que signifie la quantification universelle utilisée dans la formule ci-dessus.*

Cette définition nous permet de repérer les relations (et donc les opérateurs) qui autoriseront l'élagage par rapport à une propriété. Nous allons maintenant caractériser plus précisément l'ensemble de ces relations.

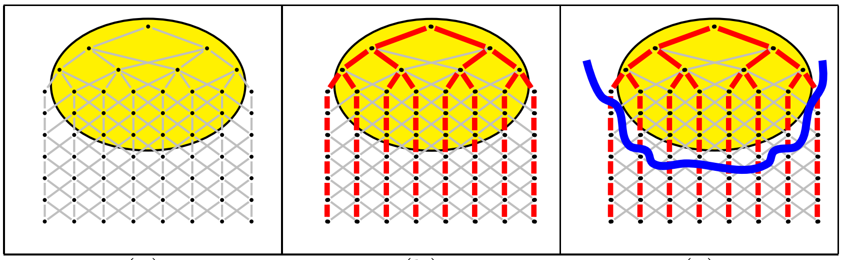
## 2.2 Relations naturelles

Beaucoup de relations peuvent rendre une propriété privée. Considérons, par exemple, la relation vide ou la relation identité : il est clair que ces relations n'ont aucun intérêt pratique pour le parcours d'un espace de recherche quel qu'il soit. Par conséquent, nous nous focalisons provisoirement sur l'autre extrême : quelles sont les relations les plus larges qui rendent une propriété privée ?

Un premier résultat est que cette relation est unique pour chaque propriété $P$, et nous l'appellerons *relation naturelle* de la propriété (nous la noterons $\geq_P$). Cette relation peut être simplement définie par

$$\geq_P = \left\{ (H, H') : \forall \left( \overline{\overline{P(H)} \wedge P(H')} \right) \right\}.$$

1. Nous considérons, ici et dans toute la suite, que la taille d'une clause correspond au nombre de littéraux dans son corps.



(a) (b) (c)

FIGURE 2 – Relation naturelle. La figure (a) montre l'organisation de l'espace produite par la relation naturelle, c'est-à-dire la relation la plus large qui rende la propriété privée. Figure (b), on choisit un sous-ensemble de la relation naturelle comme opérateur, qui à son tour rend la propriété privée. L'élagage est illustré figure (c).

Insistons, encore une fois, sur le fait que cette relation va être indépendante du paramètre utilisé dans la propriété.

Ensuite, il vient que toute relation qui rend une propriété privée est nécessairement incluse dans sa relation naturelle (voir figure 2).

Enfin, supposons que toute propriété peut s'exprimer sous la forme $f(H)\,\mathcal{R}\,k$ où :

— $f$ est une fonction de l'espace de recherche dans un domaine $\mathcal{D}_f$ ;

— $\mathcal{R}$ est un pré-ordre sur $\mathcal{D}_f \times \mathcal{D}_f$ ;

— $k$ est un élément de $\mathcal{D}_f$ (typiquement, $k$ est le(s) paramètre(s) de la propriété).

Dans ce cas, deux hypothèses $C$ et $D$ sont en relation naturelle si et seulement si $f(C)\,\mathcal{R}\,f(D)$.

**Exemple 2** *Revenons à la propriété qui limite la taille des hypothèses, que nous pouvons écrire* $|H| \leq k$. *Pour cette propriété, deux hypothèses* $H$ *et* $H'$ *seront en relation naturelle si et seulement si* $|H| \geq |H'|$. *Plus clairement, cela signifie que d'une hypothèse, on peut atteindre toutes les hypothèses de taille supérieure.*

**Exemple 3** *Pour la propriété de couverture* ($H \models e$)*, on a* $H \models H'$ *pour la relation naturelle. On retrouve donc l'obligation d'utiliser un ordre de généralité pour pouvoir élaguer par rapport au critère de complétude.*

Un autre résultat qui va nous intéresser par la suite, est que la relation naturelle d'une conjonction de propriétés est la relation obtenue par intersection des relations naturelles de chacune des propriétés.

