L'interface entre syntaxe et sémantique pour les grammaires minimalistes catégorielles

Maxime Amblard, Alain Lecomte, Christian Retoré

L'interface entre syntaxe et sémantique pour les grammaires minimalistes catégorielles

Maxime Amblard, Alain Lecomte, Christian Retoré LaBRI, UMR 5800

Les propositions [2] de systèmes déductifs décrivant les grammaires minimalistes d'Edward Stabler [3] permettent d'étudier l'interface syntaxe sémantique [1]. L'idée principale est que la forme logique n'est pas le résultat immédiat de la dérivation syntaxique mais qu'il existe une analyse sémantique construite en parallèle. Les deux analyses participent à la bonne formation de la phrases, en ceci que les dérivations correctes sont celles dont les deux pendants, syntaxique et sémantique, sont synchronisés. Nous montrons ici comment l'utilisation de contextes permet cette synchronisation.

Grammaires Minimalistes Catégorielles La fusion (merge) correspond aux règles classiques d'élimination de / et \ des grammaires catégorielles.

Le déplacement (move), correspond à l'élimination du produit noté $\times$.

Les règles du produit sont les suivantes,

$$\frac{\Gamma \vdash x : A \quad \Delta \vdash y : B}{\Gamma, \Delta \vdash (x,y) : A \times B}[i\times] \qquad \frac{\Gamma \vdash w : A \times B \quad \Delta, x : A, y : B, \Delta' \vdash z : C}{\Delta, \Gamma, \Delta' \vdash let(x,y) = (\pi_1(w), \pi_2(w)) inz : C}[e\times]$$

π_1 et π_2 sont les projections des deux composantes

la contruction *let u = v in w* de la programmation fonctionnelle

L'élimination du produit peut intervenir à tout moment dans la dérivation, dès que les hypothèses à décharger ont été introduites. En pratique, nous utilisons une règle avec trois prémisses, règles qui s'obtient à partir de celles ci-dessus :

$$\frac{\Gamma \vdash w : A \times B \quad x : A \vdash x : A \quad y : B, \Delta \vdash y : B}{\Gamma, \Delta \vdash let(x,y) = (\pi_1(w), \pi_2(w)) inz : C}[e\times]^3$$

Nous ne prenons pas en compte les déplacements de tête, qui peuvent être simulés par des déplacements de constituants [3].

Calcul de la sémantique et λ-calcul Nous utilisons le λ-calcul pour obtenir une formule logique d'ordre supérieur représentant le sens de nos énoncés.

Pour sela nous utilisons les deux règles suivantes représentant des applications de λ-termes en fonction du sens de l'application :

$$\frac{\Delta \vdash z : T \to U \to V \quad \Gamma \cup [x : T] \vdash \gamma : U}{\Delta \cup \Gamma \vdash z(\lambda x.\gamma) : V}[RAISE] \frac{\Delta \vdash z : T \quad \Gamma \cup [x : T] \vdash \gamma : U}{\Delta \cup \Gamma \vdash (\lambda x.\gamma)(z) : U}[NORAISE]$$

λ-termes contextués La principale difficulté rencontrée pour le calcul de la sémantique est de pouvoir automatiquement procéder aux applications correctes et de pouvoir abstraire certaines variables bien qu'elles aient précédemment été consommées.

En effet, dans une dérivation standard, le verbe consomme d'abord deux variables qui déchargent les positions des "DP", puis dans la dérivation, on introduit le véritable objet ou sujet, après un déplacement. Pour cela, nous ne pouvons pas nous contenter d'utiliser de simple λ-termes et nous introduisons alors un contexte.

Un λ*-terme contextué* est un λ-terme dont une partie des variables libres sont déclarées et placées à gauche du symbole $\vdash$. Ces dernières ne sont pas accessibles pour le λ-calcul et n'apparaissent pas dans le type sémantique du λ-terme. Les abstractions doivent abstraires l'une des variables du contexte, et l'application concatène les variables du contexte. Les termes doivent etre clos et le contexte vide en fin de dérivation.

