MATHÉMATIQUES ET SCIENCES HUMAINES

DOMINIQUE GUIN

# Une modélisation mathématique de la compréhension des énoncés additifs

*Mathématiques et sciences humaines*, tome 120 (1992), p. 49-77

<http://www.numdam.org/item?id=MSH_1992__120__49_0>

*Math. Inf. Sci. hum.*, (30e année, n°120, *1992*, p.49-77)

# UNE MODÉLISATION MATHÉMATIQUE DE LA COMPRÉHENSION DES ÉNONCÉS ADDITIFS

Dominique GUIN[1]

RÉSUMÉ — *Cet article fait suite à l'article paru dans le volume 113 de cette revue (D. Guin [15]) où nous avons mis en évidence certains processus cognitifs élémentaires dans l'activité de compréhension d'énoncés additifs, puis proposé une modélisation de la compréhension permettant de prendre en compte ces processus. Dans cette modélisation basée sur la notion d'opérateur, nous avons distingué deux étapes : la compréhension linguistique de l'énoncé et la réduction à un prototype. Nous utiliserons des formalismes mathématiques (Grammaire Applicative, Logique Combinatoire) dans cette modélisation inspirée du modèle de la Grammaire Applicative et Cognitive de J.-P. Desclés pour définir les archétypes cognitifs du domaine des énoncés additifs. Les lois exprimées à l'aide de combinateurs (lois linguistiques, lois d'intégration lexicales et lois sémantiques) nous permettront de réaliser par une compilation de représentations fonctionnelles une simulation des deux étapes du processus cognitif de compréhension d'un énoncé additif.*

SUMMARY — A mathematical modeling of the understanding process of additive word problems.
*This article is following an article published in volume 113 of this journal (D. Guin [15]) where we have pointed out some cognitive elementary processes in the understanding activity of additive word problems, then suggested a modeling which enables us to take into account these processes. In this modeling based on the operator notion, we have distinguished two steps : understanding the terms of the problem and reducing it to a prototypical problem. We use mathematical formalisms (Applicative Grammar and Combinatorial Logic) in this modeling inspired by the Applicative and Cognitive Grammar model of J.-P. Desclés to define the cognitive archetypes of the additive word problems domain. Laws expressed with combinators (linguistic laws, laws of lexical integration and semantic laws) enable us to realize, by a compilation of functional representations, a simulation of the two steps of the cognitive process of understanding an additive word problem.*

## AVANT-PROPOS

Nous préciserons tout d'abord notre conception de la simulation par ordinateur d'un processus de compréhension. Nous rappellerons succinctement l'architecture cognitive proposée dans (J.-P. Desclés et alii [11]) qui est fondée sur la fonctionnalité et la compilation des représentations intermédiaires : l'objectif visé n'est pas de simuler directement le comportement humain par un programme informatique, mais de construire une modélisation théorique et des représentations intermédiaires directement exécutables par les machines. La simulation rendra compte uniquement d'une *analogie entre les stratégies* de traitement.

Les processus mis en jeu au niveau fonctionnel sont définissables *indépendamment* des représentations du niveau sémantique et des processus cognitifs eux-mêmes : c'est ce que J.-P. Desclés désigne par la *compilation* des représentations fonctionnelles. Nous avons dans un

[1] Laboratoire d'Études et de Recherches sur l'Enseignement Scientifique (Université Montpellier II).

premier temps défini un vocabulaire cognitif, c'est-à-dire un corps de concepts pertinents pour la description et l'analyse des faits cognitifs, et expliqué en quoi la notion d'opérateur est bien adaptée à cette description (D. Guin [15]). Dans cet article, après avoir présenté les formalismes mathématiques nécessaires, nous montrerons que cette notion permet effectivement de décrire les différentes étapes mises en évidence dans le processus de compréhension en définissant les primitives sémantiques et archétypes cognitifs du domaine additif, puis en précisant l'architecture du niveau fonctionnel.

## 1. INTRODUCTION

Notre but est de modéliser la démarche de compréhension d'un énoncé de problème additif, comme par exemple :

*" Pierre a joué deux parties de billes. A la première partie il a gagné 16 billes, à la seconde partie il en a gagné 9. Que s'est-il passé en tout ? "* [1]

Plus précisément, nous voulons modéliser le comportement humain de l'élève expert[2] au cours de la lecture d'un tel énoncé, afin de pouvoir réaliser ensuite une simulation sur ordinateur du traitement effectué. C'est dans une perspective non seulement cognitive, mais aussi didactique qu'il nous faut expliciter le processus dynamique d'élaboration par l'élève de sa représentation du problème. A plus long terme, notre but est de modéliser l'activité de résolution d'un problème additif en vue de la réalisation d'un tutoriel intelligent.

Nous avons mis en évidence dans (D. Guin [15]) la nécessité de prendre en compte dans la modélisation de la compréhension les processus cognitifs élémentaires tels que l'*interprétation* du langage naturel et l'incidence des caractéristiques linguistiques dans cette activité, la *paraphrase*, la reconnaissance d'*invariants opératoires* (G. Vergnaud [21]). Le concept d'*opérateur*, qui est à la base du modèle de la Grammaire Applicative Universelle, permet d'appréhender ces processus élémentaires d'une part dans le processus de *compréhension linguistique*, et d'autre part dans celui de *réduction à un prototype*. Nous avons présenté cette dernière étape dans l'article précité, les types de problèmes intervenant dans cette modélisation ont été définis à partir d'opérateurs définis sur $\mathbb{N}$, et une expression en langue naturelle des exemples prototypiques y a été proposée.

Nous détaillerons dans cet article la modélisation de la compréhension linguistique inspirée du modèle de la Grammaire Applicative Universelle de S. K. Shaumyan et de l'extension de ce modèle réalisée par J.-P. Desclés, le modèle de la grammaire Applicative et Cognitive. L'étape de compréhension linguistique est celle de la prise d'information dans l'énoncé additif, qui doit permettre d'extraire la *signification* des énoncés additifs.

Nous rappellerons tout d'abord succinctement le formalisme du modèle de la Grammaire Applicative Universelle que l'on peut enrichir d'un niveau *cognitif* en y introduisant des représentations cognitives qui se veulent "indépendantes" des contraintes sémiotiques du langage naturel : nous présenterons ce modèle, désigné par Grammaire Applicative et Cognitive dans (J.-P. Desclés [9]), au paragraphe 2. Nous développerons ensuite le formalisme applicatif du domaine des problèmes additifs qui nous intéresse (paragraphe 3), nous pourrons alors l'étendre aux représentations cognitives du domaine additif (paragraphe 4). Cette modélisation de la compréhension *linguistique* s'intègre dans une modélisation complète de la compréhension des problèmes additifs dont nous présenterons les différentes étapes au paragraphe 5.

---

[1] Les énoncés français sont extraits de G. Vergnaud ([20]).

[2] Au sens de J.R. Anderson : expliciter comment l'élève " idéal " devrait se comporter dans cette situation.

## 2. FORMALISME DE LA GRAMMAIRE APPLICATIVE ET COGNITIVE

Notre modélisation s'inspire des recherches du linguiste S.K. Shaumyan qui utilise la Logique Combinatoire de H.B. Curry pour une analyse des transformations interphrases dans les langues naturelles. Il propose un modèle, appelé *Grammaire Applicative Universelle*, qui représente les phrases sous forme d'expressions applicatives, c'est-à-dire sous forme d'agencements combinatoires d'opérateurs et d'opérandes de différents types.

### 2.1. Grammaire Applicative Universelle (GAU)

2.1.1. Le modèle de la Grammaire Applicative Universelle comporte un métalangage de description désigné par langage *génotype* où l'on souhaiterait formuler tous les *invariants sémiotiques* constitutifs des langues naturelles (entités et opérations universelles). Ce système est *universel*, puisque l'on peut établir un morphisme du langage génotype vers les différentes langues phénotypes. Sous certaines hypothèses, il existe une *réduction* d'une expression du langage génotype (grâce à la *logique combinatoire*) à une *forme normale*. Ces formes normales constituent un sous-langage, ce sont les représentants des classes d'équivalences qui sont des familles paraphrastiques.

Dans ce paragraphe, nous reprenons le formalisme présenté dans cette même revue (J.-P. Desclés [8]) ; pour plus de détails, l'on pourra également consulter (J.-P. Desclés [9]). Les systèmes applicatifs sont organisés avec une opération binaire notée ' : ' (ou [ , ]) et appelée *application.* L'application de l'opérateur f à l'opérande a donne pour résultat b. Le signe ' ≥ ' exprime la relation entre l'opération d'application à effectuer et le résultat de l'opération effectuée. Nous avons donc :

$$f : a \geq b$$

Cette notion est largement utilisée dans le *lambda-calcul* de Church qui précise les règles de substitution d'un opérande à une variable liée par l'opération d'abstraction (A. Church [4]).

Un *système applicatif* est alors la donnée de :

- un ensemble d'*objets atomiques* et *d'opérateurs atomiques* (constants ou variables),
- l'*opération d'application* qui permet de *construire* des expressions dérivées à partir des atomes (objets et opérateurs).

Nous considérerons ici un système applicatif muni d'un *produit cartésien fini* qui permet de construire des *n-uples* d'objets $< a_1 , a_2 ,..., a_n >$. Nous dirons que l'opérateur est à une, deux ou n *places* si son opérande est respectivement simple, un couple d'objets ou un n-uple d'objets.

Puisque nous devons manipuler différents types d'opérateurs, nous définissons un système applicatif *typé* avec produit cartésien fini à partir d'un ensemble S de *sortes* d'expressions de base (entiers naturels, propositions, termes nominaux etc...). L'ensemble *Typ* [S] des types fonctionnels des opérateurs est défini récursivement à l'aide de l'opérateur binaire $\mathfrak{F}$ formateur de types fonctionnels (D. Guin [15], p 17). L'assignation d'un type x à une expression X sera notée ' x : X '.

2.1.2. Nous présentons une adaptation de la notion d'*algèbre hétérogène (ou Σ-algèbre )* introduite par G. Birkhoff ([3]) au formalisme applicatif (J.-P. Desclés [6]). Nous considérons maintenant un système applicatif particulier, un *S-schéma* (d'opérateurs) Σ : c'est la donnée d'un ensemble d'opérateurs élémentaires de différents types. Chaque opérateur $\sigma$ est caractérisé par un couple $<v, \mathfrak{F}(\mu)\, s>$ où v est le nom de l'opérateur, $\mu$ un élément du monoïde libre engendré par S, et $\mathfrak{F}(\mu)\, s$ est un type de *Typ* [S]. Afin de pouvoir interpréter les opérateurs $\sigma$

par des fonctions ensemblistes $\sigma'$, nous utilisons la structure algébrique de *$\Sigma$-algèbre (à sortes dans S)*, qui est la donnée :

- d'un S-schéma d'opérateurs $\Sigma$,
- d'une famille $E^{(s)}$ d'ensembles sous-jacents (à un élément s de S, on associe un ensemble $E^s$ ayant une seule sorte s d'objets),
- d'une fonction interprétative u qui associe, à chaque opérateur $\sigma$ caractérisé par un couple $<v, \mathcal{F}\mu) s>$, une opération $\sigma'_v = u(\sigma) \in \mathrm{Hom}\ (E^{\mu}, E^{s})$.

On montre (J.-P. Desclés [6]) qu'il existe un *objet initial* **A** $[\Sigma]$ dans la catégorie des $\Sigma$-algèbres, c'est-à-dire que pour toute $\Sigma$-algèbre **A** dont l'ensemble sous-jacent est A, il existe un unique *$\Sigma$-homomorphisme* u' de **A** $[\Sigma]$ dans **A**, tel qu'on ait le diagramme commutatif suivant :

$$\begin{tikzcd} \Sigma \arrow[r] \arrow[dr] & \mathbf{A}\,[\Sigma] \arrow[d, dashed, "G\,(u')"] \\ & \mathbf{A} \end{tikzcd}$$

L'algèbre **A** $[\Sigma]$, dite $\Sigma$-algèbre libre engendrée par $\Sigma$, constitue une *"syntaxe abstraite"* indépendante de toute interprétation. La *sémantique* interprète d'abord les opérateurs de base de $\Sigma$, l'interprétation est ensuite *"guidée par la syntaxe"* puisqu'elle est réalisée à partir de u'.

2.1.3. Pour être en mesure de préciser le calcul des connexions syntaxiques, l'on étend légèrement le système précédent en introduisant des types gauche et droit, d'où, pour un type t donné, un type g(t) et d(t). L'on se donne, en outre, une opération * correspondant à la formation du type "produit cartésien fini", et l'on ajoute deux schémas de simplification dans Typ [S] qui expriment les connexions des expressions contigües combinées entre elles selon la structure applicative opérateur-opérande, et compatibles avec l'ordre linéaire des expressions linguistiques, de la forme :

$$(\mathcal{F}\, g(t_1)\, t_2) * t_1 \approx t_2 \text{ pour tout } t_1, t_2 \text{ dans Typ [S]}$$
$$t_1 * (\mathcal{F}\, d(t_1)\, t_2) \approx t_2 \text{ pour tout } t_1, t_2 \text{ dans Typ [S]}$$

L'on peut aussi utiliser une notation multiplicative et fractionnaire :

$$(t_2 / g(t_1)) * t_1 \approx t_2 \qquad \text{ou} \qquad (t_2 /_g t_1) * t_1 \approx t_2$$
$$t_1 * (t_2 / d(t_1)) \approx t_2 \qquad \text{ou} \qquad t_1 * (t_2 /_d t_1) \approx t_2$$

Ce système est une généralisation du formalisme des fractions formelles proposé par K. Ajdukiewicz et Y. Bar-Hillel ([1] ; [2]) qui est à la base des grammaires catégorielles.