Nous savons maintenant où chercher nos opérateurs de raffinement pour qu'ils aient un pouvoir élagant : à l'intérieur des relations naturelles des propriétés à satisfaire. Cette indication laisse encore beaucoup de possibilités (comme la figure 2 en donne l'intuition), et il est impératif de spécifier en plus un comportement attendu pour nos opérateurs. Nous avons choisi de nous intéresser aux opérateurs *idéaux* car ces opérateurs n'existent pas dans les cas classiques.

# 3 Conditions d'existence des opérateurs idéaux

Nous allons donc porter notre attention sur la catégorie d'opérateurs définie dans [19] : les opérateurs *idéaux*. Puis nous caractériserons les cas où ces opérateurs ne peuvent exister pour finalement établir que les relations naturelles sont favorables à l'apparition de ces opérateurs.

## 3.1 Définitions

La définition de l'idéalité [19] repose sur trois notions élémentaires. Un opérateur est *localement fini* si le raffinement d'une hypothèse produit un ensemble fini et calculable. Cela assure que l'opérateur a un intérêt pratique. Le raffinement d'une hypothèse est *strict* s'il ne fournit pas de clause équivalente à cette hypothèse. Ainsi, on peut assurer que l'opérateur fait réellement progresser la recherche. Enfin, l'opérateur est *complet* si les hypothèses comparables peuvent être liées par raffinements successifs. Si cette condition est vérifiée, on a la garantie que si une solution existe, elle sera atteinte par notre opérateur. Sur ces bases, la définition de l'idéalité est la suivante.

**Définition 2 (idéalité)** *Un opérateur est dit* idéal *s'il est à la fois localement fini, strict et complet.*

Malheureusement, il a été prouvé que de tels opérateurs n'existaient pas pour des espaces non restreints ordonnés par la $\theta$-subsomption ou l'implication logique. La non-existence d'opérateurs idéaux a deux raisons précisément identifiées : les chaînes infinies non couvertes et les ensembles couvrants infinis [19, 20, 18]. Pour appréhender ces deux problèmes, il est nécessaire de définir les notions de couverture et d'ensemble couvrant.

**Définition 3 (couverture)** *$C$ couvre $D$ si et seulement si $C > D$ et il n'existe pas de $E$ telle que $C > E > D$. On dit alors que $C$ est une* couverture supérieure *de $D$ et $D$ une* couverture inférieure *de $C$.*

**Définition 4 (ensemble couvrant)** *Un ensemble couvrant inférieur (resp. supérieur) d'une clause $C$ est un ensemble maximal de couvertures non équivalentes inférieures (resp. supérieures) de $C$.*

Ces définitions sont illustrées par la figure 3. Ces notions sont particulièrement pertinentes lorsque l'on s'intéresse à l'idéalité car il a été prouvé qu'un opérateur idéal appliqué à une hypothèse quelconque $H$ fournissait, au minimum, un ensemble couvrant de $H$ [18].

La possibilité de calculer l'ensemble couvrant de chaque hypothèse de l'espace de recherche ne suffit pas à garantir l'existence d'un opérateur idéal (on songera, en particulier, aux ensembles couvrants incomplets décrits dans [18]). Par contre, l'impossibilité de calculer l'ensemble couvrant d'une hypothèse va entraîner la non-existence d'opérateurs idéaux. Cette impossibilité peut avoir deux origines. Le premier cas est bien connu sur l'ensemble des réels : il est impossible de trouver un réel

