

# ACGTK: un outil de développement et de test pour les grammaires catégorielles abstraites

Sylvain Pogodalla

INRIA, Villers-lès-Nancy, F-54600, France

Université de Lorraine, LORIA, UMR 7503, Vandœuvre-lès-Nancy, F-54500, France

CNRS, LORIA, UMR 7503, Vandœuvre-lès-Nancy, F-54500, France

[sylvain.pogodalla@inria.fr](mailto:sylvain.pogodalla@inria.fr)

## RÉSUMÉ

---

Nous présentons un outil, ACGTK, offrant un environnement de développement et d'utilisation des grammaires catégorielles abstraites pour l'analyse et la génération.

## ABSTRACT

---

### ACGTK : a Toolkit to Develop and Test Abstract Categorical Grammars

We present a toolkit, ACGTK, that offers an environment to develop and test abstract categorical grammars for parsing and generation.

**MOTS-CLÉS :** Grammaires catégorielles abstraites, syntaxe, sémantique, analyse, génération, ingénierie grammaticale .

**KEYWORDS:** Abstract categorical grammars, syntax, semantics, parsing, generation, grammatical engineering.

---

Les grammaires catégorielles abstraites (ACG, de Groote (2001)) appartiennent à la famille des grammaires de types logiques. Elles offrent un cadre grammatical dans lequel différents formalismes grammaticaux peuvent être encodés (de Groote & Pogodalla, 2004). Elles engendrent des langages de  $\lambda$ -termes, qui généralisent les chaînes de caractères et les arbres, et permettent la représentation de formules logiques pour la sémantique. Une caractéristique est de considérer explicitement les structures d'analyse des grammaires, le *langage abstrait*. Ces structures sont ensuite interprétées à l'aide d'un *lexique* pour obtenir le *langage objet* des formes de surfaces (chaînes de caractères, arbres dérivés, formules logiques, etc.). On appelle *analyse ACG* l'opération qui permet de retrouver les structures (termes) abstraites à partir d'un terme objet. Cette opération consiste à trouver le ou les antécédents de ce dernier par le lexique. La classe des ACG dites *de second ordre*<sup>1</sup> permet des analyses polynomiales dont les bornes de complexité correspondent aux meilleures connues (Kanazawa, 2008). Cette propriété ne dépendant que du vocabulaire abstrait, l'analyse ACG peut se faire aussi bien à partir d'une chaîne de caractères, d'un arbre, que d'une formule logique. Les deux premiers cas correspondent à la tâche d'analyse syntaxique, et le dernier à la réalisation syntaxique.

ACGTK<sup>2</sup> est un outil qui met en œuvre les ACG. Un compilateur, `acgc`, contrôle la bonne formation des signatures et des lexiques, et un interpréteur, `acg`, fournit un langage de commandes, notamment pour l'analyse et l'interprétation de termes.

---

1. Cette classe engendre les langages faiblement contextuels (TAG, well-nested MCFG, LCFRS, etc.).

2. Disponible à l'adresse <http://www.loria.fr/equipes/calligramme/acg/#Software>.

Nous montrerons l'encodage et l'utilisation de différentes grammaires, notamment une grammaire TAG<sup>3</sup> avec sa sémantique, pour l'analyse d'expressions et la réalisation syntaxique de formules logiques. Les Commandes ACG 1 montrent l'analyse de la chaîne de caractère *every + dog + chases + a + cat* (recherche de l'antécédent par tag\_yields), ainsi que l'analyse de la formule logique  $\forall x. \text{dog } x \Rightarrow \exists x'. \text{cat } x' \wedge \text{chase } x x'$  (recherche de l'antécédent par tag\_semantics). La réalisation du terme antécédent est illustrée Fig. 1.

```
# tag_yields parse every + dog + chases + a + cat:S;
tag_yields parse every + dog + chases + a + cat:S;
An antecedent by tag_yields in derivation_trees is:
C_chases I_s I_vp (C_dog C_every I_n) (C_cat C_a I_n) : S
# tag_semantics parse All x. (dog x) > (Ex x'. (cat x') & (chase x x')) :S;
tag_semantics parse All x. (dog x) > (Ex x'. (cat x') & (chase x x')) :S;
An antecedent by tag_semantics in derivation_trees is:
C_chases I_s I_vp (C_dog C_every I_n) (C_cat C_a I_n) : S
```

