MATHÉMATIQUES ET SCIENCES HUMAINES

FRÉDÉRIQUE SEGOND

Approches des grammaires catégorielles

Mathématiques et sciences humaines, tome 110 (1990), p. 47-60
<http://www.numdam.org/item?id=MSH_1990__110__47_0>

© Centre d'analyse et de mathématiques sociales de l'EHESS, 1990, tous droits réservés.

L'accès aux archives de la revue « Mathématiques et sciences humaines » (http://msh.revues.org/) implique l'accord avec les conditions générales d'utilisation (http://www.numdam.org/conditions). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM
Article numérisé dans le cadre du programme
Numérisation de documents anciens mathématiques
http://www.numdam.org/

Math. Inf. Sci. hum., (28e année, n°110, *1990*, p.47-60)

APPROCHES DES GRAMMAIRES CATEGORIELLES

Frédérique SEGOND[1]

RÉSUMÉ - *Etant donné l'importance que prennent les grammaires catégorielles dans le domaine de la linguistique computationnelle, il nous a semblé intéressant de dresser un panorama sur cette question. Nous espérons fournir, aux chercheurs intéressés, un matériau de base susceptible de les aider à approfondir par eux-mêmes le sujet.*

ABSTRACT - *Overview of categorical grammars.*
Given the importance of categorial grammars in the field of computational linguistics, we find it interesting to give an overview of this matter. We hope to provide interested researchers with some basic but useful material.

Nous traitons dans cet article des grammaires catégorielles telles qu'elles nous sont apparues lors de la 4ème conférence de l'ACL[2] européen.

Signalons dès maintenant que les conférences qui ont eu lieu à Manchester lors du quatrième congrès européen de l'ACL ont également abordé des sujets autres que celui des grammaires catégorielles ; il a été question de grammaires d'unification, de grammaires d'arbres adjoints, de grammaires génératives, ainsi que de la traduction, et de génération de textes. Nous n'en parlerons pas ici, laissant le soin au lecteur intéressé de se reporter aux articles contenus dans les actes de la conférence cités en bibliographie.

Signalons enfin que le but de cet article est essentiellement d'être une base d'information sur les grammaires catégorielles ; ceci afin de permettre aux chercheurs non spécialistes, de posséder une base théorique pour comprendre, plus en profondeur, les articles traitant du sujet.

Rapide historique.

Les Grammaires Catégorielles trouvent leur origine dans les années trente (1935) avec les travaux de K. Ajdukiewicz, qui le premier propose un système formalisé de vérification de la connexion syntaxique des expressions[3]. A la base de ce système se trouvent les idées du philosophe Edmund Husserl sur la théorie des catégories de signification.

---

[1] CAMS (CNRS) / Centre scientifique IBM-France.

[2] Association for Computational Linguistic.

[3] K. Ajdukiewicz [2].

A la suite des travaux d'Ajdukiewicz on citera ceux de Y. Bar-Hillel qui, dans les années cinquante, propose une grammaire de même type qu'il a quelque peu enrichie (notion d'ordre, etc.)[4].

Dans le même temps (1958) Lambek propose un "calcul syntaxique" pour la langue naturelle[5]. Enfin, dans les années soixante, nous trouvons les travaux de S. K. Shaumyan qui, partant des idées de la grammaire catégorielle et l'enrichissant à l'aide de la logique combinatoire, construit un système aujourd'hui connu sous le nom de "Grammaire Applicative"[6].

Z. Guentcheva et J.-P. Desclés développent depuis le modèle de la Grammaire Applicative[7]. Ce dernier anime, à l'EHESS, une équipe qui travaille sur le sujet.

On constate depuis le début des années 80 un regain d'intérêt pour ce type de grammaires. Beaucoup de chercheurs européens les utilisent ; ceci, sans doute à cause de leur côté "mathématique", qui les rend séduisantes pour une implémentation informatique. La meilleure preuve en est la dernière conférence de l'ACL européen (tenue début avril 89 à Manchester), où de nombreux exposés traitaient des grammaires catégorielles.