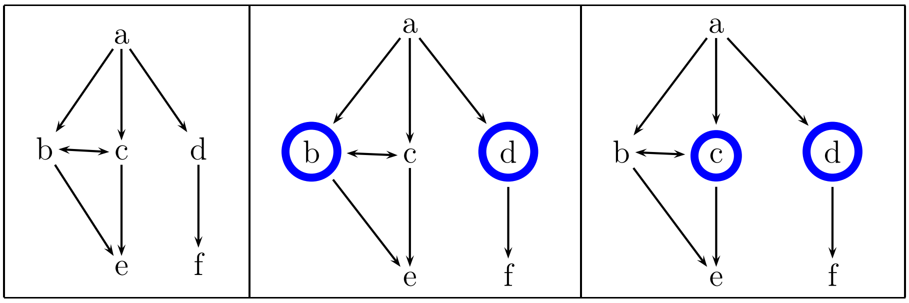



FIGURE 3 – Couvertures et ensembles couvrants. $a$ possède trois couvertures qui sont $b$, $c$ et $d$. Puisque $b$ et $c$ sont équivalents, ils ne peuvent pas apparaître dans le même ensemble couvrant. $a$ a donc deux ensembles couvrants possibles : $\{b, d\}$ et $\{c, d\}$.

immédiatement supérieur à un réel donné. Cela révèle l'impossibilité de faire un « plus petit pas » (on dit alors qu'il y a une chaîne infinie non couverte).

La seconde possibilité est que l'ensemble couvrant est infini.

La question est maintenant de savoir si ces phénomènes peuvent apparaître dans des espaces non-restreints ordonnés par nos relations naturelles. Nous allons nous restreindre au cas où l'on utilise une propriété liée aux exemples et un unique biais de langage. Pour les exemples, nous avons vu que la relation naturelle est l'implication logique mais nous utiliserons la $\theta$-subsomption qui lui est équivalente dans beaucoup de cas pratiques [4].

## 3.2 Chaînes infinies non couvertes

L'idée des chaînes infinies non couvertes est donnée à la figure 4 et un exemple sur les clauses est donné à la figure 5.

Il est important de se convaincre que l'existence d'une telle chaîne est directement liée à la relation utilisée pour ordonner l'ensemble et non pas à cet ensemble lui-même. En particulier, le phénomène décrit sur $\mathbb{R}$ n'est pas implacablement entraîné par le fait que $\mathbb{R}$ n'est pas dénombrable (considérez, par exemple, $\mathbb{R}$ doté de la relation « $x$ est plus petit que $y$ si $x$ est un préfixe de $y$ »). C'est précisément là qu'est notre espoir : nous travaillons sur le même espace mais nous allons l'organiser de manière peu conventionnelle : en utilisant les relations naturelles.

Le premier pas consiste à caractériser ces chaînes sous $\theta$-subsomption. Nous avons ainsi pu établir que certaines quantités augmentent *nécessairement* le long d'une chaîne (ascendante ou descendante) sous $\theta$-subsomption : nombre de variables existentielles, taille et nombre d'occurrences d'au moins un symbole de prédicat. Notons néanmoins que la profondeur ne croît pas dans une chaîne.

Parmi les quantités qui augmentent, seule la dernière ne correspond pas à un biais de langage classique en PLI. D'où, l'introduction d'un nouveau biais, appelé MAX-OCC, correspondant à la limitation de cette quantité. Ce nouveau biais doit empêcher un nombre d'apparitions quelconque d'un symbole de prédicat. Cela signifie que l'on peut poser soit une borne valable pour tous les symboles de prédicat, soit une borne différente pour chacun de ces symboles. Ce biais, bâti à partir de l'étude théorique des chaînes infinies non couvertes, a des utilisations bien pratiques. On pensera en

