ICHARATE
Un Atelier Logique pour les Grammaires Multimodales

Houda ANOUN
Equipe Signes
LaBRI - Bordeaux1 - INRIA Futurs
anoun@labri.fr

Mots-clefs

analyse syntaxique, grammaires catégorielles, théorie des types, assistant de preuves

Résumé

Cet article présente le projet de l'atelier logique *ICHARATE* dédié à l'étude des grammaires catégorielles multimodales. Cet atelier se présente sous la forme de bibliothèques pour l'assistant de preuves **Coq**.

1 Vers les Grammaires Multimodales

1.1 Grammaires catégorielles et calcul de Lambek

Les grammaires catégorielles sont des grammaires universelles lexicalisées permettant d'analyser les phrases dans une variété très étendue de langues naturelles. Ces grammaires partagent toutes le même noyau de règles d'inférence et différent uniquement au niveau du lexique qui encapsule toutes les caractéristiques intrinsèques à la langue choisie. Le lexique n'est autre qu'une fonction qui associe à chaque mot de la langue considérée un ensemble fini de types atomiques ou composés qui décrivent son comportement syntaxique. Les types atomiques sont associés aux expressions complètes; citons par exemple **s** et **np** qui représentent respectivement une phrase et un syntagme nominal. Les types composés quant à eux sont obtenus à partir des types de base en utilisant les connecteurs **'/'**, **'\'** et **'•'**. Ainsi, le type A/B (resp. B\A) représente une entité qui attend un élément de type B à sa droite (resp. gauche) pour former une entité de type A, en revanche, le type A•B est associé à un élément obtenu en concaténant deux éléments de types respectifs A et B.
En 1958, Joachim Lambek (Lambek58) a proposé une formalisation logique des grammaires catégorielles en définissant un système déductif voisin de la logique linéaire (Girard89).
Les grammaires de Lambek admettent plusieurs variantes obtenues en ajoutant à l'ensemble

fixé des règles d'inférence de base un ensemble de règles structurelles qui permettent la gestion des ressources linguistiques dans le contexte. Le système sans règles structurelles est connu sous le nom de **NL**. A partir de ce système on obtient le système **L** (resp. **NLP**) en ajoutant la règle d'associativité (resp. de commutativité) de l'opérateur '$\bullet$'.
L'intérêt primordial du calcul de Lambek est qu'il entretient un lien très étroit avec la sémantique de **Montague** (Carpenter97) alors que c'est loin d'être le cas pour un grand nombre de modèles linguistiques tels les grammaires de dépendance et les TAG.

1.2 Limitations du calcul pur de Lambek

Les différentes variantes de la logique pure de Lambek sont certes définies de manière élégante et offrent plusieurs avantages dont la décidabilité de recherche de preuves, toutefois, ces dernières s'avèrent trop restreintes pour l'analyse des langues naturelles. Leur faiblesse émane de deux types de problèmes : le problème de sous-génération et celui de sur-génération.
Le premier problème est souvent rencontré car ces grammaires sont incapables d'analyser un grand nombre de phénomèmes de la langue naturelle tels l'ellipse ou les dépendances croisées. A titre d'exemple, le système relativement rigide **L** ne permet pas d'analyser l'extraction médiale à cause de l'absence totale d'une règle de commutativité, ainsi seule l'extraction périphérique[1] est analysable. Par conséquent, on peut facilement déduire dans cette logique que '*Houda vend*' et '*vend un livre*' sont respectivement de types $\mathbf{s}/\mathbf{np}$ et $\mathbf{np}\backslash\mathbf{s}$, néanmoins, on ne peut guère attribuer le type **s** à la phrase correcte '*Le pays que Houda aime le plus est le Maroc*'!
Le second problème, en revanche, reflète la flexibilité de certains systèmes qui reconnaissent à tort des phrases agrammaticales. Ceci est le cas du système **NLP** supportant la commutativité globale des ressources et qui est loin d'être adapté à l'analyse des langues naturelles où l'ordre des mots est généralement fort important dans la construction des phrases.
Plusieurs solutions ont été proposées pour restreindre les limitations du calcul pur de Lambek notamment par Glyn Morrill (Morrill94), Mark Hepple (Hepple90) et Michael Moortgat (Moortgat97), on s'intéressera dans ce qui va suivre à la dernière.