Signalons plus particulièrement les chercheurs travaillant dans le cadre du projet ACORD, lui même faisant partie du projet européen ESPRIT qui se trouvaient à cette conférence. En effet, dans le cadre de ce projet, de nombreuses universités européennes telles que les universités de Clermont-Ferrand, de Stuttgart, d'Edimbourg, travaillent en commun avec des entreprises privées sur l'implémentation des grammaires catégorielles.

Signalons enfin que l'été suivant (juillet 89) s'est tenu , en Arizona, un workshop sur ce même thème des grammaires catégorielles. Devant l'importance que prennent les grammaires catégorielles, il nous a semblé intéressant de faire un compte rendu sur la quatrième conférence de l'ACL européen.

Qu'est-ce qu'une Grammaire Catégorielle ?

Avant d'exposer l'état actuel des grammaires catégorielles, donnons une définition, la plus générale possible, de ces grammaires.

Définition. Une grammaire catégorielle est un quadruplet ordonné

$$<V, W, f, \varphi_0> \text{ où :}$$

- V est un ensemble de symboles appelé *dictionnaire terminal*,
- W est un ensemble fini de catégories syntaxiques appelées *catégories élémentaires*, W est appelé *dictionnaire de catégories*,
- f est une fonction qui, à chaque symbole terminal (élément de V) assigne un sous-ensemble fini de catégories f est appelée *fonction d'adjonction*,
- φ_0 est un élément de W appelé *catégorie principale*.

---

[4] Y. Bar-Hilllel [3] [4].

[5] Lambek [16].

[6] S. Shaumyan [23].

[7] J.P. Desclés [10] [11].

A partir de ces catégories de base on construit tout l'ensemble des catégories à l'aide des règles de construction suivantes :

Règles de construction.

1. Toute catégorie élémentaire est une catégorie.
2. Si α et β sont des catégories alors les expressions (α/β [8]) et ($\alpha\backslash\beta$ [9]) sont aussi des catégories.
3. Une catégorie est un élément formé en vertu de (1) ou de (2).

On nomme K(W) l'ensemble de toutes les catégories engendrées à partir des éléments de W.

Ces catégories élémentaires étant posées, nous avons des règles de réduction entre les catégories.

Règles de réduction immédiate.

Soient α et β des catégories sur W on a :

1. réduction à droite : $(\alpha/\beta)\,\beta \rightarrow \alpha$
2. réduction à gauche : $\alpha(\alpha\backslash\beta) \rightarrow \beta$

Règle de réduction.

Soient φ et ψ deux chaînes de catégories sur W ; on dit que φ se réduit à ψ si ψ s'obtient à partir de φ par une suite de réductions immédiates.

Ce modèle mathématique ayant été posé, donnons son interprétation dans les langues naturelles.

. V est le dictionnaire de la langue considérée,
. "s" et "t" sont les éléments de W [10],
. φ_0 est la catégorie principale "s",
. f est la fonction qui à chaque mot du dictionnaire associe sa catégorie (représentée par un code donné dans le dictionnaire)[11].

Les deux catégories de base sont donc "s" pour *sentence* (proposition) et "t" pour *term* (désignant ce que l'on a coutume d'appeler un groupe nominal).

A partir de ces deux catégories de base nous pouvons engendrer un ensemble infini de catégories à l'aide des règles de construction énoncées plus haut.

---

[8] On lit "α sur β".

[9] On lit "α sous β".

[10] Ces catégories de base sont celles que l'on trouve dans la plupart des articles mais elles peuvent être plus ou moins nombreuses.

[11] Remarquons que les mots du dictionnaire pouvant avoir plusieurs catégories grammaticales, la fonction f peut prendre un ensemble fini de valeurs.

Une catégorie syntaxique est donc envisagée comme une fonction qui s'applique à une autre catégorie syntaxique pour en former une troisième.