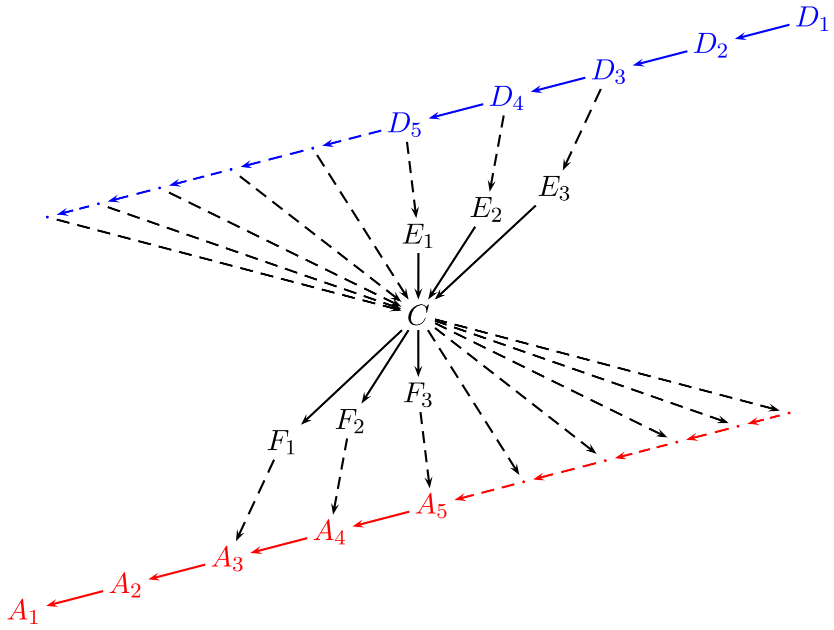



Figure 4 – Chaînes infinies non couvertes.

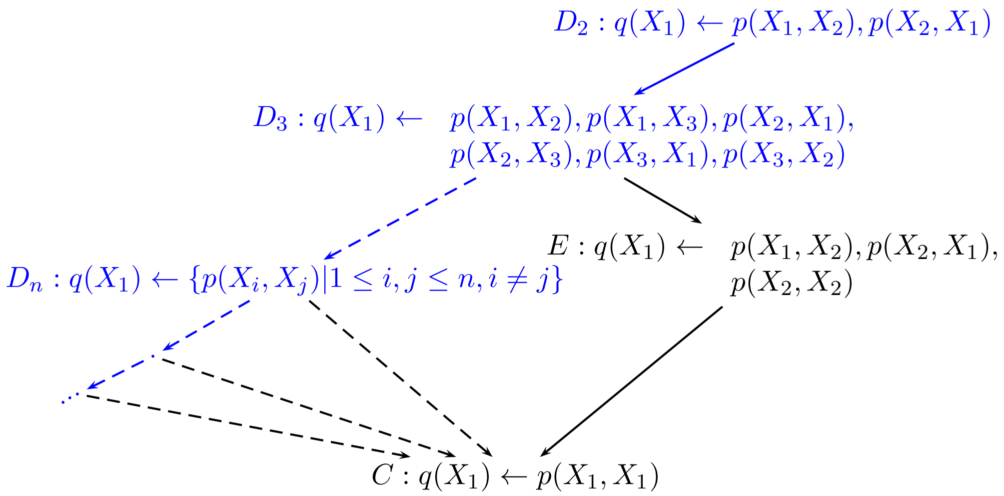



Figure 5 – Chaînes infinies non couvertes sous $\theta$-subsomption. $\{D_i\}_{i \geq 2}$ est une chaîne infinie descendante non couverte de $C$.

particulier à l'application liée à la mutagénicité [15], pour laquelle on peut avoir envie d'exprimer des contraintes comme : *la définition d'une molécule active doit faire intervenir, au plus, trois cycles benzéniques.*

Revenons à nos chaînes. Dans notre cadre, l'espace est ordonné par la conjonction de la $\theta$-subsomption et d'une relation naturelle $\geq_P$ définie par $C \geq_P D \Leftrightarrow f(C) \mathcal{R} f(D)$. Nous avons prouvé que l'existence d'une chaîne infinie non couverte dans un tel espace n'est possible que s'il existe une chaîne infinie non couverte sous $\theta$-subsomption telle que tous ses éléments ont même valeur par $f$.