1.3 Introduction aux grammaires multimodales

Michael Moortgat a enrichi le système logique de Lambek en introduisant deux opérateurs unaires de contrôle $\Box$ et $\Diamond$ et en étiquetant chacun des connecteurs logiques par des modes de composition. On obtient alors une famille d'opérateurs ($/_i$, $\backslash_i$, $\bullet_i$, $\Box_j$, $\Diamond_j$) qui coexistent dans un système multimodal rassemblant les différentes logiques individuelles précédentes. Dans un tel système, les règles d'inférence sont partagées par tous les modes pour chacun des connecteurs logiques, en outre, on peut définir des règles structurelles susceptibles d'être appliquées localement en présence de modes spécifiques.
Les grammaires multimodales permettent de pallier les faiblesses des grammaires pures de Lambek. En effet, la présence d'une variété de modes de composition et règles structurelles associées dans le même système multimodal permet de résoudre le problème de sous-génération. En outre, la gestion locale et contrôlée des ressources permet d'éviter les problèmes de sur-génération. Ce contrôle se fait grâce à l'utilisation des modes de composition et des modalités $\Box$ et $\Diamond$[2]. On peut ainsi analyser le phénomène de l'extraction médiale en ayant recours à une

[1] Les éléments extraits du fragment à analyser sont situés soit à son extrême gauche soit à son extrême droite
[2] Ceci s'inspire fortement de l'utilisation des exponentiels '!' et '?' dans la logique linéaire (Girard89)

associativié restreinte accessible uniquement en présence du mode **a** et une commutativité locale accessible en présence de formules décorées par $\diamond_c$. Un tel phénomène linguistique est par exemple présent dans la phrase nominale [3] suivante: (اَلعَالِمُ اَلَّذِي تُحِبُّ هُدَى كَثِيرًا ذَكِيٌّ). [4]

هُدَى,اَلعَالِمُ	اَلَّذِي	تُحِبُّ	كَثِيرًا	ذَكِيٌّ
np	$(\diamond_c \Box_c \text{np} \backslash_a \text{s}) \backslash_a (\text{np} /_a \text{np})$	$\text{np} \backslash_a (\text{np} \backslash_a \text{s})$	$\text{s} /_a \text{s}$	$\text{s} /_a \text{np}$

Figure 1: Lexique

L'analyse de cette phrase se fait sans beaucoup de peine. En effet, en considérant le lexique de la figure 1, on peut facilement attribuer le type $\mathbf{np}/_a\mathbf{np}$ à la subordonnée relative (اَلَّذِي تُحِبُّ هُدَى كَثِيرًا) car l'élément extrait (l'objet du verbe تُحِبُّ) a accès à la commutativité puisqu'il est décoré par le connecteur $\diamond_c$.

Outre les avantages déjà mentionnés, la logique multimodale a une capacité d'inférence plus grande que celle de l'union de tous les systèmes simples pris individuellement : en effet, cela provient de la présence de règles structurelles d'interaction qui assurent la communication entre différents modes de composition. Ces dernières règles augmentent le pouvoir expressif des grammaires multimodales et leur garantissent l'analyse de plusieurs phénomènes compliqués présents dans différentes langues naturelles tels l'ellipse (ex: '*Mathematic is the queen of sciences and arithmetic the queen of Mathematics*') et le regroupement de verbes en néerlandais (verb cluster) (ex: '*Mary wil plagen*') ('veut taquiner Mary') etc...