L'utilisation de contextes permet de faire la différence le corps d'un terme et ce même terme avec une abstraction Par exemple on distingue $u : U, v : V \vdash u : U$ de $\lambda u \lambda v.u : U \to V \to U$. Le premier peut devenir $v \vdash \lambda u.u : U \to U$ ou $u \vdash \lambda v.u : V \to U$ et les deux termes en partie droite, qui n'ont pas le même type, ne s'appliquent pas aux mêmes termes.

Pour amorcer la dérivation, on utilise simplement des axiomes de la forme $u : \tau \vdash u$ si τ est le type de u.

Synchronisation Syntaxe Sémantique Nous appellerons SYN le calcul syntaxique comprenant $\times$, / et $\backslash$ d'où les règles $[e/]$, $[e\backslash]$, $[e\times]$ et $[HM]$. De manière analogue, nous appellerons SEM le calcul sémantique avec seulement $\to$ et les règles $[e \to]$, $[RAISE]$ et $[NORAISE]$.

Chaque étape dans la dérivation syntaxique, soit dans SYN, possède sa contrepartie dans SEM et permet de construire simultanément une dérivation sémantique :

— l'étape $[e\times]$ avec Γ vide est [RAISE] et avec Γ non vide est [NORAISE].

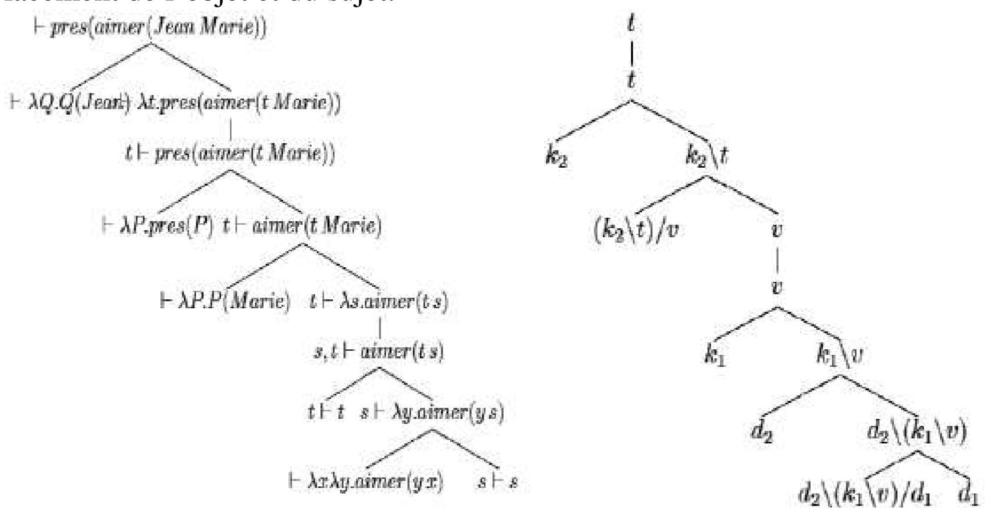
— l'étape $[e/]$ et $[e\backslash]$ sont une étape $[e \to]$.

Ainsi, les deux dérivations sont dites synchronisées ssi :

— chaque feuille de SYN a sa contrepartie dans SEM.

— chaque étape est réalisée dans les deux dérivations dans le même ordre.

On obtient alors l'interface syntaxe sémantique suivante, où on utilise les contextes des λ-termes lors du déplacement pour abstraire sur la bonne variable, comme on peut le voir sur le déplacement de l'objet et du sujet.



Perspectives Le système décrit ici permet d'envisager une véritable interaction entre syntaxe et sémantique pour les grammaires minimalistes, tout en restant aisément calculable. Nous envisageons de programmer une variante de ce modèle.

Références

[1] Maxime Amblard. Représentations sémantiques pour les grammaires minimalistes. Master's thesis, Université Bordeaux 1, 2003.

[2] Alain Lecomte and Christian Retoré. Extending lambek grammars : a logical account of minimalist grammars. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics, ACL*, pages 354–361, Toulouse, July 2001.

[3] Edward Stabler. Remnant movement and structural complexity. In *Constraints and Resources in Natural Language Syntax and Semantics*, pages 299–326. Gosse Bouma, Erhard Hinrichs, Geert- Jan M. Kruijff, and Richard Oehrle, distributed by cambridge university press edition, 1999.