2.1.4. La *logique combinatoire* née d'une recherche menée par M. Schönfinkel ([19]) vise à éliminer complètement la notion de variable liée en faisant appel à des opérateurs abstraits, les *combinateurs*, et a été développée essentiellement par H.B. Curry ([5]). Les combinateurs sont destinés à combiner des expressions applicatives entre elles pour construire une nouvelle expression applicative. L'action sur leur opérande est définie par une règle de réécriture appelée *β-réduction* formulée avec des variables nécessairement libres. Cette action définit en quelque sorte la *signification* du combinateur indépendamment de toute interprétation externe. Pour plus de détails, l'on pourra se référer aux articles précédemment parus dans cette revue (J.-P. Ginisti [13] ; J.-P. Desclés [8]).

*Exemples :*

| | |
|---|---|
| L'identificateur **I** | $\mathbf{I}\, X \geq X$ |
| Le compositeur **B** | $\mathbf{B}\, X\, Y\, Z \geq X\,(Y\, Z)$ |
| Le répétiteur **W** | $\mathbf{W}\, X\, Y \geq X\, Y\, Y$ |
| Le permutateur **C** | $\mathbf{C}\, X\, Y\, Z \geq X\, Z\, Y$ |
| L'éliminateur **K** | $\mathbf{K}\, X\, Y \geq X$ |
| Le combinateur **S** | $\mathbf{S}\, X\, Y\, Z \geq X\, Z\,(Y\, Z)$ |
| Le combinateur $\Phi$ | $\Phi\, X\, Y\, Z\, U \geq X\,(Y\, U)\,(Z\, U)$ |
| Le combinateur d'intrication $\Psi$ | $\Psi\, X\, Y\, Z\, U \geq X\,(Y\, Z)\,(Y\, U)$ |

THÉORÈME (J.-P. Desclés [9], p 163) : Tous les combinateurs précédents peuvent être exprimés en fonction des combinateurs **S** et **K** pris comme combinateurs de base.

En outre, chaque combinateur est représentable par une *lambda-expression* [1]. Les systèmes du λ-calcul et de la logique combinatoire sont équivalents si on leur adjoint le principe d'*extensionnalité* (J.-P. Desclés [9], p 163) :

[*Ext*] Si pour toute expression combinatoire U, on a XU = YU, alors X = Y.

On peut définir la *puissance n-ième* d'un combinateur $\chi$ régulier par :

$$\chi^{1} \equiv \chi \;,\quad \chi^{n+1} \equiv \mathbf{B}\, \chi\, \chi^{n}$$

*Exemple :* $\mathbf{K}^2\; X\, Y\, Z \geq X$

De même, on peut définir l'action à distance d'un combinateur par :

$$\chi_{(0)} \equiv \chi \,,\; \chi_{(1)} \equiv \mathbf{B}\, \chi \;,\; \chi_{(n+1)} \equiv \mathbf{B}\; \chi_{(n)} \,,$$

$\chi_{(k)}$ *diffère l'action* du combinateur $\chi$ de k pas et se comporte comme $\chi$ à partir de la k+1ème opérande , il agit donc effectivement sur la k+2 ème opérande :

$$\chi_{(k)} = \mathbf{B}^{k}\, \chi$$

*Exemple :* $\mathbf{C}_{(2)}\, X\, Y\, Z\, T\, U = \mathbf{B}^2\, \mathbf{C}\, X\, Y\, Z\, T\, U \geq \mathbf{C}\,(X\, Y\, Z)\, T\, U \geq X\, Y\, Z\, U\, T$

Les combinateurs sont composables entre eux dans une structure de *monoïde.* Ils donnent la possibilité de construire des opérateurs complexes à partir d'opérateurs élémentaires.

2.1.5. Le premier type de réduction effectué grâce aux β-réductions des combinateurs est la réduction d'une expression applicative à une forme *normale* utilisant des lois *linguistiques* (interprétation d'un prédicat transitif actif, réduction grammaticale passif-actif etc ...). L'*existence* d'une forme normale pour une expression applicative n'est pas assurée. L'*unicité* est une conséquence du théorème de Church-Rosser (J.-P. Desclés [8]). Cependant l'existence, pour les expressions applicatives *typées,* est une conséquence du théorème de Sanchis. Deux expressions applicatives sont *équivalentes* si elles se réduisent à une *même forme normale.*

---

[1] Par exemple, pour le combinateur C : $\lambda\, x\, y\, z\, .\, x\, z\, y$.

Chaque phrase est alors représentée par une expression applicative typée d'un langage applicatif $L_1$ engendré à partir des opérateurs linguistiques de différents types, des combinateurs et des prédicats complexes. L'ensemble des formes normales constitue un sous-langage $L_0$. Les formes normales sont les représentants des classes d'équivalences qui sont des familles paraphrastiques.

Nous allons voir dans le paragraphe suivant comment il est possible de continuer la décomposition des éléments du sous-langage $L_0$ en analysant les prédicats lexicaux à l'aide de *primitives sémantiques* combinées entre elles par les combinateurs de la logique combinatoire.

2.2. Les archétypes cognitifs

Pour une analyse détaillée l'on pourra consulter (J.-P. Desclés [9]). Dans le domaine des représentations des connaissances J.-P. Desclés propose de recourir à des formalismes applicatifs pour représenter la *signification* des prédicats linguistiques au moyen de combinateurs qui agencent les primitives sémantiques entre elles. La signification de chaque phrase, qui est représentée par une expression applicative, est analysée grâce au processus de réduction à un archétype cognitif, exprimable dans un langage applicatif. Les *archétypes cognitifs* constituent un système de représentation des connaissances mettant en jeu différents types d'opérateurs (primitives sémantiques, opérateurs ensemblistes, etc...). Ils décrivent des relations prototypiques qui tiennent entre des objets, des lieux, et plus généralement entre des situations. Une propriété fondamentale de ce système de représentation est son *indépendance* par rapport à l'encodage dans une langue naturelle donnée. Les archétypes cognitifs semblent bien adaptés à la représentation des connaissances nécessaires à l'activité de compréhension, de paraphrasage et de raisonnement : bien que leur pertinence cognitive ne soit pas assurée, ils permettent de simuler complètement les mécanismes de compréhension et raisonnement en explicitant toutes les étapes (création, suppression, modification d'objets...).

La notion de *primitive sémantique* est due à R. Schank ([18]). Son système limité aux verbes est basé sur l'hypothèse de la *dépendance conceptuelle* : les verbes de toutes les langues sont exprimables au moyen d'un nombre restreint de primitives sémantiques. Les primitives sémantiques se veulent être des relateurs et opérateurs invariants par rapport à la diversité des langues, à portée universelle *indépendante* des moyens d'expression *linguistique*. Elles ne sont pas relatives à une langue déterminée : ce sont des entités situées dans le métalangage. Ces universaux abstraits sont nécessaires pour permettre de décrire la signification des items lexicaux. Il est ainsi possible d'exprimer les prédicats en fonction de combinateurs et de primitives sémantiques. On peut décrire les *mécanismes intégratifs* (D. Piotrowski [16]) qui ont pour fonction de "composer" entre eux les unités sémantiques organisées dans la structure cognitive.

On associe à une expression applicative une *combinaison de primitives sémantiques*, la forme normale associée à cette expression applicative typée définit un archétype cognitif qui analyse la signification de l'expression applicative. C'est la loi d'*intégration lexicale* qui exprime comment composer les unités sémantiques, elle décrit la *compilation* de l'expression linguistique dans un système de repésentation sémantique :

$$\textbf{Prédicat} \cong \mathsf{X}\ [\textbf{primitives sémantiques}]$$

où $\mathsf{X}$ désigne un combinateur qui représente un programme d'intégration des primitives sémantiques qui figurent à sa droite dans l'expression.

*Exemple :* La phrase "*la feuille entre dans la pièce*" a pour représentation applicative :

"entre-dans" $T^2\ T^1$ où $T^2 := $ la pièce, $T^1 := $ la feuille

On peut exprimer le prédicat "entre-dans" en fonction de primitives sémantiques dont nous rappelons les définitions (J.-P. Desclés [9]) :

$e_0$ : relateur de *localisation* exprime qu'un objet est localisé par rapport à un lieu ou qu'un lieu est localisé par rapport à un autre lieu [1].
in (resp ex) : sont des opérateurs de détermination *topologique* d'un lieu, ils déterminent l'intérieur (resp l'extérieur) de ce lieu.

MOUVT : exprime un *mouvement* affectant une entité, qui la fait passer d'un lieu à un autre.

L'analyse sémantique de "entre-dans" est exprimée par la loi (*) d'*intégration lexicale* des primitives sémantiques MOUVT, $e_0$, in et ex décrite à l'aide des combinateurs **B**, **Φ** et $\psi$ :

$$\text{" entre-dans "} \cong \psi\,(\mathbf{B}\ \Phi\ \Phi\ \text{MOUVT})\,(\mathbf{B}\ e_0)\ \text{ex in} \qquad (*)$$

En utilisant les règles de réduction des combinateurs, on se ramène à la *forme normale (sans combinateurs) :*

$$\text{MOUVT}\ (\,e_0\,(\,\text{ex}\ T^2\,)\ T^1\,)\ (\,e_0\,(\,\text{in}\ T^2\,)\ T^1)$$

Cette forme normale sera l'*archétype cognitif* qui analyse la *signification* de l'expression applicative " entre-dans " $T^2\,T^1$ que nous pouvons représenter de la manière suivante :

| **état initial** | **MOUVT** | **état final** |
|---|---|---|
| La feuille est à l'extérieur de la pièce | $\longrightarrow$ | La feuille est à l'intérieur de la pièce |

Autrement dit, il y a *mouvement* d'un objet représenté par $T^1$ dont :

- l'origine est la situation statique : $e_0\,(\,\text{ex}\ T^2\,)\ T^1$,
- le but est la situation statique : $e_0\,(\,\text{in}\ T^2\,)\ T^1$.

Substituons maintenant à $T^2$ un lieu indéterminé loc, nous représentons alors l'archétype cognitif de " entre-dans " sous la forme :

| **SIT1 [ y ]** | **MOUVT** | **SIT2 [ y ]** |
|---|---|---|
| $e_0$ ex (loc) y | $\longrightarrow$ | $e_0$ in (loc) y |

On peut calculer explicitement le combinateur **X** qui représente un programme d'intégration des primitives sémantiques :

$$\text{" entre-dans "} \cong \mathbf{X}\ [\ e_0,\ \text{MOUVT, ex, in}\ ]$$

[1] Le relateur de localisation est l'une des valeurs spécifiques de l'archirelateur de repérage qui admet d'autres valeurs : attribution, ingrédience, etc... (J.-P. Desclés [7]). Pour une axiomatisation de la théorie du repérage, l'on peut consulter (J.-P. Desclés [6] ; J.-P. Desclés & C. Froidevaux [10]).

THÉORÈME ✕ $\cong$ $\mathbf{B}_{(4)}\, \mathbf{C}_{(3)}\, \psi_{(3)}\, \mathbf{B}\, \Phi\, \Phi\, \mathbf{B}$.

Nous utilisons la présentation des réductions comme dans la "déduction naturelle" de Gentzen présentée dans (J.-P. Desclés [8], p 83 ; J.-B. Grize [14]) :

| | | | |
|---|---|---|---|
| 1 | $\mathbf{B}_{(4)}\, \mathbf{C}_{(3)}\, \psi_{(3)}\, \mathbf{B}\, \Phi\, \Phi\, \mathbf{B}\, e_0$ MOUVT ex in | | hyp |
| 2 | $\mathbf{B}^4\, \mathbf{B}\, \mathbf{C}_{(3)}\, \psi_{(3)}\, \mathbf{B}\, \Phi\, \Phi\, \mathbf{B}\, e_0$ MOUVT ex in | | 1, $\mathbf{B}_{(4)}$ |
| 3 | $\mathbf{B}\, (\, \mathbf{C}_{(3)}\, \psi_{(3)}\, \mathbf{B}\, \Phi\, \Phi\, )\, \mathbf{B}\, e_0$ MOUVT ex in | | 2, $\mathbf{B}^4$ |
| 4 | $\mathbf{C}_{(3)}\, \psi_{(3)}\, \mathbf{B}\, \Phi\, \Phi\, (\, \mathbf{B}\, e_0\, )$ MOUVT ex in | | 3, $\mathbf{B}$ |
| 5 | $\mathbf{B}^3\, \mathbf{C}\, \psi_{(3)}\, \mathbf{B}\, \Phi\, \Phi\, (\, \mathbf{B}\, e_0\, )$ MOUVT ex in | | 4, $\mathbf{C}_{(3)}$ |
| 6 | $\mathbf{C}\, (\psi_{(3)}\, \mathbf{B}\, \Phi\, \Phi\, )\, (\, \mathbf{B}\, e_0\, )$ MOUVT ex in | | 5, $\mathbf{B}^3$ |
| 7 | $\psi_{(3)}\, \mathbf{B}\, \Phi\, \Phi$ MOUVT $(\, \mathbf{B}\, e_0\, )$ ex in | | 6, $\mathbf{C}$ |
| 8 | $\mathbf{B}^3\, \psi\, \mathbf{B}\, \Phi\, \Phi$ MOUVT $(\, \mathbf{B}\, e_0\, )$ ex in | | 7, $\psi_{(3)}$ |
| 9 | $\psi\, (\, \mathbf{B}\, \Phi\, \Phi$ MOUVT $)\, (\, \mathbf{B}\, e_0\, )$ ex in | loi (*) | 8, $\mathbf{B}^3$ |

Grâce au formalisme applicatif et à la logique combinatoire, nous pouvons ainsi décrire le mécanisme d'intégration des primitives sémantiques pour réaliser l'archétype cognitif qui analyse la signification de l'expression applicative.