1.4 Outils pour les grammaires multimodales

La manipulation des grammaires multimodales s'avère en général une tâche compliquée notamment en présence de plusieurs modes de composition et diverses règles structurelles d'interaction ou d'inclusion. Plusieurs outils ont été réalisés pour faciliter cette tâche et automatiser l'analyse syntaxique des phrases pour des lexiques donnés, entre autres l'outil Grail implémenté en SICStus Prolog par Richard Moot. Cet analyseur est basé sur les réseaux de preuves qui représentent les dérivations d'une manière élégante sous forme de graphes (Moot02).
Certes, Grail propose maintes fonctionnalités mais il ne permet d'effectuer des preuves qu'une fois que les paramètres de la grammaire (le lexique et les règles structurelles) sont choisis. Il s'avèrerait utile d'avoir un outil qui permet en plus d'établir des propriétés génériques et réutilisables satisfaites par différentes classes de grammaires voire même par toutes les grammaires. L'atelier $\mathcal{ICHARATE}$ se veut une réponse à ce besoin.

2 Atelier $\mathcal{ICHARATE}$

Je présente dans ce qui va suivre une description de l'atelier $\mathcal{ICHARATE}$ que je suis en train de réaliser en Coq avec l'aide de Pierre Castéran, et ce dans le cadre de ma thèse au LaBRI.

[3] Phrase composée d'un sujet et d'un attribut tandis que le verbe 'être' est implicite

[4] Cet exemple est pris de la langue arabe classique, sa traduction -mot à mot- est : 'intelligent énormement Houda aime que le savant', sa traduction en français est : 'Le savant que Houda aime énormement est intelligent'.

2.1 Noyau d'$\mathcal{ICHARATE}$

Le noyau de l'atelier contient toutes les définitions formelles et spécifications nécessaires pour la formalisation des grammaires multimodales.
Ces dernières sont formalisées de trois manières différentes: sous forme axiomatique, sous forme de calcul des séquents et sous forme de déduction naturelle. Ces trois systèmes logiques proposent des avantages complémentaires. En effet, le premier formalisme garantit une preuve de cohérence et de complétude relativement aisée. Le deuxième quant à lui, assure la décidabilité de recherche de preuves. En revanche, le troisième est le plus intuitif et le plus adapté à l'analyse sémantique.
Coq s'avère bien adapté à la formalisation des différentes présentations des grammaires multimodales. En effet, cet assistant de preuves est basé sur le calcul des constructions inductives (CCI) qui est une extension de la logique d'ordre supérieur (CoqManual). Ceci permet de représenter les dérivations syntaxiques des différents formalismes précédents comme des structures de données en définissant les opérateurs de déduction sous forme de types inductifs.
Outre les différents modules dédiés à l'analyse syntaxique, le noyau comprend un module de calcul de la sémantique basé sur un λ-calcul simplement typé. Ce module permet de calculer le λ-terme associé à la sémantique d'une phrase en fonction des λ-termes associés à chacun des mots qui la composent et ce à partir de son arbre de dérivation syntaxique.

2.2 Bibliothèque d'$\mathcal{ICHARATE}$

$\mathcal{ICHARATE}$ comprend une bibliothèque qui contient les preuves de plusieurs propriétés de la logique multimodale. Ci dessous, je cite quelques unes en précisant leur intérêt:

2.2.1 Propriétés dérivées

Plusieurs propriétés dérivées génériques sont prouvées dans chacun des formalismes précédents. Ces dernières sont réutilisables dans n'importe quelle dérivation où elles peuvent être appliquées comme les règles d'inférence de base du système considéré. On distingue deux catégories principales de ces propriétés:

- Propriétés totalement génériques: Ces propriétés sont vérifiées par toute grammaire multimodale. Elles peuvent ainsi être appliquées quelque soient le lexique et les règles structurelles choisis. Parmi ces dernières citons la propriété de montée de type (type raising) ($\forall$ A B i : A $\Rightarrow$ B$/_i$(A$\backslash_i$B)) aussi bien que celle d'isotonicité ($\forall$ A B C i: si A $\Rightarrow$ B alors A$/_i$C $\Rightarrow$ B$/_i$C). Ces deux propriétés sont prouvées une seule fois et peuvent être utilisées autant de fois qu'on souhaite. On peut ainsi avoir recours à la propriété de montée de type pour effectuer la dérivation de la phrase '*Saida and I watch TV*'. En effet, les deux entités reliées par la conjonction de coordination 'and' portent deux types syntaxiques distincts : **s**$/_n$**(np** $\backslash_n$ **s)** pour le pronom sujet '*I*' et **np** pour le nom propre '*Saida*'. Toutefois, grâce à la propriété de montée de type, on peut facilement déduire que **np** $\Rightarrow$ **s**$/_n$**(np** $\backslash_n$ **s)** et attribuer ainsi le type **s** à la phrase correcte précédente[5].