Donnons des exemples :

. Un verbe intransitif peut être considéré comme une fonction qui s'applique à un groupe nominal (le sujet) pour former une phrase.
. Un verbe transitif peut être considéré comme une fonction qui s'applique à un groupe nominal (le complément d'objet direct ou COD) pour former une fonction (groupe verbal attendant un sujet) qui s'appliquera à un groupe nominal (le sujet) pour former une phrase, etc.

De ce fait, nous allons pouvoir assigner un ou plusieurs types à chaque catégorie syntaxique.

Si nous reprenons les exemples précédents nous pouvons dire que les verbes intransitifs peuvent être codés $t \backslash s$ et les verbes transitifs peuvent être codés $(t \backslash s)/t$ [12].

Nous voyons que, contrairement aux grammaires dites de constituants qui indiquent, dans un grand nombre de règles, quelle est la catégorie qui se combine avec telle autre pour former tel type de catégorie, la grammaire catégorielle indique dans le codage même des catégories syntaxiques avec quel type de catégorie elles vont se combiner pour former quel type de catégorie syntaxique comme résultat. Les règles de la grammaire à proprement parler sont alors fort simples, ce sont les règles de réduction que nous avons données précédemment.

Faisons fonctionner ce formalisme sur un exemple :

Considérons la phrase :

Joseph aime Marie

.Les types associés à chacune des expressions de cette phrase sont les suivants :

. Joseph : t
. Marie : t
. aime : $(t \backslash s)/t$, $t \backslash s$ [13]

Une fois le "bon type" associé au bon mot, on applique les règles de simplification et on obtient :

Joseph	aime	Marie
t	$t \backslash s / t$	t

Joseph	(aime Marie)
t	$t \backslash s$

(Joseph (aime Marie))
s

Ceci étant posé, on s'aperçoit rapidement que ces deux règles de simplification ne vont pas suffire à analyser les langues naturelles. Il faut y ajouter d'autres règles ou des métarègles pour obtenir un système qui permette de traiter avec le langage naturel.

---

[12] L'expression $t \backslash s$ se lit : "t sous s" pour signifier qu'un verbe intransitif attend son argument à gauche : le sujet. De la même façon, l'expression $(t \backslash s)/t$ se lit : "t sous s, sur t", pour signifier qu'un verbe transitif attend son premier argument le COD à droite et son deuxième argument le sujet à gauche.

[13] Remarquons qu'un même item lexical peut avoir plusieurs catégories syntaxiques donc plusieurs codages. Ainsi, "aime" peut être un verbe transitif et un verbe intransitif.

Les Grammaires Catégorielles à Manchester.

Nous en arrivons donc à l'exposé de l'état des grammaires catégorielles tel qu'il a été présenté lors de l'ACL de Manchester.

Il semble que la liste des types de base soit la même que celle que nous avons énoncée dans la définition, si ce n'est que "t" est remplacé par "n". Les types de base sont donc s et n.

Cependant nous trouvons, d'autres types. Les plus courants sont : vp, np, pp, et ap définis de la manière suivante :

. vp sert en fait à désigner un verbe intransitif ; on peut donc également l'exprimer à l'aide de s et n.
. np semble désigner ce que j'appellerai un groupe nominal "complet" (avec déterminant ou bien nom propre seul).
. pp désigne un groupe prépositionnel.
. ap désigne un groupe adjectival.

A partir de ces types de base on définit les autres types syntaxiques de la façon suivante :

Définition. Un type syntaxique est un triplet de la forme :

<résultat, direction, argument>

où :

. le résultat et l'argument sont eux-mêmes des types syntaxiques,
. la direction est indiquée soit par " / " pour indiquer que l'argument se trouve à droite de l'opérateur, soit par " \ " pour indiquer que l'argument se trouve à gauche de l'opérateur[14].