## 2.3. Multi-représentation des connaissances

L'extension du modèle initial de la GAU au niveau *cognitif* constitue le modèle de la *Grammaire applicative et Cognitive* (GA&C, J.-P. Desclés [9]) : les représentations linguistiques, génotypiques, les représentations cognitives et le processus de *compilation* du langage externe dans les représentations cognitives s'expriment dans un même formalisme. Dans les paragraphes précédents, nous avons rappelé deux différents types de réduction effectués grâce aux $\beta$-réductions des combinateurs :

- la réduction d'une expression applicative à une expression applicative normale utilisant des lois *linguistiques* ( interprétation d'un prédicat transitif actif, réduction grammaticale passif-actif etc ...),
- la réduction d'une expression applicative à un archétype cognitif utilisant des *lois d'intégration lexicales* (interprétation de la signification du prédicat à l'aide de *primitives sémantiques*).

La compilation d'une phrase peut alors être représentée par le schéma suivant :

**phrase** **langage phénotype**

↓

**représentation applicative**

↓ ***1ère réduction (lois linguistiques grammaticales)*** **langage génotype**

**représentation applicative normale**

↓ ***2nde réduction (lois d'intégration lexicale)***

**représentation cognitive**
**archétype cognitif**

De même que deux expressions applicatives sont équivalentes si elles se réduisent à une même forme normale, deux représentations cognitives sont équivalentes si elles se réduisent à une même forme normale. Une famille paraphrastique est une classe d'équivalence pour cette relation d'équivalence. Cette relation d'équivalence sur les représentations cognitives nous permet de définir ainsi une *multi-représentation* des connaissances : nous pourrons choisir un représentant quelconque (une paraphrase) de cette classe d'équivalence.

Ce point est fondamental, car il permet de prendre en compte dans la représentation des connaissances les processus cognitifs élémentaires tels que l'*interprétation* du langage naturel et l'incidence des caractéristiques linguistiques dans cette activité, la *paraphrase*, l'*association*, et de pouvoir traduire *explicitement* les transformations à l'aide de primitives sémantiques et combinateurs. Le choix de la représentation des connaissances présentée dans ce paragraphe est donc une conséquence de l'analyse cognitive et didactique effectuées dans (D. Guin [15]). Il nous faut maintenant appliquer cette théorie au domaine particulier des problèmes addditifs.

## 3. LE FORMALISME APPLICATIF DU DOMAINE ADDITIF

### 3.1. Type sémantique des opérateurs

Nous allons maintenant adapter le formalisme applicatif basé sur la notion de Σ-algèbre au domaine qui nous intéresse. Puisque nous voulons modéliser le comportement de l'élève au cours de la lecture d'un énoncé additif, nous définirons les sortes de base dans une perspective essentiellement *cognitive* : nous essaierons donc de privilégier l'aspect *sémantique* des énoncés. Les types qui seront définis à partir de ces types élémentaires seront donc des types *sémantiques*.

#### 3.1.1. Le S-schéma d'opérateurs Σ

Voici les quatre sortes de base de notre système applicatif typé :

$\mathfrak{h}$ : propositions

$\mathfrak{j}$ : classes d'individus

$\mathbf{c}$ : collections

$\mathfrak{l}$ : lieux

L'ensemble des sortes S est donc $\{\mathfrak{h}, \mathfrak{j}, \mathfrak{l}, \mathbf{c}\}$. Nous pouvons alors définir un S-schéma d'opérateurs Σ. Donnons quelques exemples d'opérateurs de Σ :

$\sigma_1 = <$ *gagne*, $\mathfrak{F}\ (\ \mathfrak{j}\ \mathbf{c}\ )\ \mathfrak{h} >$, opérateur *gagne* à deux places,

$\sigma_2 = <$ *donne-à*, $\mathfrak{F}\ (\ \mathfrak{j}\ \mathfrak{j}\ \mathbf{c}\ )\ \mathfrak{h} >$, opérateur *donne-à* à trois places,

$\sigma_3 = <$ *est-rouge*, $\mathfrak{F}\ \mathbf{c}\ \mathfrak{h} >$, opérateur *est-rouge* à une place,

$\sigma_4 = <$ *a-dans*, $\mathfrak{F}\ (\ \mathfrak{j}\ \mathfrak{l}\ \mathbf{c}\ )\ \mathfrak{h} >$, opérateur *a-dans* à trois places,

$\sigma_5 = <$ *et*, $\mathfrak{F}\ (\ \mathbf{x}\ \mathbf{x}\ )\ \mathbf{x} >$, opérateur *et* à deux places, où $\mathbf{x} \in$ S.

#### 3.1.2. La Σ-algèbre interprétative

Pour définir la Σ-algèbre interprétative, nous nous donnons une famille $E^{(s)}$ d'ensembles:

$E^{\mathbf{c}}$ = ensemble de collections finies muni de la relation d'inclusion $\subset$, de la réunion disjointe $\cup$, de la différence $\Delta$ c c' si c'$\subset$ c.

$E^{\mathfrak{l}}$ = ensemble de lieux muni de la relation d'inclusion $\subset$, de la réunion disjointe $\cup$, de la différence $\Delta$ l l' si l'$\subset$ l.

$E^{\mathfrak{j}}$ = ensemble des classes finies d'individus muni de la relation d'inclusion $\subset$, de la réunion disjointe $\cup$, de la différence $\Delta$ j j' si j'$\subset$ j.

$E^{h}$ = ensemble des propositions des énoncés additifs.

*Exemple : 1 -* L'expression *Paul gagne 3 billes* est un élément de $E^{h}$, est donc de type **h.**
Les expressions *Paul , Paul et Jean* sont des éléments de $E^{j}$
Les expressions *billes, billes rouges* sont des éléments de $E^{c}$
Les expressions *la classe, la cour , la cour d'école* sont des éléments de $E^{l}$

*2 -* Considérons l'opérateur *gagne*, la fonction interprétative u lui associe une opération $E^{j} \times E^{c} \rightarrow E^{h}$ telle que :

[*gagne* , < Paul, 3 billes >] ≥ *Paul gagne 3 billes*

*3* - Considérons l'opérateur *et*, la fonction interprétative u lui associe l'opération réunion disjointe :
$E^{j} \times E^{j} \dashrightarrow E^{j}$ telle que [*et* , < Pierre, Paul >] ≥ *Pierre et Paul*
$E^{c} \times E^{c} \dashrightarrow E^{j}$ telle que :
[*et* , < *des billes rouges, des billes vertes* >] ≥ des *billes rouges et des billes vertes*

3.1.3. Fonctions de détermination
Soit f un concept ou propriété, c'est-à-dire une fonction à valeur dans {le-vrai, le-faux} (Frege [12]). A un concept f, nous associons une fonction δ ( f ), appelée ***fonction de détermination*** associée au concept f (J.-P. Desclés [9]). Ces fonctions agissent de manière interne sur les éléments d'une sorte, elles permettent de les déterminer plus précisément.

*Exemples : 1 -* Soit le concept "*est-en-quantité-trois* ". A ce concept est associée la fonction de détermination " *en-quantité-trois* " :

δ " en-quantité-trois " ( billes ) = " trois billes "

*2 -* Soit le concept "*est-rouge* ". A ce concept est associée la fonction de détermination " *rouge* " :

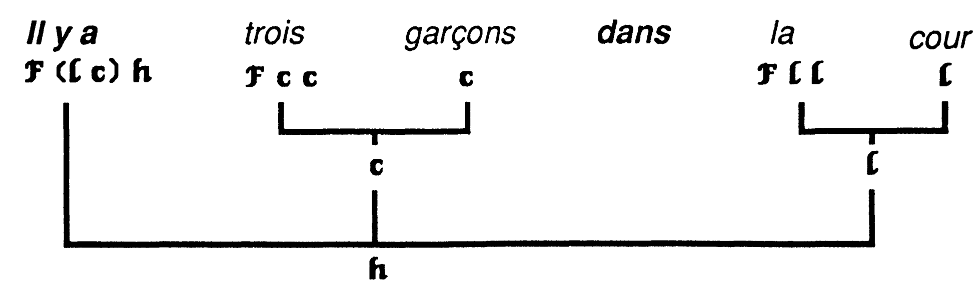
δ " rouge " ( billes ) = " billes rouges "

*3 -* Soit le concept "*est-à-Jean"*. A ce concept est associée la fonction de détermination "*à - Jean* " :

δ " à Jean " (poche) = " la poche de Jean "

3.1.4. Exemples de types sémantiques
Nous pouvons maintenant donner quelques exemples de types sémantiques d'expressions utilisées dans les énoncés additifs :

*1 - Il y a* 3 garçons dans la cour



En effet, selon la règle de simplification des types énoncée en (2.1.3) :

$$\mathbf{F}\,(\mathbf{l}\,\mathbf{c})\,\mathbf{h}\ *\mathbf{c}*\mathbf{l} \approx \mathbf{F}\,\mathbf{l}\,\mathbf{h}\ *\mathbf{l} \approx \mathbf{h}$$

Ou si l'on utilise une notation multiplicative :

$$((\mathbf{h}/\mathbf{l})/\mathbf{c})\ *\mathbf{c}*\mathbf{l} \approx (\mathbf{h}/\mathbf{l})\ *\mathbf{l} \approx \mathbf{h}$$

Le type sémantique de l'opérateur *il-y-a-dans* est donc : $\mathbf{F}\,(\mathbf{l}\,\mathbf{c})\,\mathbf{h}$ ou $((\mathbf{h}/\mathbf{l})/\mathbf{c})$.

*Remarque :* le type sémantique n'est *pas modifié* par les déterminations.

2 - Paul *gagne* 3 billes

| *Paul* | | *gagne* | *3 billes* | | *Paul* | | *gagne 3 billes* | |
|---|---|---|---|---|---|---|---|---|
| $\mathbf{j}$ | $*$ | $((\mathbf{h}/_d\,\mathbf{j})/\mathbf{c})$ | $*\mathbf{c}$ | $\approx$ | $\mathbf{j}$ | $*$ | $\mathbf{h}/_d\,\mathbf{j}$ | $\approx \mathbf{h}$ |

Le type sémantique de l'opérateur *gagne* est donc : $\mathbf{F}\,(\mathbf{j}\,\mathbf{c})\,\mathbf{h}$ ou $((\mathbf{h}/_d\,\mathbf{j})/\mathbf{c})$.

3 - Paul *doit* 3 billes *à* Pierre

| *Paul* | | *doit à* | | *3 billes* | | *Pierre* |
|---|---|---|---|---|---|---|
| $\mathbf{j}$ | $*$ | $(((\mathbf{h}/_d\mathbf{j})/\mathbf{j})/\mathbf{c})$ | $*$ | $\mathbf{c}$ | $*$ | $\mathbf{j}$ |

| | *Paul* | | *doit 3 billes à* | | *Pierre* | |
|---|---|---|---|---|---|---|
| $\approx$ | $\mathbf{j}$ | $*$ | $((\mathbf{h}/_d\,\mathbf{j})/\mathbf{j})$ | $*$ | $\mathbf{j}$ | $\approx \mathbf{h}$ |

Le type sémantique de l'opérateur *doit à* est donc : $\mathbf{F}\,(\mathbf{j}\,\mathbf{j}\,\mathbf{c})\,\mathbf{h}$ ou $(((\mathbf{h}/_d\,\mathbf{j})/\mathbf{c})/\mathbf{j})$.

4 - Paul a gagné 3 billes *en tout*

| *Paul a gagné 3 billes* | *en tout* | |
|---|---|---|
| $\mathbf{h}$ | $*\,(\mathbf{h}/_d\mathbf{h})$ | $\approx \mathbf{h}$ |

Le type sémantique de l'opérateur *en tout* est donc : $\mathbf{F}\,\mathbf{h}\,\mathbf{h}$ ou $(\mathbf{h}/_d\mathbf{h})$.

5 - Dans l'expression *Pierre et Jean ont (ensemble) cinq billes*, le type sémantique de l'opérateur *et* est $\mathbf{F}(\mathbf{j}\,\mathbf{j})\,\mathbf{j}$ ou $((\mathbf{j}/_d\mathbf{j})/\mathbf{j})$.

- Dans l'expression *Pierre a 4 boules rouges et 4 boules vertes*, le type sémantique de l'opérateur *et* est $\mathbf{F}\,\mathbf{c}\,\mathbf{c}$ ou $((\mathbf{c}/_d\,\mathbf{c})/\mathbf{c})$.
- Dans l'expression *Il y a trois chaises dans la cour et deux dans la classe*, le type sémantique de l'opérateur *et* est $\mathbf{F}\,\mathbf{h}\,\mathbf{h}$ ou $((\mathbf{h}/_d\mathbf{h})/\mathbf{h})$.

Nous assignerons donc à l'opérateur *et*, le *type variable* $\mathbf{F}\,(\mathbf{x}\,\mathbf{x})\,\mathbf{x}$ ou $((\mathbf{x}/_d\mathbf{x})/\mathbf{x})$.