- Propriétés conditionnées : Ces propriétés sont démontrées pour des classes particulières de grammaires qui vérifient certaines contraintes.

[5]Dans ce cas on attribue à la conjonction de coordination '*and*' le type (X $\backslash_n$X)$/_n$X; avec X=**s**$/_n$**(np** $\backslash_n$ **s)**

Ainsi, la propriété de composition ($\forall$ A B C i j: A $/_i$B $\bullet_i$ B$/_j$C $\Rightarrow$ A $/_j$C) peut être appliquée si la grammaire en question supporte la règle d'interaction d'associativité mixte MA(i, j) ($\forall$ A B C : (A $\bullet_i$ B) $\bullet_j$ C $\rightarrow$ A $\bullet_i$ (B $\bullet_j$ C)). D'autre part, on a prouvé que la règle de coupure ($\forall$ Δ Γ A C: si $\Delta \vdash$ A et Γ[A] $\vdash$ C alors $\Gamma[\Delta] \vdash$ C) est dérivable dans le système logique de déduction naturelle multimodale à condition que les règles structurelles supportées vérifient la condition de Sahlqvist (Moortgat97).[6] La règle de coupure est fort utile car elle permet très souvent de faciliter les preuves en ayant recours à l'application de lemmes intermédiaires. Par conséquent, on peut appliquer cette propriété pour effectuer une dérivation de la phrase *'Le coeur a ses raisons que la raison ignore'* en utilisant une dérivation intermédiaire permettant d'attribuer le type **n**$\backslash$**n** (modifieur de nom commun) à la subordonnée relative *'que la raison ignore'*.

2.2.2 Intéropérabilité entre les trois formalismes multimodaux

Les trois formalismes de la logique multimodale admettent des règles d'inférence différentes. Toutefois, ils permettent d'engendrer le même langage. La preuve de l'intéropérabilité entre ces trois formalismes est effectuée grâce à des fonctions certifiées qui permettent de transformer la dérivation d'un formalisme source à un formalisme cible supportant les règles structurelles du premier. La preuve étant constructive, il est possible d'exporter divers théorèmes prouvés dans un formalisme et les utiliser dans d'autres formalismes sans les redémontrer. A titre d'exemple, citons le théorème d'isotonicité (cf. 2.2.1) qui est facilement démontrable en calcul des séquents alors qu'il est relativement ardu à prouver en déduction naturelle. Grâce à l'intéropérabilité entre ces deux derniers formalismes, on obtient sans effort une preuve de ce théorème en déduction naturelle et ce à partir de sa preuve en calcul des séquents.

2.2.3 Divers méta-théorèmes

La bibliothèque contient aussi les preuves de plusieurs méta-théorèmes. Ainsi, on a prouvé en utilisant les modèles de **Kripke** (Anoun03) que les systèmes logiques formalisés sont à la fois cohérents et complets. D'autre part, on a démontré quelques méta-théorèmes permettant de définir des conditions suffisantes pour la non dérivabilité d'un séquent ($\Gamma \Rightarrow$ A) dans un formalisme donné. Citons par exemple le méta-théorème de la polarité qui permet d'effectuer des preuves négatives par un simple calcul arithmétique en ayant recours à la technique de raisonnement par reflexion (CoqArt04). En effet, si la polarité d'un atome p [7] dans 'Γ' est différente de celle dans ' A', ce méta-théorème permet de déduire que le séquent n'est pas démontrable. Grâce à ce méta-théorème, on déduit une preuve formelle de la non dérivabilité du séquent **(np , (np**$\backslash_i$**s)**$/_i$ **np**$)^i \Rightarrow$ **s**, ce qui correspond linguistiquement à l'agrammaticalité d'une phrase composée d'un sujet et d'un verbe transitif (ex: '*Houda loves*').