Exemples :

Un verbe intransitif est codé : s \ np ou vp
Un verbe transitif est codé : (s \ np) / np ou vp / np
Un adverbe est codé : vp \ vp

Une fois les types définis nous avons des règles de simplification. Ces règles sont de différentes sortes : les premières sont celles que nous avons déjà exposées. Et sont appelées règles *d'application fonctionnelle.*

Les règles d'application fonctionnelle (functional application rules).

1. Application avant (>) : x/y y → x
2. Application arrière (<) : y x\y → x

Exemples :

```
 Jean      mange
NP         S\NP
-------------- <<
 S
```

[14] Nous nous permettons de faire remarquer au lecteur que bien que ces notations soient les mêmes que celles que nous avons exposées précédemment, elles n'ont pas la même signification.

Cet exemple nous montre que le verbe intransitif "manger" s'applique au groupe nominal NP qui se trouve à sa gauche. La règle de simplification que l'on a utilisée pour obtenir une phrase (S) est la règle de simplification arrière.

```
Jean      mange        la soupe
NP        (S\NP)/NP    NP
----------------------- <  >
NP        (S\NP)
--------------- <  <
```

Nous voyons que le verbe transitif "mange" attend un argument à droite (le NP "la soupe") ; nous obtenons, après avoir appliqué la règle d'application-avant un groupe verbal, "mange la soupe", qui attend un argument à gauche (le NP-sujet "Jean"). Et enfin, après avoir appliqué la règle d'application-arrière, on obtient une phrase.

En plus de ces règles d'application fonctionnelle, des *règles de composition fonctionnelle* sont posées.

Les règles de composition.

1. x/y y/z → x\z (> B))
2. x/y y\z → x\z (> Bx) (composition croisée avant)
3. y\z x\y → x\z (< B)
4. y/z x\y → x/z (< Bx) (composition croisée arrière)

Examinons de plus près la signification de ces règles.

Les deux premières règles signifient que si une expression qui attend pour argument à sa droite une expression de type "y" est suivie d'une expression qui forme comme résultat une expression type "y", alors on peut "ignorer" les expressions de type "y". Ainsi on peut directement tout réduire à une expression de type x\z, c'est-à-dire à une expression dont le type est celui d'un opérateur qui attend à sa gauche une expression de type z et qui forme comme résultat une expression de type x.

La différence entre ces deux règles réside dans le sens du "slash". Autrement dit ce qui diffère, c'est la place de l'argument "y" dans le type des deux expressions de départ. Dans la première règle, la deuxième expression attend son argument à droite ; dans la deuxième règle, elle attend son argument à gauche.

Les deux dernières règles signifient que si une expression qui forme comme résultat une expression de type "y" est suivie d'une expression qui attend comme argument à gauche une expression de type "y", alors on peut "ignorer" les expressions de type "y". Ainsi on peut directement tout réduire à une expression de type x\z ou x/z, c'est-à-dire à une expression qui attend une expression de type z comme argument à sa gauche ou à sa droite pour former une expression de type x comme résultat.

La différence entre ces deux règles réside dans le sens du "slash". Autrement dit ce qui diffère, c'est la place de l'argument "y" dans les le type des deux expressions de départ. Dans la première règle la première expression attend son argument à gauche, dans la deuxième règle elle attend son argument à droite. Remarquons que, dans ce cas, le sens du "slash" influe également sur le résultat. Ainsi, l'expression formée attend son argument "z" soit à droite, soit à gauche.

Exemple :

```
Jean      voit        souvent       le médecin
NP        VP/NP       VP\VP         NP
          .---------------------- < B (>Bx)
          VP/NP
          ---------------------------------- <>
          VP
--------------- <   <
  S
```

Nous voyons sur cet exemple qu'en appliquant la composition croisée avant au verbe transitif "voir" (VP/NP) et à l'adverbe "souvent" (VP\VP), on obtient un verbe transitif "voir souvent" (VP/NP). La suite des simplifications s'opère à l'aide des règles d'application : application-avant puis application-arrière.