Par suite, si le fait de fixer la valeur de $f$ empêche les quantités repérées de croître, alors il ne peut exister de chaînes infinies non couvertes dans notre espace. Avec ce critère, nous tenons une caractérisation des propriétés ne faisant pas apparaître de chaînes infinies non couvertes. Les candidates sont, au premier chef, les propriétés qui consistent à borner l'une des quantités dont on a vu qu'elle devait croître : limitation du nombre de variables, de la taille des clauses ou du nombre d'occurrences d'un symbole de prédicat. Malheureusement, les autres biais de langage cités dans l'introduction préservent les chaînes infinies non couvertes de la $\theta$-subsomption (en particulier, la chaîne de la figure 5 est toujours valable dans ces cas).

Nous nous sommes limités à une unique propriété mais l'argument reste valable pour un espace ordonné par une conjonction de taille quelconque : si l'une des propriétés empêche l'une des quantités pertinentes de croître, l'espace ne contiendra pas de chaînes infinies non couvertes. Ainsi, nous sommes certains qu'un espace ordonné par une conjonction (de taille quelconque) faisant intervenir la $\theta$-subsomption et la relation naturelle d'une des trois propriétés citées ci-dessus, ne présentera pas de chaînes infinies non couvertes.

## 3.3 Ensembles couvrants infinis

Envisageons maintenant le second cas empêchant l'existence d'opérateurs idéaux, celui où des clauses de l'espace possèdent un ensemble couvrant infini. Par définition, cela se produit si pour une hypothèse on peut trouver une infinité d'hypothèses qui lui soient immédiatement supérieures (ou inférieures) et incomparables entre elles.

Il est plus délicat d'identifier les propriétés amenant ce phénomène. Signalons simplement que, parmi nos propriétés, ce cas ne survient qu'une seule fois, pour la limitation sur le nombre de variables. Pour nous en convaincre, considérons la clause $C : q \leftarrow$ et les clauses $D_i$ définies par $D_i : q \leftarrow p(f^i(a))$. $C$ couvre chacun des $D_i$, les $D_i$ sont incomparables entre eux pour la $\theta$-subsomption et les $D_i$ ont même nombre de variables que $C$ (aucune). Par suite, $C$ a une couverture descendante infinie.

Le seul moyen de se débarrasser d'un ensemble couvrant infini d'une clause consiste à faire en sorte que cette clause ne soit plus en relation avec une infinité d'hypothèses de son ensemble couvrant. Dans notre exemple sur le nombre de variables, il suffit de poser que les hypothèses doivent être DATALOG (c'est-à-dire de profondeur nulle). Il n'y a plus alors qu'un seul $D_i$ dans l'ensemble couvrant de $C$, l'hypothèse $D_0 : q \leftarrow p(a)$.

L'idée est simplement que l'on peut annuler les ensembles couvrants infinis en considérant des combinaisons plus complexes et l'on a vu que l'ajout de nouvelles propriétés ne pouvaient amener de chaînes infinies non couvertes.

Par ailleurs, le problème ne se pose pas pour les propriétés autres que la limitation du nombre de variables car elles n'amènent pas d'ensembles couvrants infinis. Il nous reste comme « bonnes » propriétés : la limitation de la taille des clauses ou du nombre d'occurrences d'un prédicat et toujours le nombre de variables mais uniquement dans le cas ascendant. Dans chacun de ces cas, nous allons pouvoir donner un opérateur idéal.

# 4 Opérateurs naturels

Faute de place, nous ne décrivons ici qu'un opérateur pour un espace DATALOG [2], ordonné par les relations naturelles de la couverture et la limitation de la taille des clauses, c'est-à-dire par rapport à la relation suivante : $C$ et $D$ sont en relation si $C$ $\theta$-subsume $D$ et la taille de $C$ est inférieure ou égale à la taille de $D$.