2.3 Résultats & Perspectives

L'atelier $\mathcal{ICHARATE}$ que j'ai présenté dans cet article est un projet en cours de réalisation. Il constitue un outil pédagogique adressé aux gens désirant explorer la logique multimodale qui

[6] Toutes les règles structurelles utilisées par Moortgat dans (Moortgat97) vérifient cette condition

[7] Valeur algébrique représentant la différence entre le nombre d'occurrences positives et le nombre d'occurences négatives d'un atome dans une formule ou un contexte de formules (Rétoré03).

est certes loin d'être simple à appréhender. L'utilisation du mode interactif de preuves de Coq pourra certainement faciliter cette tâche à priori ardue.
$\mathcal{ICHARATE}$ comprend un noyau cohérent contenant différentes formalisations de la logique multimodale, les preuves de plusieurs propriétés réutilisables et génériques portant sur ces formalismes aussi bien qu'un ensemble de tactiques élémentaires et un module de calcul de la sémantique.
La prochaine piste envisageable est d'étendre la sémantique de Montague qui est basée sur le λ-calcul simplement typé pour englober aussi des types plus compliqués (inductifs ou avec produits dépendants) et tirer ainsi profit de la richesse du calcul des constructions inductives de Coq. D'autre part, la réalisation d'une interface graphique est prévue, cette dernière facilitera l'interaction entre $\mathcal{ICHARATE}$ et les utilisateurs non spécialistes de Coq .
On envisage aussi de faire communiquer l'Atelier avec d'autres outils existants réalisés par les membres de l'équipe **Signes** (Signes04) et tirer ainsi profit de leurs fonctionnalités. Ceci est possible car Coq est fondé sur la logique intuitionniste et l'isomorphisme de Curry-Howard, il permet ainsi d'extraire des fonctions certifiées conformes à leurs spécifications à partir de preuves constructives et ce dans un langage fonctionnel tel Ocaml.
Dans une autre direction, la formalisation des grammaires minimalistes est envisagée ainsi que l'implantation de fonctions certifiées traduisant les dérivations de ces dernières grammaires dans les grammaires multimodales et réciproquement (Vermaat99).

Références

[Anoun03] Houda Anoun. *Réalisation d'un atelier logique sur le calcul de Lambek* Mémoire de dea, Université Bordeaux 1 et ENSEIRB, 2003. http://www.labri.fr/Recherche/LLA/signes.

[CoqArt04] Yves Bertot et Pierre Castéran. *Interactive Theorem Proving and Program Development: Coq'Art: The Calculus of Inductive Constructions*. Springer Verlag.

[Carpenter97] Bob Carpenter. *Type Logical Semantics*. Massachusetts Institute of Technology, 1997.

[CoqManual] The Coq Development Team. *The Coq Proof Assistant, Reference Manual, Version 8.0* January 2004

[Girard89] Jean-Yves Girard, Paul Taylor and Yves Lafont. *Proofs and Types*. Cambridge Tracts in Theoretical Computer sciences 7, Cambridge, 1989.

[Hepple90] Mark Hepple *The Grammar and Processing of Order and Dependency: a Categorial Approach*. PhD thesis, Edinbergh, 1990.

[Lambek58] Joachim Lambek *The mathematics of sentence structure* 1958

[Morrill94] Glyn Morrill *Type Logical Grammar. Categorial Logic of Signs* Kluwer,Dordrecht, 1994

[Moortgat97] Michael Moortgat *Categorial Type Logic* Chapter 2 in Van Benthem & ter Meulen (eds) Handbook of Logic and Language, 1997

[Moot02] Richard Moot. *Proof nets for linguistic analysis*. PhD thesis, UIL-OTS, Universiteit Utrecht, 2002.

[Rétoré03] Christian Rétoré *The logic of categorial grammars*. 2003

[Signes04] Porposition du projet Signes par l'INRIA *Signes linguistiques, grammaire et sens: algorithmique logique de la langue* http://www.labri.fr/Recherche/LLA/signes/resources/docs/signes.pdf.

[Vermaat99] Willemijn Vermaat *Controlling Movement:Minimalism in a deductive perspective*. Thesis, Universiteit Utrecht, April 1999