6 - Pierre *a* des billes *dans sa* main

| *Pierre* | | *a dans sa* | | *des billes* | | *main* |
|---|---|---|---|---|---|---|
| $\mathbf{j}$ | $*$ | $(((\mathbf{h}/_d\mathbf{j})/\mathbf{l})/\mathbf{c})$ | $*$ | $\mathbf{c}$ | $*$ | $\mathbf{l}$ |

| | *Pierre* | | *a des billes dans sa* | | *main* |
|---|---|---|---|---|---|
| $\approx$ | $\mathbf{j}$ | $*$ | $((\mathbf{h}/_d\mathbf{j})/\mathbf{l})$ | $*$ | $\mathbf{l}$ |

| | *Pierre* | | *a des billes dans sa main* | |
|---|---|---|---|---|
| $\approx$ | $\mathbf{j}$ | $*$ | $(\mathbf{h}/_d\mathbf{j})$ | $\approx \mathbf{h}$ |

Le type sémantique de *a dans sa* est donc $\mathbf{F}\ (\mathbf{j}\,\mathbf{\ell}\,\mathbf{c})\,\mathbf{h}$ ou $(((\mathbf{h}/_d\mathbf{j})/\mathbf{\ell})/\mathbf{c})$.

*7* - Paul a *trois* billes

| *Paul* | *a* | *trois* | *billes* | | *Paul* | *a* | *trois billes* |
|---|---|---|---|---|---|---|---|
| $\mathbf{j}$ * | $((\mathbf{h}/_d\mathbf{j})/\mathbf{c})$ * | $(\mathbf{c}/\mathbf{c})$ | * $\mathbf{c}$ | $\approx$ | $\mathbf{j}$ * | $((\mathbf{h}/_d\mathbf{j})/\mathbf{c})$ | * $\mathbf{c}$ |

| *Paul* | *a trois billes* | | *Paul a trois billes* |
|---|---|---|---|
| $\mathbf{j}$ * | $(\mathbf{h}/_d\mathbf{j})$ | $\approx$ | $\mathbf{h}$ |

Le type sémantique de l'opérateur *trois* est donc $\mathbf{F}\ \mathbf{c}\ \mathbf{c}$ ou $(\mathbf{c}/\mathbf{c})$. Plus généralement, considérons un concept f du S-schéma d'opérateurs $\Sigma$, de type $\mathbf{F}\ \mathbf{x}\ \mathbf{h}$, où $\mathbf{x} \in \{\mathbf{c}, \mathbf{\ell}, \mathbf{j}\}$, l'opérateur de détermination $\delta$ fait correspondre à f un opérateur de type $\mathbf{F}\ \mathbf{x}\ \mathbf{x}$.

### 3.2. Les primitives sémantiques du domaine additif

#### 3.2.1. L'archirelateur a et les primitives sémantiques associées

Nous utiliserons dans ce paragraphe les résultats de (J.-P. Desclés [7]). Si nous considérons le domaine des problèmes additifs, l'archirelateur *a* restreint au domaine comprend *deux valeurs sémantiques* distinctes :

##### 3.2.1.1. La possession $a_{poss}$

Il est possible de distinguer deux spécifications :

- la possession *aliénable :* Jean *a* une maison dans le midi
- la possession *inaliénable :* Jean *a* une main blessée (Cette dernière spécification est très liée à la notion d'*ingrédience*).

Le relateur $a_{poss}$ est le converse du relateur *est-à :*

*ex :* Jean *a* une maison ----> une maison *est à* Jean

Nous désignerons par *loi sémantique* la loi suivante qui exprime une relation sémantique :

$$a_{poss} = \mathrm{C}\ \textit{est-à}$$

##### 3.2.1.2. La localisation $a_{loc}$

- Dans un *lieu* : par exemple, *il y a* 3 garçons *dans* la cour
- *Chez* un individu : par exemple, Jean *a* le chapeau de Paul sur la tête

Dans ce cas, nous pourons dire que le relateur *a* est le converse du relateur *est-chez* que nous noterons $e_{loc}$ : Jean *a* le chapeau de Paul ----> le chapeau de Paul *est chez* Jean

On a la loi sémantique suivante :

$$a_{loc} = \mathrm{C}\ e_{loc}$$

##### 3.2.1.3. Une spécialisation commune à la possession et la localisation :

Dans la phrase "Jean *a* son chapeau vert", nous noterons le relateur $a_{loc/poss}$. On a la loi sémantique suivante :

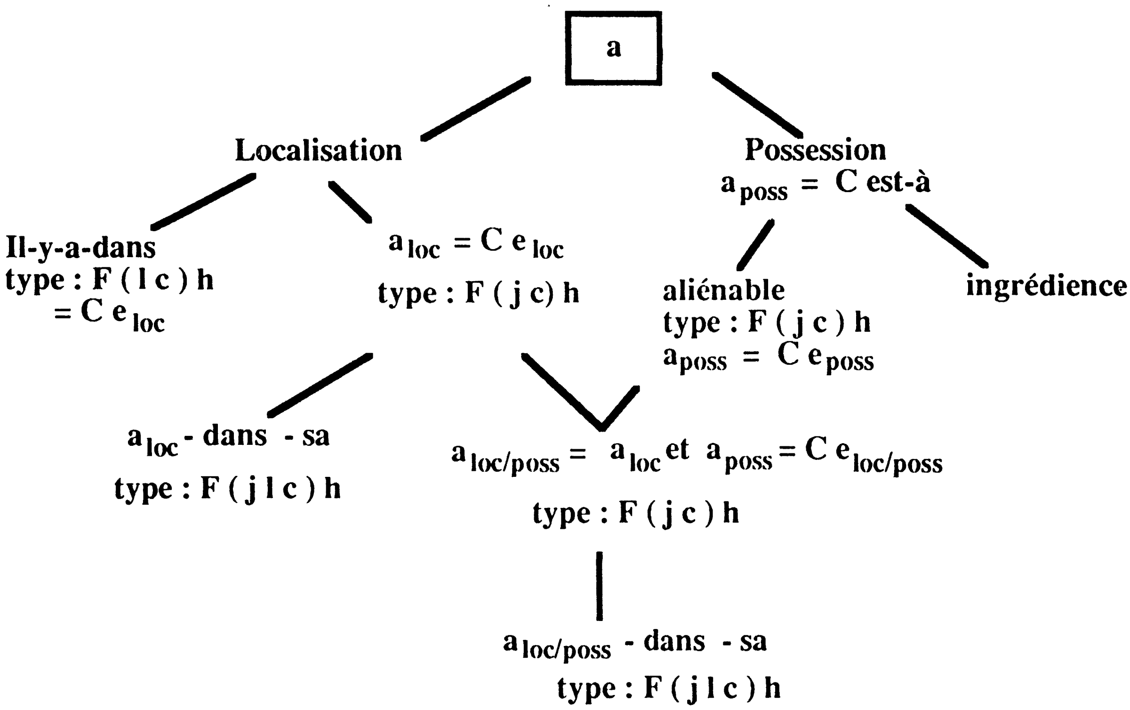
$$a_{loc/poss} = a_{loc}\ \textit{et}\ a_{poss} = \mathrm{C}\ e_{loc/poss}$$

*Remarques :* - Une distinction analogue peut être effectuée pour l'opérateur *a-dans-sa* que nous noterons :

$a$ $_{loc}$ - *dans-sa* dans "Jean $a$ 3 billes *dans* sa poche"
$a$ $_{loc/poss}$ - *dans-sa* dans "Jean $a$ son mouchoir *dans* sa poche"

- On ne peut toujours déterminer la valeur sémantique du relateur $a$ (par exemple, Jean $a$ 5 billes, il en donne 4 à Pierre. Combien lui en reste-t-il ?).

Nous pouvons résumer ces différentes valeurs sémantiques dans le tableau suivant :




Les différents archétypes correspondant à la signification des verbes font intervenir ces différentes valeurs sémantiques du relateur $a$ :

$a$ $_{loc}$ : *perd, trouve, lui-reste, rend-à, donne-à, doit-à*
$a$ $_{poss}$ : *perd (jeu), lui-reste, doit-à, gagne*
$a$ $_{loc/poss}$ : *donne-à*
$a$ $_{loc}$ - *dans* ou $a$ $_{loc/poss}$ - *dans* : *met -dans*

3.2.2. Primitives sémantiques associées au opérateurs additifs
Nous avons défini en (5.2) de (D. Guin [15]) des opérateurs sur les entiers naturels qui interviennent dans notre typologie des problèmes et justifié, pour des raisons cognitives, la distinction entre opérateur unaire et binaire. Nous pouvons les exprimer en langue naturelle :

ADD = " *Faire la somme de*", opérateur binaire
SOUS = "*Faire la différence de*", opérateur binaire
AJOUT n = "*Ajouter n à*", opérateur unaire
RET n = "*Retrancher n à*", opérateur unaire

Le *principe applicatif* de Schönfinkel (J.-P. Desclés [8]) permet d'exprimer les opérateurs unaires AJOUT n et RET n à l'aide des primitives sémantiques ADD et SOUS grâce à l'opération de Curriage *Cur* (f) qui associe à un opérateur n-aire f, un opérateur unaire :

$$Cur : \mathrm{Hom}(\mathbb{N} \times \mathbb{N}, \mathbb{N}) \approx \mathrm{Hom}(\mathbb{N}, \mathrm{Hom}(\mathbb{N}, \mathbb{N}))$$

$$\text{ADD} \longmapsto Cur\,(\text{ADD}) : \mathbb{N} \rightarrow \text{Hom}\,(\mathbb{N}, \mathbb{N})$$
$$n \rightarrow \text{AJOUT } n$$

*Remarque* . Les relations ADD n n' = AJOUT n ( n' ) et SOUS n n' = RET n' ( n ) ne sont qu'une traduction de la propriété caractéristique de l'opération de Curriage.

3.2.3. Primitives sémantiques connecteurs
Nous utiliserons plusieurs primitives sémantiques *connecteurs d'états* :

*ET : état* x *état -----> état*
*Ex :* il y a 3 chaises dans la cour et 2 dans la classe.

Le connecteur *MODIF* sera utilisé pour décrire un *processus :*

*MODIF : état* x *état -----> processus*
*Ex :* Lundi Pierre avait 5 francs, aujourd'hui il lui en reste 4.

Le connecteur *MOUVT*, spécialisation de *MODIF*, sera utilisé dans le cas d'un mouvement :

*Ex :* Pierre *donne à* Paul 5 francs.

La primitive sémantique *SUC* exprimera la succession de deux processus :

*SUC : processus* x *processus -----> processus*
*Ex :* A la première partie Pierre gagne 3 billes, à la seconde partie il en perd 2.

*Remarque.* Ces primitives sémantiques permettent de rendre compte du processus cognitif d'association.

3.2.4. Primitives sémantiques temporelles statiques
Les primitives sémantiques *INIT* et *FIN* permettent d'exprimer qu'un état est l'état initial ou final d'un processus. Nous avons donc les relations :

état1 = *INIT MODIF* état1 état2
état2 = *FIN MODIF* état1 état2

*Ex :* Pierre a gagné 4 billes, avant il en avait 5.
Pierre a gagné 4 billes, il en a maintenant 5.

3.2.5. Primitives sémantiques temporelles dynamiques
Les primitives sémantiques *PREM* et *DER* permettent d'exprimer qu'une modification est initiale ou finale dans une succession de processus. Nous avons donc les relations :

Modif1 = PREM SUC Modif1 Modif2
Modif2 = DER SUC Modif1 Modif2

Ex : Pierre à la première partie a gagné 5 billes.
Pierre à la deuxième partie a gagné 5 billes.

*Remarques.* 1 - Les expressions exprimant une marque temporelle telle que *à la première partie, avant, hier, etc...* pouront être représentées à partir de ces primitives sémantiques.

2 - Nous utiliserons également les primitives sémantiques de contrôle : *FAIRE, CONTR, TRANS* définies p. 298 dans (J.-P. Desclés [9]).

3.2.6. Primitives sémantiques ensemblistes

Dans un grand nombre de recherches cognitives sur les problèmes additifs (D. Guin [15], § (3.1.1) p 7) l'on estime que, dans les énoncés arithmétiques, seuls les éléments relatifs aux ensembles mathématiques, aux cardinaux et aux relations entre eux doivent être pris en compte dans l'activité de compréhension. Afin de ne pas négliger l'importance de ces notions au détriment de l'interprétation du langage naturel, nous définissons des primitives sémantiques sur la sorte des collections finies **c** correspondant aux opérations cognitives sur les ensembles décrites par des schémas d'action dans les articles précités, en conservant la distinction entre opérateur unaire et binaire :

Nous utiliserons deux primitives sémantiques *ensemblistes :*

$\cup$ = " *Faire la réunion de"*, opérateur binaire

$\Delta$ = " *Faire la différence de"*, opérateur binaire

*Remarque*. Comme précédemment, grâce à l'opération de Curriage, nous en déduirons les opérateurs unaires :

INS c = " *Insérer la collection c dans"*
ENL c = " *Enlever la collection c à"*

Afin de représenter la signification des énoncés additifs, nous pouvons dorénavant construire des représentations cognitives à l'aide de ces primitives sémantiques et des combinateurs.