Remarquons au passage que nous trouvons dans ces règles de composition le combinateur B de la logique combinatoire. D'autres combinateurs comme les combinateurs C et W sont utilisés par certains auteurs. Tous ces combinateurs ainsi que d'autres sont employés en grammaire applicative, mais à des fins différentes[15].

A ces règles d'application fonctionnelle et de composition s'ajoutent les règles de *type raising* ou *d'émergence de type* suivantes :

Règles d'émergence de type (Type raising)[16].

1. $x \rightarrow y/(y\backslash x)$ (> T)
2. $x \rightarrow y/(y/x)$ (> Tx)
3. $x \rightarrow y\backslash(y/x)$ (<T)
4. $x \rightarrow y\backslash(y\backslash x)$ (<Tx)

Autrement dit, lorsque l'on a une catégorie syntaxique de type "x" on peut toujours la remplacer par une catégorie syntaxique de type y/(y/x), y/(y\x), y\(y/x) ou bien (y\(y\x)).

Ainsi l'émergence du type permet de réanalyser l'argument comme un nouvel opérateur qui prendra comme argument l'opérateur qui devait lui être appliqué, avant qu'il n'y ait "type raising".

Dans la pratique il semble que l'on ne trouve que deux de ces règles :

1. *Règle d'émergence du sujet (>T)* :
   $np \rightarrow s\backslash(s\backslash np)$
2. *Règle d'émergence de l'objet (<T)* :
   $np \rightarrow vp/(vp\backslash np)$

La première de ces règles signifie que l'on peut considérer un NP se trouvant en position sujet comme une fonction qui agit sur le verbe intransitif pour former une phrase.

---

[15] Pour plus de détails, nous renvoyons le lecteur intéressé à l'ouvrage de Shaumyan, Desclés et de Guentcheva [11].

[16] Remarquons que certains auteurs préfèrent traduire *type raising* par "montée du type" [19]

De la même façon on peut considérer un NP se trouvant en position d'objet comme un opérateur qui agit sur un verbe transitif pour former un verbe qui attend un sujet.

Exemples :

```
Joseph      aime        Marie
NP          VP/NP       NP
                        --- <  T
                        VP/(VP\NP)
            ---------- <  <
            VP
------------------- <  <
S
```

Dans cet exemple il apparaît clairement que lorsque l'on applique la règle d'émergence du type de l'objet à l'expression Marie, expression qui dans ce cas est COD, cette expression change de type : du type NP elle passe au type vp/(vp\np) ; ainsi, elle devient un opérateur qui attend comme argument à sa gauche un verbe transitif.

Nous finissons par obtenir une phrase en appliquant deux fois de suite la règle d'application arrière.

Syntaxe-Sémantique.

A chacune des règles syntaxiques exposées plus haut corresponde une "règle sémantique".

Commençons par énoncer ces règles.

1. *Les règles d'application* :

   a. Application avant (>) :
      f : x/y y → x où fab = ab
   b. Application arrière (<) :
      b : y x\y → x où byx = xy

2. *Les règles de composition* :

   a. F : x/y G : y/z → BFG : x/z (> B)
   b. F : x/y G : y\z → BFG : x\z (> Bx)
   c. G : y\z F : x\y → BFG : x\z (< B)
   d. G : y/z F : x\y → BFG : x/z (< Bx)
   où B xyz = x(yz)

3. *Règles d'émergence de type (Type raising)* :
   a. x : g → y/(y\x) : Tg (> T)
   b. x : g → y/(y/x) : Tg (> Tx)
   c. x : g → y\(y/x) : Tg (< T)
   d. x : g → y\(y\x) : Tg (< Tx)
   où T yx = xy

Exemples :

L'analyse syntaxique de la phrase *Harry mange les pommes* nous donne :

```
Harry     mange          les pommes
NP        (S\NP)/NP      NP
          ---------------------------- <  f
          S\NP
------------------- <b
          S
```

A cette analyse syntaxique correspond l'analyse sémantique suivante[17] :

On commence par écrire explicitement quelle fonction on applique à quel syntagme, en s'appuyant sur l'analyse syntaxique précédente:

f(mange' les pommes')
b (Harry'(f (mange' les pommes')))

D'après les règles de réduction on a donc :

b (Harry' (mange' les pommes'))
((mange' les pommes') Harry')

Grâce à cette analyse sémantique nous savons que celui qui mange c'est *Harry* et que ce qui est mangé ce sont *les pommes*.