**Définition 5** $\rho^{\theta}_{||}(C)$ *est obtenu par application des transformations suivantes.*

1. *Ajouter à $C$ un littéral ne contenant que des nouvelles variables.*
2. *Unifier deux variables $X_1$ et $X_2$ de $C$, telles que $C >^{\theta} C\{X_1/X_2\}$. Si la taille de la clause-résultat a diminué, ajouter un littéral ne contenant que des nouvelles variables.*
3. *Appliquer l'une des précédentes transformations (1, 2) sur une clause équivalente à $C$ ($\theta$-équivalente et de même taille).*
4. *Appliquer la transformation 1 sur un sous-ensemble de $C$ $\theta$-équivalent à $C$, l'un des nouveaux littéraux utilisant un symbole de prédicat qui n'apparaissait pas dans $C$.*

Cette opérateur est idéal (se référer à [17] pour une preuve formelle) et il respecte la relation naturelle de la limitation de la taille des clauses. Cela nous autorisera l'élagage dynamique par rapport à cette propriété, ainsi que par rapport à la complétude.

# 5 Bilan

Dans l'esprit, notre approche est assez semblable à celle de [14] qui contraignait les raffinements d'une hypothèse à être en nombre fini. Il utilisait pour cela une mesure *size* et des contraintes très fortes sur cette mesure (*size* est à valeur dans $\mathbb{N}$ et un nombre fini d'hypothèses ont même valeur par *size*). Dans notre cas, on contraint seulement les propriétés à être de la forme $f(H)\,\mathcal{R}\,k$. De plus, l'opérateur exposé dans [14] n'était pas idéal puisque incomplet [10, 18].

Nous avons adopté les définitions de [19] pour ce qui est la spécification des opérateurs de raffinement, tout comme d'ailleurs [2] qui définit une *relation de subsomption empirique* autorisant l'existence d'opérateurs idéaux ou encore [3] qui fait de même avec un pré-ordre dédié aux clauses DATALOG.

2. Cet opérateur peut facilement être étendu pour un espace absolument non restreint.

Notre démarche était toute différente car motivée par la nécessité d'élaguer l'espace de recherche. Nous avons défini la relation naturelle d'une propriété et montré que seul le respect de cette propriété pouvait permettre un élagage dynamique. Nous nous sommes ensuite interrogés sur la manière de construire un opérateur fondé sur cette relation et, en particulier, sur l'obtention d'un opérateur idéal. L'étude de ces opérateurs nous a permis de caractériser les propriétés favorables à leur existence et d'introduire un nouveau biais de langage ayant cette particularité.

Cela dit, nous envisageons d'étendre notre étude à tout type d'opérateurs jugés théoriquement intéressants et, plus particulièrement, aux opérateurs optimaux. Un autre prolongement de notre travail consistera à travailler avec d'autres relations de généralité que la $\theta$-subsomption : la subsomption généralisée pour prendre en compte une théorie du domaine et finalement l'implication logique. Notre approche s'inscrit dans la lignée des systèmes génériques comme $\mathcal{HAIKU}$ [9]. Le but final est d'implémenter nos opérateurs dans un tel système : un utilisateur pourra ainsi obtenir un opérateur de raffinement dédié à son problème, simplement à partir de la spécification de ce problème.

## Remerciements

Ce travail a été en partie financé par la CEE à travers le contrat Esprit LTR Project n°20237 (ILP$^2$). Nous tenons à remercier chaleureusement nos relecteurs anonymes et, parmi eux, Marc Champesme. Bien qu'ayant échoué à garder l'anonymat, il nous a été d'une aide très précieuse.

# Références

[1] W. Buntine. Generalized subsumption and its application to induction and redundancy. *Artificial Intelligence*, 36 :375–399, 1988.

[2] M. Champesme, P. Brézellec, and H. Soldano. Empirically conservative search space reductions. In L. De Raedt, editor, *Proceedings of the 5th International Workshop on Inductive Logic Programming*, pages 387–402. Department of Computer Science, Katholieke Universiteit Leuven, 1995.