## 4. LES ARCHÉTYPES COGNITIFS ET LOIS SÉMANTIQUES DU DOMAINE ADDITIF

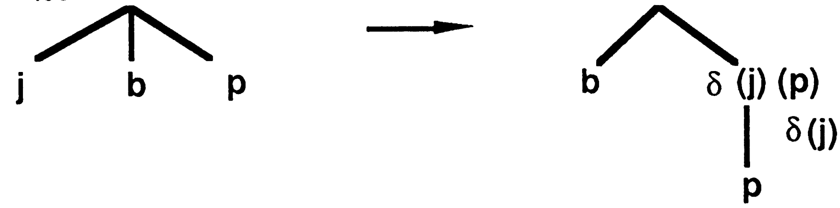
Remarquons que la validité des archétypes que nous définissons est limitée au domaine restreint que nous nous sommes fixé. Ces archétypes cognitifs permettent de donner une représentation *sémantique* des items lexicaux, notamment verbaux. Nous présentons ici quelques archétypes du domaine :

### 4.1. Archétypes statiques

4.1.1. L'archétype cognitif de $A_{loc}$-DANS-SA peut s'exprimer à l'aide de la primitive sémantique de localisation $e_{loc}$ et de l'opérateur de détermination $\delta$ :

$A_{loc}$-DANS-SA p b j ----> $e_{loc}$ ( $\delta$ (j) (p) ) b
*"Jean a des bonbons dans sa poche "* *"des bonbons sont dans la poche de Jean "*
*où* *p : = sa poche, b : = des bonbons, j : = Jean.*

Nous pouvons représenter cet archétype cognitif sous forme d'arbre :

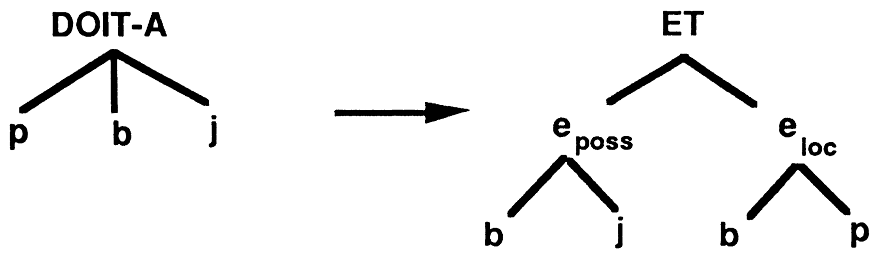



Archétype de $A_{loc}$-DANS-SA p b j : $e_{loc}$ ( $\delta$ (j) (p) ) b

4.1.2. L'archétype cognitif de DOIT-A peut s'exprimer à l'aide des primitives sémantiques $e_{poss}$ et $e_{loc}$ :

DOIT-A j b p ----> ET ( $e_{poss}$ j b ) ( $e_{loc}$ p b )
*"Paul doit des billes à Jean "* *"des billes sont à Jean et sont chez Paul "*
*(C'est-à-dire qu'une collection de billes qui appartient à Jean est actuellement détenue par Paul)*
*où p : = Paul , j : = Jean, b : = des billes*

Nous pouvons représenter cet archétype cognitif sous forme d'arbre :




Archétype de DOIT-A j b p : ET ( $e_{poss}$ j b ) ( $e_{loc}$ p b )

4.1.3. L'archétype cognitif de A-DE-MOINS-QUE peut s'exprimer à l'aide des primitives sémantiques $e_{poss}$ ou $e_{loc}$, de l'opérateur de détermination δ, et de la primitive ENL. Cet opérateur ne s'applique que sur des collections dont la quantité est déterminée du type δ n c où δ n est la fonction de détermination "en quantité n" (3.1.3).

A-DE-MOINS-QUE j ( δ 4 b ) p ----> ET ( $e_{poss}$ j (δ q c ))
( $e_{poss}$ p ((ENL (δ 4 b ) (δ q d ) )))

*"Paul a quatre billes de moins que Jacques "* *connection d'états :*
*une collection de q billes c est à Jacques*
*et une collection de q billes à laquelle on ôte quatre billes est à Paul.*
*où j : = Jacques, δq c : = une collection de q billes , δq d : = une collection de q billes,*
*δ4 b : = une sous-collection de 4 billes de la collection d, p : = Paul.*

Archétype de A-DE-MOINS-QUE j (δ n b) p : ET ( $e_{poss}$ j (δ q c ))
( $e_{poss}$ p ((ENL (δ n b ) (δ q d ) )))

4.1.4. L'archétype cognitif de RESTE-A peut s'exprimer à l'aide des primitives sémantiques FIN, $e_{poss}$ (ou $e_{loc}$) et MODIF :

RESTE-A j c -----> FIN ( MODIF ( $e_{poss}$ j d ) ( $e_{poss}$ j c ) )
*"Il reste à Jean des billes "* *"Jean a une collection de billes c est l'état final d'une modification d'états où l'état initial est Jean a une collection d contenant ces billes"*

*où j : = Jean, c : = des billes.*

Archétype de RESTE-A j c : FIN ( MODIF ( $e_{poss}$ j d ) ( $e_{poss}$ j c ) ) où $c \subset d$

4.2. Archétypes dynamiques

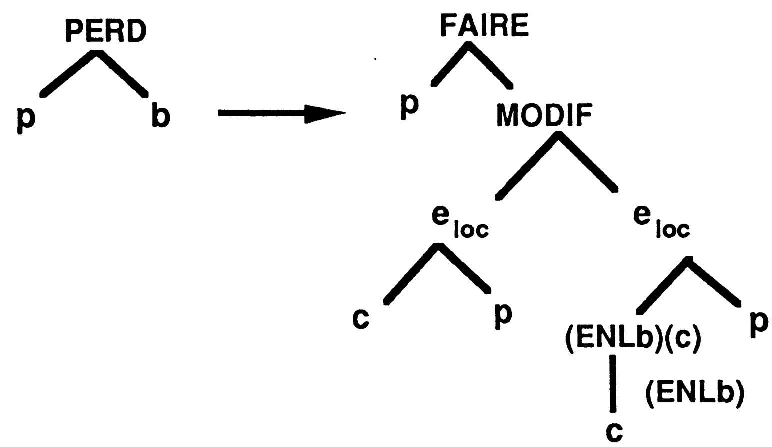
4.2.1. L'archétype cognitif de PERD peut s'exprimer à l'aide de primitives sémantiques $e_{loc}$ ou $e_{poss}$, FAIRE [1], MODIF, ENL b :

PERD b p ---> FAIRE ( MODIF ( $e_{loc}$ p c ) ( $e_{loc}$ p ((ENL b) (c)) )) j

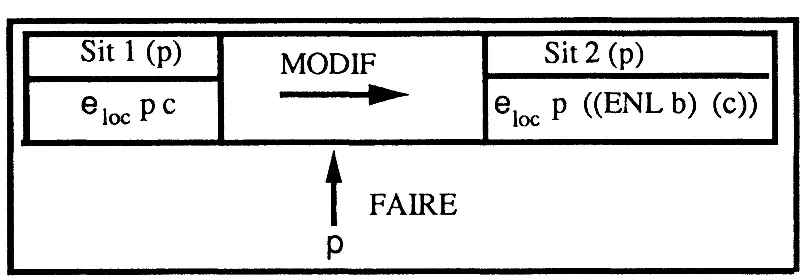
*" Paul perd des billes "* *changement d'état sans contrôle :*
- *état initial : Paul a une collection de billes c,*
- *état final : Paul a une collection c à laquelle on a ôté la sous-collection b.*

*où b : = des billes, p : = Paul.*

Nous pouvons représenter cet archétype cognitif sous forme d'arbre :




Nous pouvons représenter également cet archétype cognitif par le schéma suivant :




Archétype de PERD b p : FAIRE ( MODIF ( $e_{loc}$ p c ) ( $e_{loc}$ p ((ENL b) (c)) )) p

4.2.2. L'archétype cognitif de REND-A peut s'exprimer à l'aide des primitives sémantiques TRANS[1], MODIF, ENL b et de l'opérateur DOIT-A :

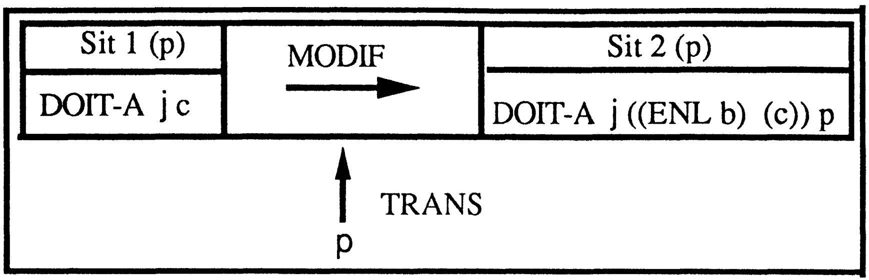
REND-A j b p ---> TRANS ( MODIF ( DOIT-A j c p )
( DOIT-A j ((ENL b) (c)) p )) j

[1] la primitive FAIRE exprime qu'un individu *effectue* une modification d'états.

[1] TRANS est l'opérateur de *transitivité sémantique* obtenu en composant les opérateurs CONTR et FAIRE, il est lié à la notion de contrôle intentionnel et d'effectuation.

*Paul rend des billes à Jean* *changement d'état avec contrôle :*
*- état initial : Paul doit une collection c de billes àJean contenant b.*
*- état final : Paul doit à Jean la collection c à laquelle on a ôté b.*
*où b: = des billes , p : = Paul , j: = Jean*

Nous pouvons représenter également cet archétype cognitif par le schéma suivant :




Archétype de REND-A j b p : TRANS ( MODIF ( DOIT-A j c p ) ( DOIT-A j ((ENL b) (c)) p )) j

Nous allons donner quelques exemples de lois d'intégration lexicales (2.2), qui décrivent le programme d'intégration des primitives sémantiques :

4.3.1. Une loi d'*intégration lexicale* pour l'opérateur $A_{loc}$-DANS-SA est donnée par :

$$A_{loc}\text{-DANS-SA } p\, b\, j \cong \boldsymbol{\times}\ [\, e_{loc}\, , \delta\, ]\ p\, b\, j$$

où $\boldsymbol{\times}$ désigne un combinateur qui représente un programme d'intégration des primitives sémantiques $e_{loc}$ et $\delta$, permettant de passer d'un opérateur ternaire à un opérateur binaire :

THÉORÈME : $\boldsymbol{\times} = \mathbf{B}\mathbf{B}^2\, \mathbf{C}_{(2)}\, \mathbf{C}\, \mathbf{B}^2$.

En effet, nous avons les réductions suivantes :

| | |
|---|---|
| 1 $\boldsymbol{\times}\ e_{loc}\ \delta\ p\ b\ j$ | 6 $\mathbf{B}^2\ \mathbf{C}\ \mathbf{C}\ f\ p\ b\ j$ |
| 2 $\mathbf{B}\ \mathbf{B}^2\ \mathbf{C}_{(2)}\ \mathbf{C}\ \mathbf{B}^2\ e_{loc}\ \delta\ p\ b\ j$ | 7 $\mathbf{C}\ (\ \mathbf{C}\ f\ p\ )\ b\ j$ |
| 3 $\mathbf{B}^2\ (\ \mathbf{C}_{(2)}\ \mathbf{C}\ )\ \mathbf{B}^2\ e_{loc}\ \delta\ p\ b\ j$ | 8 $\mathbf{C}\ f\ p\ j\ b$ |
| 4 $\mathbf{C}_{(2)}\ \mathbf{C}\ (\ \mathbf{B}^2\ e_{loc}\ \delta\ )\ p\ b\ j$ | 9 $f\ j\ p\ b$ |
| 5 $\mathbf{B}^2\ \mathbf{C}\ \mathbf{C}\ (\ \mathbf{B}^2\ e_{loc}\ \delta\ )\ p\ b\ j$ | 10 $\mathbf{B}^2\ e_{loc}\ \delta\ j\ p\ b$ |

11 $e_{loc}\ (\ \delta\ j\ p\ )\ b$ {archétype de $A_{loc}$-DANS-SA }

4.3.2. Une loi d'*intégration lexicale* pour l'opérateur DOIT-A est donnée par :

$$\text{DOIT-A} \cong \boldsymbol{\times}\ [\ \text{ET}, e_{poss}, e_{loc}\ ]$$

THÉORÈME : $\boldsymbol{\times} = \mathbf{B}\ \mathbf{B}\ \mathbf{C}_{(4)}\ \mathbf{H}\ \phi$ [1].

---

[1] où $\mathbf{H}$ est le combinateur tel que : $\mathbf{H}\ f\ x\ y\ z\ t = f\ (\ x\ z\ )\ (\ y\ t\ )$, $\mathbf{H} = \mathbf{C}_{(4)}\ \mathbf{B}^3\ \mathbf{B}\ \mathbf{B}$, en effet :

| | |
|---|---|
| 1 $\mathbf{C}_{(4)}\ \mathbf{B}^3\ \mathbf{B}\ \mathbf{B}\ f\ x\ y\ z\ t$ | 4 $\mathbf{B}^3\ \mathbf{B}\ \mathbf{B}\ f\ x\ z\ y\ t$ |
| 2 $\mathbf{B}^4\ \mathbf{C}\ \mathbf{B}^3\ \mathbf{B}\ \mathbf{B}\ f\ x\ y\ z\ t$ | 5 $\mathbf{B}\ (\ \mathbf{B}\ f\ x\ z\ )\ y\ t$ |
| 3 $\mathbf{C}\ (\ \mathbf{B}^3\ \mathbf{B}\ \mathbf{B}\ f\ x\ )\ y\ z\ t$ | 6 $\mathbf{B}\ f\ x\ z\ (\ y\ t\ )$ |

7 $f\ (\ x\ z\ )\ (\ y\ t\ )$

En effet, nous avons les réductions suivantes :

1 $\mathbf{B}\ \mathbf{B}\ \mathbf{C}_{(4)}\ \mathbf{H}\ \phi\ \mathrm{ET}\ \ e_{poss}\ e_{loc}\ j\ b\ p$
2 $\mathbf{B}\ (\ \mathbf{C}_{(4)}\ \mathbf{H}\ )\ \ \phi\ \mathrm{ET}\ \ e_{poss}\ e_{loc}\ j\ b\ p$
3 $\mathbf{C}_{(4)}\ \mathbf{H}\ (\ \phi\ \mathrm{ET}\ )\ e_{poss}\ e_{loc}\ j\ b\ p$
4 $\mathbf{B}^4\ \mathbf{C}\ \mathbf{H}\ (\ \phi\ \mathrm{ET}\ )\ e_{poss}\ e_{loc}\ j\ b\ p$
5 $\mathbf{C}\ (\ \mathbf{H}\ (\ \phi\ \mathrm{ET}\ )\ e_{poss}\ e_{loc}\ j\ )\ b\ p$
6 $\mathbf{H}\ (\ \phi\ \mathrm{ET}\ )\ \ e_{poss}\ e_{loc}\ j\ p\ b$
7 $\mathbf{H}\ f\ x\ y\ z\ t$
8 $f\ (\ x\ z\ )\ (\ y\ t\ )$
9 $\phi\ \mathrm{ET}\ \ (\ e_{poss}\ j\ )\ \ (\ e_{loc}\ p\ )\ b$
10 $\mathrm{ET}\ (\ e_{poss}\ j\ b\ )\ (\ e_{loc}\ p\ b\ )$ { archétype de DOIT-A}