### COMMANDES ACG 1 – Exemple d'utilisation d'une ACG définissant encodant une TAG

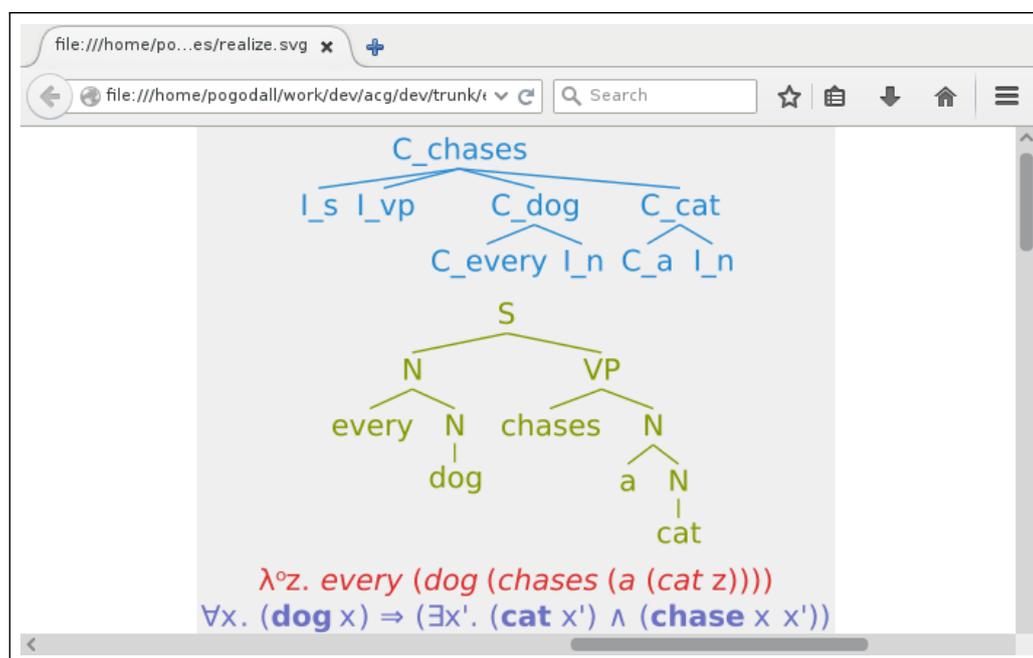


FIGURE 1 – Réalisation d'un terme abstrait en un arbre syntaxique (dérivé), une chaîne de caractères et une formule logique

## Références

- DE GROOTE P. (2001). Towards Abstract Categorical Grammars. In *Association for Computational Linguistics, 39th Annual Meeting and 10th Conference of the European Chapter, Proceedings of the Conference*, p. 148–155. ACL anthology: P01-1033.
- DE GROOTE P. & POGODALLA S. (2004). On the expressive power of Abstract Categorical Grammars : Representing context-free formalisms. *Journal of Logic, Language and Information*, 13(4), 421–438. HAL open archive: inria-00112956. DOI: 10.1007/s10849-004-2114-x.
- KANAZAWA M. (2008). A prefix-correct earley recognizer for multiple context-free grammars. In *Proceedings of the Ninth International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+9)*, p. 49–56, Tuebingen, Germany. <http://tagplus9.cs.sfu.ca/papers/Kanazawa.pdf>.

3. Un exemple est disponible à <http://www.loria.fr/equipes/calligramme/acg/software/examples/tag.acg>.