[3] F. Esposito, A. Laterza, D. Malerba, and G. Semeraro. Refinement of datalog programs. In *Proceedings of the MLnet Familiarization Workshop on Data Mining with Inductive Logic Programming (ILP for KDD)*, pages 73–94, July 1996.

[4] Georg Gottlob. Subsumption and implication. *Information Processing Letters*, 24(2) :109–111, January 1987.

[5] P. Idestam-Almquist. Generalization of clauses under implication. *Journal of Artificial Intelligence Research*, 3 :467–489, 1995.

[6] T. M. Mitchell. Generalization as search. *Artificial Intelligence*, 18 :203–226, 1982.

[7] T. M. Mitchell. The need for biases in learning generalizations. In *Readings in Machine Learning*. Morgan Kaufmann, 1991.

[8] S. Muggleton and L. De Raedt. Inductive logic programming : Theory and methods. *Journal of Logic Programming*, 19 :629–679, 1994.

[9] C . Nédellec, C. Rouveirol, H. Adé, F. Bergadano, and B. Tausend. Declarative bias in ILP. In L. De Raedt, editor, *Advances in Inductive Logic Programming*, pages 82–103. IOS Press, 1996.

[10] T. Niblett. A note on refinement operators. In P. B. Brazdil, editor, *Proceedings of the 6th European Conference on Machine Learning*, volume 667 of *Lecture Notes in Artificial Intelligence*, pages 329–335. Springer-Verlag, April 1993.

[11] S.H. Nienhuys-Cheng and R. de Wolf. Least generalizations and greatest specializations of sets of clauses. *Journal of Artificial Intelligence Research*, 4 :341–363, 1996.

[12] G. Plotkin. A note on inductive generalization. In *Machine Intelligence*, volume 5. Edinburgh University Press, 1970.

[13] J. R. Quinlan. Learning logical definitions from relations. *Machine Learning*, 5(3) :239–266, 1990.

[14] E. Y. Shapiro. Inductive inference of theories from facts. Technical Report 192, Yale University Department of Computer Science, February 1981.

[15] A. Srinivasan, S. Muggleton, R. D. King, and M. J. E. Sternberg. Mutagenesis : ILP experiments in a non-determinate biological domain. In S. Wrobel, editor, *Proceedings of the 4th International Workshop on Inductive Logic Programming*, volume 237 of *GMD-Studien*, pages 217–232. Gesellschaft für Mathematik und Datenverarbeitung MBH, 1994.

[16] F. Torre and C. Rouveirol. Natural ideal operators in inductive logic programming. In M. van Someren and G. Widmer, editors, *9th European Conference on Machine Learning (ECML'97)*, volume 1224 of *LNAI*, pages 274–289. Springer-Verlag, April 1997.

[17] F. Torre and C. Rouveirol. Private properties and natural relations in inductive logic programming. Technical Report 1118, Laboratoire de Recherche en Informatique, Université Paris Sud, July 1997.

[18] P.R.J. van der Laag. *An Analysis of Refinement Operators in Inductive Logic Programming*. PhD thesis, Erasmus Universiteit, Rotterdam, the Netherlands, 1995.

[19] P.R.J van der Laag and S.H. Nienhuys-Cheng. Existence and nonexistence of complete refinement operators. In F. Bergadano and L. de Raedt, editors, *Proceedings of the 7th European Conference on Machine Learning*, volume 784 of *Lecture Notes in Artificial Intelligence*, pages 307–322. Springer-Verlag, April 1994.

[20] P.R.J. van der Laag and S.H. Nienhuys-Cheng. A note on ideal refinement operators in ILP. In S. Wrobel, editor, *Proceedings of the 4th International Workshop on Inductive Logic Programming*, volume 237 of *GMD-Studien*, pages 247–262. Gesellschaft für Mathematik und Datenverarbeitung MBH, September 1994.