La pseudo-ambiguïté (spurious ambiguity)

Cependant, étant donné les règles syntaxiques qui ont été posées, on peut aussi analyser la phrase *Harry mange les pommes*, de la façon suivante (émergence de l'objet) :

```
Harry       .mange        les pommes
NP           VP/NP        NP
                          ----- < T
                          VP\(VP/NP)
            --------------------- <  b ·
              V
---------- <  b
 S
```

Sémantiquement on a donc :

T pommes'
b (T (les pommes') mange')
b(b (T (les pommes') mange') Harry')

En réduisant cette expression à l'aide des règles pos&es plus avant, on a :

b(b ((mange' les pommes') Harry'))
b(Harry' (mange' les pommes'))
((mange' les pommes') Harry')

[17].Un mot suivi du signe ' désigne l'interprétation sémantique de ce mot. Ainsi pomme' désigne la sémantique de pomme.

Ainsi, deux analyses syntaxiques différentes correspondent à deux analyses sémantiques identiques.

Mais, on peut également avoir l'analyse syntaxique suivante (émergence du sujet) :

```
Harry     mange    les pommes
NP        VP/NP    NP
--- < T
S/VP
---------------- < B
 S/NP
---------------------------- < f
          S
```

Ce qui sémantiquement donne le même résultat que précédemment, c'est-à-dire :

T Harry'
B (T (Harry') mange')
f(B (T (Harry') mange') les pommes')

L'application des règles de réduction de la sémantique nous donnent :

T (Harry') (mange' les pommes')
((mange' les pommes') Harry')

Nous voyons que pour une même phrase nous obtenons de nombreuses analyses syntaxiques mais que ces différentes analyses ne correspondent, en fait, qu'à une seule interprétation sémantique.

C'est ce phénomène que l'on appelle la *spurious ambiguity* ou la *pseudo-ambiguïté*. La pseudo-ambiguïté est donc le fait qu'il n'y a pas de relation systématique entre les structures syntaxiques et les représentations sémantiques. C'est un problème important que tentent actuellement de résoudre plusieurs chercheurs.

Lors de l'ACL à Manchester, et également lors d'autres congrès ainsi que dans de nombreuses publications, plusieurs solutions ont été proposées. Citons en rapidement quelques unes.

Winttenburg propose de restreindre la règle de "type raising". On pourrait, par exemple, l'appliquer uniquement dans le cas d'un NP-sujet. Il propose également de n'appliquer les règles de composition que dans des cas bien déterminés. Ceci reviendrait à établir une sorte de hiérarchie d'application des règles, ce qui ne semble pas être le cas pour le moment puisque la stratégie est d'appliquer toutes les règles en même temps[18].

Steedman et Pareschi, eux, proposent un algorithme d'analyse basé sur une analyse de graphe d'unification[19].

Heeple et Morrill en s'appuyant sur le théorème de Church-Rosser, proposent de construire une forme normale pour la sémantique. Ainsi ils montrent que toutes les analyses sémantiques sont réductibles à une seule : la forme normale[20].

---

[18] Cf. Winttenburg [27].

[19] Cf.Pareschi et Steedman [21].

[20] Cf. Heeple et Morrill [17].

Enfin, Bouma propose de définir une sous-grammaire qui soit encore plus flexible : il s'appuie sur le calcul de Lambek et il propose de rajouter aux grammaires catégorielles une règle d'introduction et des règles d'inférence[21].

Cependant, ces solutions sont encore inefficaces dans certains cas.

La coordination.