4.3.3 . Une loi d'*intégration lexicale* pour l'opérateur RESTE-A est donnée par :

$$\text{RESTE-A} \ \equiv\ \boldsymbol{\times}\ \ [\ \text{FIN, MODIF, } e_{poss}]$$

THÉORÈME: $\boldsymbol{\times} = \mathbf{B}^6\ (\ \mathbf{C}\ \mathbf{B}\ \mathbf{C}\ )\ \mathbf{B}^4\ (\mathbf{C}_{(3)}\ \psi)\ \mathbf{B}^2\ \phi\ \mathbf{B}^2.$

En effet, nous avons les réductions suivantes :

1 $\mathbf{B}^6\ (\ \mathbf{C}\ \mathbf{B}\ \mathbf{C}\ )\ \mathbf{B}^4\ (\mathbf{C}_{(3)}\ \psi)\ \mathbf{B}^2\ \phi\ \mathbf{B}^2\ \mathrm{FIN\ MODIF}\ e_{poss}\ d\ j\ c$
2 $\mathbf{C}\ \mathbf{B}\ \mathbf{C}\ (\ \mathbf{B}^4\ (\mathbf{C}_{(3)}\ \psi)\ \mathbf{B}^2\ \phi\ \mathbf{B}^2\ \mathrm{FIN\ MODIF}\ )\ e_{poss}\ d\ j\ c$
3 $\mathbf{C}\ \mathbf{B}\ \mathbf{C}\ (\ \mathbf{C}_{(3)}\ \psi\ (\ \mathbf{B}^2\ \phi\ \mathbf{B}^2\ \mathrm{FIN\ MODIF}\ ))\ e_{poss}\ d\ j\ c$
4 $\mathbf{C}\ \mathbf{B}\ \mathbf{C}\ (\ \mathbf{C}_{(3)}\ \psi\ (\ \phi\ (\ \mathbf{B}^2\ \mathrm{FIN\ MODIF}\ )))\ e_{poss}\ d\ j\ c$
5 $\mathbf{B}\ (\ \mathbf{C}_{(3)}\ \psi\ (\ \phi\ (\ \mathbf{B}^2\ \mathrm{FIN\ MODIF}\ )))\ \mathbf{C}\ e_{poss}\ d\ j\ c$
6 $\mathbf{C}_{(3)}\ \psi\ (\ \phi\ (\ \mathbf{B}^2\ \mathrm{FIN\ MODIF}\ ))\ (\ \mathbf{C}\ e_{poss}\ )\ d\ j\ c$
7 $\mathbf{B}^3\ \mathbf{C}\ \psi\ (\ \phi\ (\ \mathbf{B}^2\ \mathrm{FIN\ MODIF}\ ))\ (\ \mathbf{C}\ e_{poss}\ )\ d\ j\ c$
8 $\mathbf{C}\ (\ \psi\ (\ \phi\ (\ \mathbf{B}^2\ \mathrm{FIN\ MODIF}\ ))\ (\ \mathbf{C}\ e_{poss}\ )\ d\ )\ j\ c$
9 $\psi\ (\ \phi\ (\ \mathbf{B}^2\ \mathrm{FIN\ MODIF}\ ))\ (\ \mathbf{C}\ e_{poss}\ )\ \ d\ c\ j$
10 $\phi\ (\ \mathbf{B}^2\ \mathrm{FIN\ MODIF}\ )\ (\ \mathbf{C}\ e_{poss}\ d\ )\ (\ \mathbf{C}\ e_{poss}\ c\ )\ j$
11 $\mathbf{B}^2\ \ \mathrm{FIN\ MODIF}\ (\ \mathbf{C}\ e_{poss}\ d\ j\ )\ (\ \mathbf{C}\ e_{poss}\ c\ \ j\ )$
12 $\mathbf{B}^2\ \ \mathrm{FIN\ MODIF}\ \ (\ e_{poss}\ j\ d\ )\ (\ e_{poss}\ j\ c\ )$
13 $\mathrm{FIN}\ (\ \mathrm{MODIF}\ (\ e_{poss}\ j\ d\ )\ (\ e_{poss}\ j\ c\ )\ )$ {archétype de RESTE-A}

4.3.4. Une loi d'*intégration lexicale* pour l'opérateur PERD est donnée par :

$$\mathrm{PERD}\ \equiv\ \mathbf{W}_{(3)}\ \mathbf{B}^2\ \mathrm{FAIRE}\ (\ \phi^2\ \mathrm{MODIF}\ (\ \mathbf{B}^2\ (\ \mathbf{C}\ e_{loc}\ )\ \mathbf{K}\ c\ )$$
$$(\ \mathbf{B}^3\ (\ \mathbf{C}\ e_{loc}\ )\ \mathbf{C}\ (\ \mathbf{B}\ \mathbf{I}\ \mathrm{ENL})\ c\ )\ )$$

Considérons le prédicat $P_2$ (PERD), nous avons les réductions suivantes :

1 $P_2\ \ b\ p$
2 $[P_2 = \ \mathbf{B}^3\ \mathbf{W}\ \mathbf{B}^2\ \mathrm{FAIRE}\ \ \ P_3]$
3 $\mathbf{B}^3\ \mathbf{W}\ \mathbf{B}^2\ \mathrm{FAIRE}\ P_3\ \ b\ p$
4 $\mathbf{W}\ (\ \mathbf{B}^2\ \mathrm{FAIRE}\ P_3\ \ b\ )\ p$
5 $\mathbf{B}^2\ \mathrm{FAIRE}\ P_3\ b\ p\ p$
6 $\mathrm{FAIRE}\ (\ P_3\ \ b\ p\ )\ p$
7 $[\ P_3 = \ \mathbf{B}\ \phi\ \phi\ \ \mathrm{MODIF}\ \qquad (\ \mathbf{B}^2\ (\ \mathbf{C}\ e_{loc}\ )\ \mathbf{K}\ c\ )$
$(\ \mathbf{B}^3\ (\ \mathbf{C}\ e_{loc}\ )\ \mathbf{C}\ (\ \mathbf{B}\ \mathbf{I}\ \mathrm{ENL})\ c\ )]$
8 $\mathbf{B}\ \phi\ \phi\ \mathrm{MODIF}\ (\ \mathbf{B}^2\ (\ \mathbf{C}\ e_{loc}\ )\ \mathbf{K}\ c\ )$
$\mathbf{B}^3\ (\ \mathbf{C}\ e_{loc}\ )\ \mathbf{C}\ (\ \mathbf{B}\ \mathbf{I}\ \mathrm{ENL})\ c\ )\ b\ p$
9 $\mathbf{B}\ \phi\ \phi\ \ \mathrm{MODIF}$
10 $\phi\ (\phi\ \mathrm{MODIF})$
11 $\phi\ (\phi\ \mathrm{MODIF})\ \ (\ \mathbf{B}^2\ (\ \mathbf{C}\ e_{loc}\ )\ \mathbf{K}\ c\ )$

10 $\phi$ ($\phi$ MODIF)
11 $\phi$ ($\phi$ MODIF) ( $\mathbf{B}^2$ ( $\mathbf{C}$ e $_{loc}$ ) $\mathbf{K}$ c )
( $\mathbf{B}^3$ ( $\mathbf{C}$ e $_{loc}$ ) $\mathbf{C}$ ( $\mathbf{B}$ $\mathbf{I}$ ENL) c ) b p
12 $\phi$ MODIF ( $\mathbf{B}^2$ ( $\mathbf{C}$ e $_{loc}$ ) $\mathbf{K}$ c b )
( $\mathbf{B}^3$ ( $\mathbf{C}$ e $_{loc}$ ) $\mathbf{C}$ ( $\mathbf{B}$ $\mathbf{I}$ ENL) c b ) p
13 MODIF ( $\mathbf{B}^2$ ( $\mathbf{C}$ e $_{loc}$ ) $\mathbf{K}$ c b p )
( $\mathbf{B}^3$ ( $\mathbf{C}$ e $_{loc}$ ) $\mathbf{C}$ ( $\mathbf{B}$ $\mathbf{I}$ ENL) c b p )
14 $\mathbf{B}^2$ ( $\mathbf{C}$ e $_{loc}$ ) $\mathbf{K}$ c b p
15 $\mathbf{C}$ e $_{loc}$ ( $\mathbf{K}$ c b ) p
16 e $_{loc}$ p ( $\mathbf{K}$ c b )
17 e $_{loc}$ p c
18 $\mathbf{B}^3$ ( $\mathbf{C}$ e $_{loc}$ ) $\mathbf{C}$ ( $\mathbf{B}$ $\mathbf{I}$ ENL) c b p
19 $\mathbf{C}$ e $_{loc}$ ( $\mathbf{C}$ ( $\mathbf{B}$ $\mathbf{I}$ ENL) c b ) p
20 e $_{loc}$ p ( $\mathbf{C}$ ( $\mathbf{B}$ $\mathbf{I}$ ENL) c b )
21 e $_{loc}$ p ( $\mathbf{B}$ $\mathbf{I}$ ENL b c)
22 e $_{loc}$ p ( $\mathbf{I}$ (ENL b) c)
23 e $_{loc}$ p ((ENL b) c)
24 MODIF ( e $_{loc}$ p c ) ( e $_{loc}$ p ((ENL b) c) )
25 FAIRE ( MODIF ( e $_{loc}$ p c ) ( e $_{loc}$ p ((ENL b) c) )) p {archétype de PERD}

4.4. Lois sémantiques

Il existe des relations *sémantiques* (que l'on peut exprimer dans le même formalisme que les lois précédentes) entre primitives sémantiques, ainsi qu'entre opérateurs. Ces relations, que nous désignerons par *lois sémantiques,* interviennent explicitement dans la réduction à un prototype (D. Guin [15]), en voici quelques exemples :

4.4.1. Considérons la loi sémantique :

ET ( e $_{loc}$ j c) ( e $_{loc}$ j c' ) = e $_{loc}$ j ( $\cup$ c c')
*" j a la collection c et la collection c' "* équivaut à *" j a la réunion des collections c et c'"*

Voici son expression dans le formalisme applicatif :

$$\mathbf{B}\ \mathbf{B}\ \phi\ \text{ET e}_{loc}\ \text{j c c'} = \mathrm{C}_{(2)}\ \mathbf{B}\ \mathbf{B}^2\ \text{e}_{loc} \cup \text{j c c'}$$

En effet, nous avons les réductions suivantes :

| | |
|---|---|
| 1 $\mathbf{B}$ $\mathbf{B}$ $\phi$ ET e $_{loc}$ j c c' | 1 $\mathrm{C}_{(2)}$ $\mathbf{B}$ $\mathbf{B}^2$ e $_{loc}$ $\cup$ j c c' |
| 2 $\mathbf{B}$ ( $\phi$ ET ) e $_{loc}$ j c c' | 2 $\mathbf{B}^2$ $\mathbf{C}$ $\mathbf{B}$ $\mathbf{B}^2$ e $_{loc}$ $\cup$ j c c' |
| 3 $\phi$ ET ( e $_{loc}$ j ) c c' | 3 $\mathbf{C}$ ( $\mathbf{B}$ $\mathbf{B}^2$ e $_{loc}$ ) $\cup$ j c c' |
| 4 ET ( e $_{loc}$ j c ) ( e $_{loc}$ j c') | 4 $\mathbf{B}$ $\mathbf{B}^2$ e $_{loc}$ j $\cup$ c c' |

5 $\mathbf{B}^2$ ( e $_{loc}$ j ) $\cup$ c c'
6 e $_{loc}$ j ( $\cup$ c c' )

Autrement dit : $\boldsymbol{\times}$ [ ET, e $_{loc}$ ] $\cong$ $\boldsymbol{\times}'$ [ e $_{loc}$ , $\cup$ ]
avec : $\boldsymbol{\times}$ = $\mathbf{B}$ $\mathbf{B}$ $\phi$ et $\boldsymbol{\times}'$ = $\mathrm{C}_{(2)}$ $\mathbf{B}$ $\mathbf{B}^2$

4.2. De même, la loi sémantique :

$$\cup\ (\ \delta\ n\ c)\ (\ \delta\ n'\ c') = \delta\ (\ \mathrm{ADD}\ n\ n'\ )\ (\ \cup\ c\ c')$$ [1]

*"La réunion disjointe de deux collections a pour cardinal la somme des cardinaux de chaque collection ".*

a pour expression dans le formalisme applicatif :

$$\mathbf{X}\ [\ \cup, \delta, n, n'\ ] \cong \mathbf{X'}\ [\ \delta, \mathrm{ADD}, n, n', \cup\ ]$$

avec : $\mathbf{X} = \mathbf{H}_{(2)}\ \psi$ et $\mathbf{X'} = \mathbf{B^2}_{(4)}\ \mathbf{B^2}$ [2].

En effet, nous avons les réductions suivantes :

| | |
|---|---|
| 1 $\mathbf{H}_{(2)}\ \psi\ \cup\ \delta\ n\ n'\ c\ c'$ | 1 $\mathbf{B^2}_{(4)}\ \mathbf{B^2}\ \delta\ \mathrm{ADD}\ n\ n'\ \cup\ c\ c'$ |
| 2 $\mathbf{B^2}\ \mathbf{H}\ \psi\ \cup\ \delta\ n\ n'\ c\ c'$ | 2 $\mathbf{B^2}\ (\ \mathbf{B^2}\ \delta\ \mathrm{ADD}\ n\ \ n'\ )\ \cup\ c\ c'$ |
| 3 $\mathbf{H}\ (\ \psi\ \cup\ \delta\ )\ n\ n'\ c\ c'$ | 3 $\mathbf{B^2}\ \delta\ \mathrm{ADD}\ n\ n'\ (\ \cup\ c\ c')$ |
| 4 $\psi\ \cup\ \delta\ (\ n\ c\ )\ (\ n'\ c')$ | 4 $\delta\ (\ \mathrm{ADD}\ n\ n'\ )\ (\ \cup\ c\ c')$ |
| 5 $\cup\ (\ \delta\ n\ c\ )\ (\ \delta\ n'\ c')$ | |

4.4.3. La loi sémantique entre primitives : $\cup\ (\ \Delta\ c\ c'\ )\ c' = c$ où $c' \subset c$ [3] a pour expression dans le formalisme applicatif :

$$\mathbf{W}_{(3)}\ \mathbf{B^2}\ \cup\ \ \Delta\ c\ c' = \mathbf{B}\ \mathbf{K}\ \mathbf{I}\ c\ c'$$ [4]

4.4.4. De même, la loi sémantique :

SUC ( MODIF état1 état2 ) ( MODIF état2 état3 ) = MODIF état1 état3

*"Dans une succession de deux changements d'états, le changement obtenu a pour état initial celui du premier changement, l'état final du premier changement est l'état initial du second, son état final est l'état final du deuxième changement ".*

a pour expression dans le formalisme applicatif :

$$\mathbf{X}\ [\ \mathrm{SUC}, \mathrm{MODIF}\ ] \cong \mathbf{X'}\ [\ \mathrm{MODIF}\ ]$$

avec : $\mathbf{X} = \mathbf{B}\ \mathbf{B^2}\ \mathbf{W}_{(2)}\ \mathbf{H}\ \psi$ et $\mathbf{X'} = \mathbf{B}\ \mathbf{K}$ [2]

---

[1] Ici la fonction de détermination $\delta$ (n) signifie : "de quantité n", provenant du concept "est-en-quantité n" de type sémantique $\mathfrak{F}$ **c** $\mathfrak{h}$.

[2] **H** est le combinateur tel que : $\mathbf{H}\ f\ x\ y\ z\ t = f\ (\ x\ z\ )\ (\ y\ t\ )$.

[3] $\Delta$ est défini en (3.1.2).

4 Résultat obtenu grâce aux réductions suivantes :

| | |
|---|---|
| 1 $\mathbf{B^3}\ \mathbf{W}\ \mathbf{B^2}\ \cup\ \ \Delta\ c\ c'$ | 1 $\mathbf{B}\ \mathbf{K}\ \mathbf{I}\ c\ c'$ |
| 2 $\mathbf{W}\ (\ \mathbf{B^2}\ \cup\ \ \Delta\ c\ )\ c'$ | 2 $\mathbf{K}\ (\ \mathbf{I}\ c\ )\ c'$ |
| 3 $\mathbf{B^2}\ \cup\ \ \Delta\ c\ c'\ c'$ | 3 $\mathbf{I}\ c$ |
| 4 $\cup\ (\ \ \Delta\ c\ c'\ )\ c'$ | |

[2] **H** est le combinateur tel que : $\mathbf{H}\ f\ x\ y\ z\ t = f\ (x\ z)\ (y\ t)$. Le résultat est obtenu grace aux réductions suivantes :

| | |
|---|---|
| 1 $\mathbf{B}\ \mathbf{B^2}\ \mathbf{W}_{(2)}\ \mathbf{H}\ \psi$ SUC MODIF état1 état2 état3 | 1 $\mathbf{B}\ \mathbf{K}$ MODIF état1 état2 état3 |
| 2 $\mathbf{B^2}\ (\ \mathbf{W}_{(2)}\ \mathbf{H}\ )\ \psi$ SUC MODIF état1 état2 état3 | 2 $\mathbf{K}$ (MODIF état1) état2 état3 |

4.4.5. L'on peut également exprimer dans le même formalisme des lois sémantiques entre opérateurs telles que :

A-DE-MOINS-QUE j' ( δ n c ) j = A-DE-PLUS-QUE j ( δ n c ) j'
*"j a n éléments de plus que j' "* équivaut à *" j' a n éléments de moins que j "*

En effet, nous avons les réductions suivantes :

1 $\mathbf{C}_{(2)}$ $\mathbf{C}_{(2)}$ **C** A-DE-PLUS-QUE j' ( δ n c ) j

Posons f = A-DE-PLUS-QUE
2 $\mathbf{C}_{(2)}$ $\mathbf{C}_{(2)}$ **C** f a b c
3 $\mathbf{B}^2$ **C** $\mathbf{C}_{(2)}$ **C** f a b c
4 **C** ( $\mathbf{C}_{(2)}$ **C** f ) a b c
5 $\mathbf{C}_{(2)}$ **C** f b a c
6 $\mathbf{B}^2$ **C C** f b a c
7 **C** ( **C** f b ) a c
8 **C** f b c a
9 f c b a
10 A-DE-PLUS-QUE j ( δ n c ) j'

d'où $\mathbf{C}_{(2)}$ $\mathbf{C}_{(2)}$ **C** A-DE-PLUS-QUE j' ( δ n c ) j = A-DE-MOINS-QUE j' ( δ n c ) j

4.4.6. Considérons la loi sémantique entre opérateurs :

ET ( JOUE-PARTIE-AVEC j' c j ) ( GAGNE c j ) =
ET ( JOUE-PARTIE-AVEC j' c j ) ( PERD c j' )

*"Sachant que j et j' jouent une partie de billes, " j' gagne la collection c "* équivaut à *" j perd la collection c "*.

Nous avons les réductions suivantes :

1 $\phi^2$ ( $\phi$ ET ) ( JOUE-PARTIE-AVEC ) ( **K** GAGNE ) j' c j
2 **B** $\phi$ $\phi$ ( $\phi$ ET ) ( JOUE-PARTIE-AVEC ) ( **K** GAGNE ) j' c j
3 $\phi$ ( $\phi$ ( $\phi$ ET )) ( JOUE-PARTIE-AVEC ) ( **K** GAGNE ) j' c j
4 $\phi$ ( $\phi$ ET ) ( JOUE-PARTIE-AVEC j' ) ( **K** GAGNE j' ) c j
5 $\phi$ ET ( JOUE-PARTIE-AVEC j' c ) ( **K** GAGNE j' c ) j
6 ET ( JOUE-PARTIE-AVEC j' c j ) ( GAGNE c j )

1 $\phi^2$ ( $\phi$ ET ) ( JOUE-PARTIE-AVEC ) ( $\mathbf{K}_{(3)}$ C PERD ) j' c j
2 **B** $\phi$ $\phi$ ( $\phi$ ET ) ( JOUE-A-AVEC ) ( $\mathbf{K}_{(3)}$ C PERD ) j' c j
3 $\phi$ ($\phi$ ( $\phi$ ET )) ( JOUE-PARTIE-AVEC ) ( $\mathbf{K}_{(3)}$ C PERD ) j' c j
4 $\phi$ ( $\phi$ ET ) ( JOUE-PARTIE-AVEC j' ) ( $\mathbf{K}_{(3)}$ C PERD j' ) c j

---

3 $\mathbf{W}_{(2)}$ **H** ( ψ SUC MODIF ) état1 état2 état3 — 3 MODIF état1 état3
4 $\mathbf{B}^2$ **W H** ( ψ SUC MODIF ) état1 état2 état3
5 **W** ( **H** ( ψ SUC MODIF ) état1) état2 état3
6 **H** ( ψ SUC MODIF ) état1 état2 état2 état3
7 ψ SUC MODIF ( état1 état2 ) ( état2 état3 )
8 SUC ( MODIF état1 état2 ) ( MODIF état2 état3 )

7 ET ( JOUE-PARTIE-AVEC j' c j ) ( $\mathbf{B^3 K}$ C PERD j' c j )
8 ET ( JOUE-PARTIE-AVEC j' c j ) ( **K** ( C PERD j' c ) j )
9 ET ( JOUE-PARTIE-AVEC j' c j ) ( C PERD j' c )
10 ET ( JOUE-PARTIE-AVEC j' c j ) ( PERD c j' )

d'où

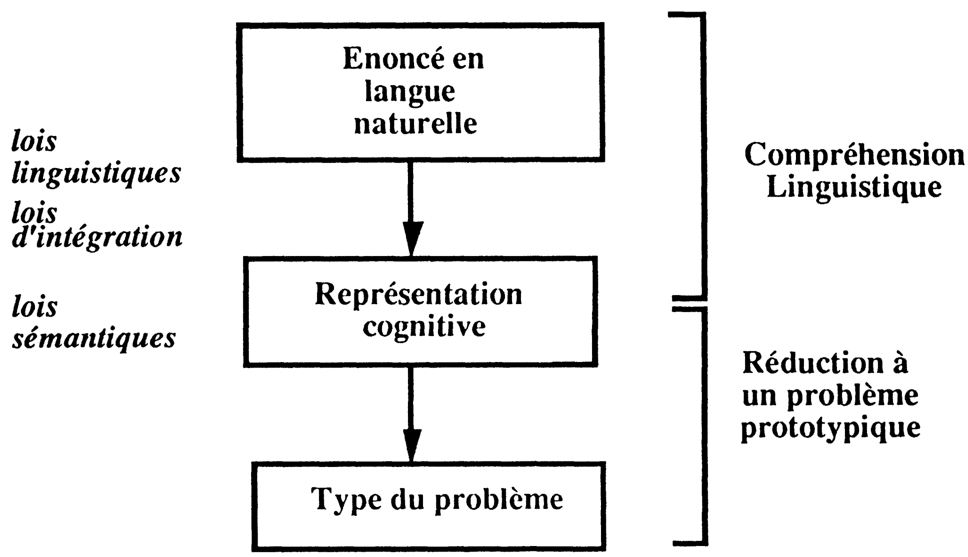
$\phi^2$ ( $\phi$ ET) ( JOUE-PARTIE-AVEC) ( **K** GAGNE) j' c j =
$\phi^2$ ( $\phi$ ET) ( JOUE-PARTIE-AVEC) ( $\mathbf{K}_{(3)}$ C PERD) j' c j

## 5. UNE MODELISATION DE LA COMPREHENSION D'ÉNONCES ADDITIFS

La modélisation mathématique de la compréhension linguistique d'énoncés additifs que nous venons de présenter permet d'expliciter réellement les processus d'un même signifié en signifiants différents, et ainsi de prendre en compte, de manière opératoire, la *congruence* ou *non-congruence sémantique* des énoncés d'une famille paraphrastique (D. Guin [15], § (3.2.3)). Cette *multi-représentation* des connaissances (§ (2.3)) est fondamentale : J. Pitrat ([17]) a souligné l'intérêt de disposer de *plusieurs représentations* pour un même objet, la représentation la plus adaptée à l'énoncé du problème n'étant pas forcément la plus adaptée pour la résolution.

Le formalisme de la Grammaire Applicative Universelle compatible avec des représentations informatiques, est un outil de représentation des connaissances particulièrement bien adapté à notre domaine, car il nous permet de prendre en compte d'une part les règles de transformations correspondant aux processus cognitifs élémentaires mis en évidence dans l'analyse cognitive (D. Guin [15], § 3), et d'autre part de distinguer les compréhensions linguistique et mathématique, de les réunir dans un même formalisme, et de les articuler.

Nous pouvons résumer l'activité de compréhension d'un problème additif dans le schéma suivant :




Afin de préciser l'architecture du niveau fonctionnel, nous reprendrons l'exemple (6.2) de (D. Guin [15] en y intégrant le formalisme mathématique que nous avons présenté dans les paragraphes précédents :

Exemple : *Pierre a joué deux parties de billes avec Paul. A la première partie, il a gagné 12 billes, à la seconde partie Paul en a gagné 3. Que s'est-il passé en tout ?*

La représentation cognitive de ce problème s'obtient à partir des archétypes cognitifs GAGNE et PERD, et la primitive sémantique de composition de deux processus :

| état initial | $\longrightarrow$ | état final |
|---|---|---|
| Pierre a une collection de billes | premier changement d'états | Pierre a cette collection de billes dans laquelle on insère 12 billes |

succession de deux changements d'états $\downarrow$

| état initial | $\longrightarrow$ | état final |
|---|---|---|
| Paul a une collection de billes | deuxième changement d'états | Paul a cette collection de billes dans laquelle on insère 3 billes |

Ce qui s'exprime dans le formalisme applicatif par la représentation cognitive :

SUC MODIF1 MODIF2
*succession de deux changements d'états*

avec INIT MODIF1 FIN

$e_{poss}\ j\ b$ ---------> $e_{poss}\ j\ (INS\ (\delta\ 12\ a)\ b)$ [1]

où b : = *une collection de billes*, j : = *Pierre*, $\delta$ 12 a : = *une collection de 12 billes.*

INIT MODIF2 FIN

$e_{poss}\ p\ d$ ---------> $e_{poss}\ p\ (INS\ (\delta\ 3\ e)\ d)$

où d : = *une collection de billes*, p : = *Paul*, $\delta$ 3 e : = *une collection de 3 billes.*

L'identification de l'état final du premier changement d'états et l'état initial du second changement d'états ne peut être immédiatement effectuée, la recherche du type de problème nécessite donc un processus cognitif de *paraphrase* permettant de changer de représentation. Ce processus cognitif se traduit par la *loi sémantique* traduite en langue naturelle :

*"Sachant que j et j' jouent une partie de billes, j' gagne la collection c équivaut à j perd la collection c ".*

[1] Ce qui s'écrit avec les parenthèses : $e_{poss}\ j\ (\ (INS\ (\delta\ (12)\ (a)))\ (b))$. Dans ce paragraphe, pour ne pas trop surcharger les expressions, nous réduisons l'utilisation des parenthèses.