Nous pouvons dire que la "spurious ambiguïty" est le prix à payer pour avoir une méthode qui permette de traiter du problème épineux de la coordination, problème sur lequel butent de nombreuses théories grammaticales.

Pour terminer nous voudrions illustrer par un exemple comment de telles grammaires parviennent à traiter de la coordination.

Avant de passer à l'exemple à proprement parler, donnons les règles qui vont servir à traiter de la coordination. Précisons qu'au niveau syntaxique nous n'ajoutons pas de nouvelles règles si ce n'est de dire implicitement que l'on ne coordonne entre elles que des expressions de même type. La règle que l'on va poser pour la coordination est sémantique ; c'est grâce à elle que nous allons récupérer le sens de la phrase formée des phrases coordonnées.

Pour traiter de la coordination on utilise un autre combinateur, le combinateur φ.

La conjonction de coordination est interprétée à l'aide du combinateur φ tel que :

$$\varphi\ \&fgx \rightarrow \&\ (fx)\ (gx)$$

Exemple :

Considérons la phrase :

John aime Mary follement et Suzy sauvagement.

L'analyse syntaxique nous donne :

```
        John     aime     Mary     follement et      Suzy     sauvagement
        NP       VP/NP    NP       VP\VP             NP       VP\VP
                          --- <                      ----- < T
                          VP\VP/NP                            VP\VP/NP
                          ------------- < B          ------------- < B
                          VP\VP/NP                   VP\VP/NP
                 ------------------------------------------ < φ
                          VP\VP/NP
                 ---------------------------- < b
                          VP
--------------------------- < b
    S
```

[21] Cf. Bouma [5].

Sémantiquement on a :

T(Mary') T(Suzy')
B(T(Mary') follement') B(T(Suzy') sauvagement')
φ &<B T(Mary') follement'> B<T(Suzy') sauvagement'>
b(aime' (φ &<B T(Mary') follement'> B<T(Suzy') sauvagement'>))
b (John' (b aime'(φ &<B T(Mary') follement'> B<T(Suzy') sauvagement'>)))

En réduisant à l'aide des règles posées pour la sémantique on a :

b (John' (φ &<B T(Mary') follement'> B<T(Suzy') sauvagement'> aime'))
b (John' (&<B T(Mary') follement' aime'> B<T(Suzy') sauvagement' aime'>))
b (John' (&< T(Mary') (follement' aime')> <T(Suzy') (sauvagement' aime')>))
b (John' (& (follement' aime' Mary') (sauvagement' aime' Suzy')))
((& (follement' aime' Mary') (sauvagement' aime' Suzy')) John')

Nous aboutissons donc à la bonne interprétation sémantique, à savoir : *John aime Mary follement et John aime Suzy sauvagement.*

Signalons également les travaux de Mary Mc Gee Wood sur la coordination, qui proposent un algorithme permettant de traiter de la coordination en utilisant les règles que l'on a déjà vues précédemment mais dans des cas bien déterminés, évitant souvent ainsi le problème de la "spurious ambiguïty"[22]. Ainsi que les travaux de l'équipe de G. Bès. Dans le cadre des grammaires catégorielles cette équipe traite de problèmes linguistiques précis en français, et propose un modèle de grammaires catégorielles utilisant le mécanisme d'unification[23].

Nous espérons avoir réussi à donner une idée de ce que peut être le courant des grammaires catégorielles en Europe et invitons le lecteur intéressé à se reporter aux ouvrages cités en bibliographie.

BIBLIOGRAPHIE

[1] ACL (Association for computational Linguistic), 1989, *Proceedings of the fourth congres of the ACL european chapter*, Manchester.

[2] AJDUKIEWICZ K., 1978, *The scientific world-perspective and other essays, 1931-1933"*, Dordrecht, Riedel publishing company.

[3] BAR-HILLEL Y., 1968, "A quasi arithmetical notation for syntactic description", *Language* 29, pp.47-58, 1953, traduit in *Language* 9, pp. 9-22.