Ce qui s'exprime dans le formalisme applicatif par la loi sémantique entre opérateurs :

ET ( JOUE-PARTIE-AVEC j' c j ) ( GAGNE c j' ) =
ET ( JOUE-PARTIE-AVEC j' c j ) ( PERD c j )

La compilation des représentations intermédiaires se traduit ici par une substitution dans la représentation du deuxième changement d'états :

| état initial | ——→ | état final |
|---|---|---|
| Pierre a une collection de billes | premier changement d'états | Pierre a cette collection de billes dans laquelle on insère 12 billes |

succession de deux changements d'états ↓

| état initial | ——→ | état final |
|---|---|---|
| Pierre a une collection de billes | deuxième changement d'états | Pierre a cette collection de billes à laquelle on enlève 3 billes |

Ce qui s'exprime dans le formalisme applicatif par la nouvelle représentation cognitive :

SUC MODIF1 MODIF2
*succession de deux changements d'états*

avec INIT MODIF1 FIN
e poss j b ---------> e poss j (INS (δ 12 a) b)
où b : = *une collection de billes*, j : = *Pierre*, δ 12 a : = *une collection de 12 billes.*

INIT MODIF2 FIN
e poss j f ---------> e poss j (ENL (δ 3 e) f)
où f : = *une collection de billes*, δ 3 e : = *une collection de 3 billes.*

Le processus cognitif de *composition* se traduit alors par l'application de la *loi sémantique* :

*"Dans une succession de deux changements d'états, le changement obtenu a pour état initial celui du premier changement, l'état final du premier changement est l'état initial du second, son état final est l'état final du deuxième changement ".*

Ce qui s'exprime dans le formalisme applicatif par la loi sémantique entre primitives sémantiques :

SUC ( MODIF état1 état2 ) ( MODIF état2 état3 ) = MODIF état1 état3

Par identification de l'état intermédiaire, nous obtenons la représentation :

| état initial | | état final |
|---|---|---|
| Pierre a une collection de billes | $\longrightarrow$ Changement d'états | Pierre a cette collection de billes dans laquelle on insère 12 billes, puis à laquelle on enlève 3 billes |

Ce qui s'exprime dans le formalisme applicatif par la nouvelle représentation :

$$\begin{array}{ccc} \text{INIT} & \text{MODIF} & \text{FIN} \\ {}^{e}\text{poss}\ j\ b & \text{--------->} & {}^{e}\text{poss}\ j\ (\,(\text{ENL}\ (\delta 3\ e))\ (\text{INS}\ (\delta 12\ a)\ b)\,) \end{array}$$

où b, a, e : = *collections de billes*, j : = *Pierre* .
On applique ensuite les *lois sémantiques* qui s'expriment en langue naturelle sous la forme :

*"Soit une partie p de la collection c : le nombre d'éléments de la collection à laquelle on a ôté p est obtenu en retranchant le nombre d'objets de p au nombre d'objets de c ."*

Ce qui s'exprime dans le formalisme applicatif par la loi sémantique :

$$(\text{ENL}\ (\delta\ n'\ p))\ (\delta\ n\ c)\ = \delta\ (\ (\text{RET}\ n')\ (n)\ )\ (\ \Delta\ c\ p)$$

*"Soit une collection c, on y insère une collection c' : le nombre d'objets de la collection ainsi créée est obtenu en ajoutant au nombre d'objets de c le nombre d'objets de c'."*

Ce qui s'exprime dans le formalisme applicatif par la loi sémantique :

$$(\text{INS}\ \ (\delta\ n'\ c'))\ (\delta\ n\ c)\ = \delta\ (\ (\text{AJOUT}\ n')\ (n)\ )\ \ (\ c \cup c')$$

La nouvelle représentation cognitive obtenue est donc :

$$\begin{array}{ccc} \text{INIT} & \text{MODIF} & \text{FIN} \\ {}^{e}\text{poss}\ j\ b & \text{--------->} & {}^{e}\text{poss}\ j\ (\,(\delta\ (\text{RET}\ 3\ \text{o}\ \text{AJOUT}\ 12)\ (x))\ f) \end{array}$$

où x : = *nombre d'éléments de la collection b,* f : = *collection de billes.*
Le cardinal de la collection de l'état final est donc alors obtenu en ajoutant 12, puis en retranchant 3 au cardinal de la collection initale, ce qui nous permet de déterminer le type du problème :

$$\text{RET}\ 3\ \ \text{o}\ \text{AJOUT}\ \ 12\ \ = ?\ \ \textit{type 6.1.1}$$

C'est donc un problème de la sixième catégorie (composition de AJOUT n et RET n'), qui sera résolu grâce à la loi sémantique correspondant à un théorème-en-acte :

$$\text{RET}\ n'\ \text{o}\ \ \text{AJOUT}\ \ n = \ \text{AJOUT}\ \ n\ \ \text{o}\ \text{RET}\ n' = \text{AJOUT}\ (\text{SOUS}\ \ n\ n')\quad \text{si}\ n' < n$$

Notons que cette modélisation permet de prendre en compte, d'une part les représentations ensemblistes mises en évidence dans les stratégies de compréhension grâce aux primitives sémantiques définies sur les collections *enlever* et *insérer*, et d'autre part les stratégies de comptage. Le lien entre ces deux types d'opérateurs est explicité par les lois sémantiques que nous venons d'utiliser, par exemple :

*"Soit une collection c, on y insère une collection c' : la collection ainsi créée est la réunion disjointe des collections c et c', son nombre d'objets est obtenu en ajoutant au nombre d'objets de c le nombre d'objets de c'."*

Ce qui s'exprime dans le formalisme applicatif par la loi sémantique :

$$(\text{INS} ( \delta n' c' )) ( \delta n\ c ) = \delta ( (\text{AJOUT } n') (n) ) ( c \cup c' )$$

## 6. CONCLUSION

Nous avons présenté, pour un énoncé additif, les différentes étapes d'une modélisation de la compréhension des problèmes additifs basée sur la notion d'opérateur : l'élaboration des représentations applicatives et cognitives de l'énoncé additif, le choix à partir de la représentation cognitive du problème prototypique, les lois linguistiques et sémantiques intervenant dans le traitement des paraphrases nécessaires à la résolution du problème. Pour celà, nous utilisons le formalisme applicatif, où les compilations des représentations intermédiaires sont exprimées grâce aux combinateurs de la logique combinatoire.

Cette modélisation cognitive de l'activité de compréhension de l'élève *"expert"* est une tentative d'intégrer les différents points de vue des linguistes, psychologues cognitifs, didacticiens et informaticiens. En effet, elle prend en compte les différents processus élémentaires cognitifs mis en évidence dans le domaine additif (l'interprétation des formes verbales, des marques temporelles dans la compréhension linguistique de l'énoncé, la composition et la paraphrase dans la réduction à un problème prototypique, l'utilisation d'opérateurs sur les entiers naturels et l'importance des valeurs relatives des opérandes dans la définition du type de problème, enfin la recherche de théorème-en-acte pour la résolution du problème).

Il s'agissait de décrire complètement la conduite de l'élève expert dans la situation de résolution d'un problème additif, sous le point de vue du traitement de l'information, afin que cette description puisse être intégrable à un environnement informatique. Cependant, cette modélisation cognitive pourrait être dès maintenant expérimentée avant la réalisation informatique d'un environnement d'apprentissage. En effet, la *verbalisation* des différentes étapes nécessaires pour déterminer le type du problème (lois linguistiques, sémantiques) peut constituer la base explicative d'un enseignement basé sur les *méthodes* de résolution.

Puisque nous disposons de l'expertise complète (mathématique et cognitive) du modèle expert exprimée à l'aide de compilation de représentations cognitives, nous avons entrepris de réaliser l'implémentation de telles représentations sur ordinateur : les simulations des traitements présentés dans le paragraphe 5 sont réalisées grâce à une représentation objet où les opérateurs sont représentés par des classes d'objets et les compilations des représentations intermédiaires (qui sont les objets) sous formes de méthodes. Il conviendra ensuite de spécifier l'interaction didactique du prototype expert en menant des expérimentations en situation de classe pour développer le module pédagogique et des explications adaptées aux conceptions des élèves. Ce travail nécessite évidemment une collaboration entre didacticiens et informaticiens.

L'approche *opératoire* pour la modélisations cognitive paraît très intéressante, aussi elle devrait pouvoir être généralisée à d'autres domaines de connaissance si les résultats des recherches linguistiques, cognitives et didactiques permettent d'expliciter les processus élémentaires sous-jacents aux activités à modéliser. Le concept d'opérateur pourrait jouer un rôle *fédérateur* par sa capacité à exprimer ces processus cognitifs mis en évidence dans les différentes disciplines. En effet, nous avons montré que c'est un outil puissant, adéquat à la description explicite des schèmes [1] (G. Vergnaud [22]) sous forme de compilation des représentations cognitives. Cette description explicite est d'autant plus intéressante qu'elle est implémentable sur ordinateur, ce qui devrait intéresser fortement les chercheurs en Intelligence Artificielle.

## BIBLIOGRAPHIE


[1] AJDUKIEWICZ K., "Die syntaktishe Konnexität", *Studia Philosophica*, vol 1, (1935), 1-27.

[2] BAR-HILLEL Y., "A quasi-arithmetical notation for syntactic description", *Language*, vol 29, (1953), 47-58, traduit in *Langages*, vol 9, (1968), 9-22.

[3] BIRKHOFF G., "On the structure of abstract algebras", *Proc. Cambridge Philos. Soc.*, vol 31, (1935), 433-454.

[4] CHURCH A., *The Calculi of Lambda Conversion* , Princeton University Press, 1941.

[5] CURRY H. B., FEYS R., *Combinatorial Logic*, vol 1, North Holland, 1958.

[6] DESCLES J.- P., "Opérateurs/Opérations, méthodes intrinsèques en informatique fondamentale ; applications à la linguistique et aux bases de données", thèse d'état, Université René Descartes, 1980.

[7] DESCLES J.- P., "Réseaux sémantiques : La nature logique et linguistique des relateurs", *Langages*, vol 87, (1987), 55-78.

[8] DESCLES J.-P., "Théorème de Church-Rosser et structuration des Langues naturelles", *Mathématiques Informatique et Sciences humaines*, vol 103, (1988), 67-92.

[9] DESCLES J.-P., *Langages applicatifs, langues naturelles et cognition*, Paris, Hermès, 1990.

[10] DESCLES J.-P., FROIDEVEAUX CH., "Axiomatisation de la notion de repérage abstrait", *Mathématiques et Sciences humaines*, vol 78, (1982), 73-119.

[11] DESCLES J.-P., ABRAHAM M., PIOTROWSKI D., SEGOND F., "Langage naturel et représentations cognitives : un problème d'architecture et de compilation", *Actes du Colloque de l'A.R.C*, (1990), 290-303.

[12] FREGE G., *Grundgesetze der Arithmetik, begriffsschriftlich abgeleitet*, Band I. Jena, 1893 ; traduction en anglais par M. Furth : *The basic Laws of Arithmetic, exposition of the system*, Univ. of California Press, 1964.

[13] GINISTI J.-P., "Présentation de la Logique Combinatoire en vue de ses applications", *Mathématiques Informatique et Sciences humaines*, vol 103, (1988), 45-66.

[14] GRIZE J.-B., *Logique Moderne III*, Paris, Mouton, Gauthier-Villars, 1973.

[15] GUIN D., "La notion d'opérateur dans une modélisation cognitive de la compréhension des problèmes additifs", *Mathématiques, Informatique et Sciences humaines*, vol 113, (1991), 5-33.

[16] PIOTROWSKI D., "Systèmes applicatifs et analyse informatique du langage naturel. Recherche sur les fondements du modèle "Grammaire Applicative et Cognitive", Thèse de doctorat de l'Ecole des Hautes Etudes en Sciences Sociales, Paris, 1990.

[17] PITRAT J., *Métaconnaissance, futur de l'IA*, Paris, Hermès, 1990.

[18] SCHANK R. K., *Conceptual information Processing*, North-Holland, 1975.

[19] SCHÖNFINKEL M., "Uber die Bausteine der Mathematischen Logik", *Math. Annalen*, vol 92, (1924), 355-366.

[20] VERGNAUD G., *L'enfant, la mathématique et la réalité*, Peter Lang, Berne, 1981.


---

[1] G. Vergnaud définit le concept de schème comme "l'organisation invariante de la conduite pour une classe de situations donnée". C'est dans le schème que résident "les éléments cognitifs qui permettent à l'action du sujet d'être opératoire"; certains éléments du schème des problèmes additifs mis en évidence par G. Vergnaud, en particulier les théorèmes-en-acte, ont été succinctement présentés dans D. Guin [15] au § 3.2.1.

[21] VERGNAUD G., "Questions de représentation et de formulation dans la résolution de problèmes mathématiques", *Annales de Didactique et de sciences Cognitives*, IREM de Strasbourg, vol 1, (1988), 33-55.
[22] VERGNAUD G., "La théorie des champs conceptuels", *Recherches en Didactique des Mathématiques,* La pensée Sauvage, vol 10, n° 2/3, (1990), 133-170.