[4] BAR-HILLEL Y., 1964, *Language and information: selected essays on their theory and application*, Reading, Adison Westley.

[5] BASCHUNG K., BES G., CORLUY A., GUILLOTIN T., 1987, "Auxilary and clitics in French UCG Grammar", *Proceedings ACL 87 (european chapter)*, pp. 173-178, Copenhague.

[22] M. Mc Gee Wood [18],[19].

[23] Cf Bès et alii [5], [6].

[6] BES G., GARDENT C., 1989, "French order without order", *Proceedings ALC 89 (european chapter)*:ehp1., pp. 249-255, Manchester.

[7] BOUMA G., 1989, "Efficient processing of flexible categorial grammar" in *Proceedings ACL (European Chapter)*, pp.12-26.

[8] BUSZKOWSKI W., MARCISZEWSKI W. and VAN BENTHEM J., (ed.), 1988, *Categorial Grammar*, John Benjamins publishing company.

[9] CHANOD J.-P. et SEGOND F., 1988, "Grammaire applicative : traitement informatique de la composante morpho-syntaxique", *Mathématiques, informatique et sciences humaines* 103, pp.23-43.

[10] DESCLES J.-P., 1990, *Sciences cognitives et grammaire applicative universelle*, Paris, Hermès.

[11] DESCLES J.-P., GUENTCHEVA Z., SHAUMYAN S.-K., 1985, *Theoritical aspects of passivization in the framework of applicative grammar*, Amsterdam, John Benjamins publishing company.

[12] DOWTY D., 1985, "Type raising" *in Oehrle et alii*, 1988, pp. 153-197.

[13] GARDIES J.-L., 1975, *Esquisse d'une grammaire pure*, Paris, Vrin.

[14] GLADKIJ A.V., MEL'CUK I.A., 1972, *Eléments de linguistique mathématique*, Dunod.

[15] HEPPLE M. and MORRILL G., 1989, "Parsing and derivational equivalence" in *Proceedings ACL (European Chapter)*, pp.10-18.

[16] LAMBEK J., 1958, "The mathematics of Sentence Structure", *American Mathematical Monthly*, 65, pp. 154-165.

[17] Mc GEE WOOD M., 1987, "Paradigmatic rules for categorial grammars" in *CCL-UMIST Report 87-6*.

[18] Mc GEE WOOD M., 1986, "The description and processing of coordinate constructions" in *CCL-UMIST Report 86-4*.

[19] MILLER P.et TORRIS T., 1990, *Formalismes syntaxiques pour le traitement automatique du langage naturel*, Hermès, Paris.

[20] MOORGAT M., 1988, *Categorial Investigations : Logical and linguistic aspects of the Lambek calculus*, Groningen-Amsterdam Studies in Semantics, Foris publications.

[21] PARESCHI R. and STEEDMAN M., 1987, "A lazy way to chart-parse with categorial grammars" in *Proceedings ACL*, pp. 81-88.

[22] OEHRLE R.T., BACH E., WHEELER D., (ed.), 1988, *Categorial Grammars and Natural Languages Structures*, Reidel publishing company, Vol.32.

[23] SHAUMYAN S.-K., 1982, "The goals of linguistic theory and applicative grammar", *Mathématiques et sciences humaines* 77, pp.7-43.

[24] SEGOND F., 1989, Grammaire categorielle enrichie : une implémentation, *Actes du 7ème congrès AFCET-RFIA*, Paris, pp. 599-613.

[25] STEEDMAN M., 1985,"Dependency and coordination in the grammar of Dutch and English", *Language*, vol. 61-3, pp.523-568.

[26] STEEDMAN M., 1989, "Combinators, Grammars and processors for natural language", *Tutorial notes of the ACL (European chapter).*

[27] WITTENBURG K., 1987, "Predictive combinators : a method for efficient processing of combinatory categorial grammars", *Proceedings ACL*, pp. 73-80.