Une métagrammaire de l'interface morpho-sémantique dans les verbes en arabe

Simon Petitjean, Younes Samih, Timm Lichte
Heinrich-Heine-Universität Düsseldorf, Allemagne
simon.petitjean@hhu.de, samih@phil.hhu.de, lichte@phil.hhu.de

Résumé. Dans cet article, nous présentons une modélisation de la morphologie dérivationnelle de l'arabe utilisant le cadre métagrammatical offert par XMG. Nous démontrons que l'utilisation de racines et patrons abstraits comme morphèmes atomiques sous-spécifiés offre une manière élégante de traiter l'interaction entre morphologie et sémantique.

Abstract.

A metagrammar of the morphology-semantics interface in Arabic verbs
In this article we propose to model the derivational morphology of Arabic using the metagrammatical framework of XMG. We demonstrate that treating abstract roots and patterns as semantically underspecified atomic morphemes offers an elegant way to account for the interaction between morphology and semantics.

Mots-clés : Morphologie, arabe, metagrammaire, frame semantics.

Keywords: Morphology, Arabic, metagrammar, frame semantics.

1 Introduction

Dans cet article, nous présentons une implémentation de la morphologie arabe utilisant le formalisme XMG (pour eXtensible MetaGrammar). Dans le même temps, nous proposons et décrivons des stratégies générales pour exploiter les capacités des mécanismes de résolution de contraintes pour représenter les propriétés de la morphologie verbale de type 'racine-et-patron' de l'arabe. Bien que nous privilégions ici une approche basée sur le concept de métagrammaire, nous reconnaissons que les approches basées sur les automates finis pour l'analyse et la génération des langues sémitiques, comme celles de (Beesley, 1998) et (Yona & Wintner, 2005), (Shaalan *et al.*, 2012), (Habash & Rambow, 2006), (Kiraz, 2001), ont été fructueuses. Cependant, nous pensons que notre approche a un important avantage sur les méthodes utilisant les automates finis : de par sa modularité, elle permet d'exprimer de manière simple l'interface morphologie-sémantique, ou plus brièvement le problème de l'interface. Dans les automates finis, le travail du transducteur est de traduire des analyses en formes de surface et inversement, sans permettre de façon directe de fournir de l'information sémantique indiquant quel segment contribue à quelles parties ou comment. De nombreux linguistes ont tendance à trouver cette méthode de description lourde, en particulier s'ils considèrent généralement que les systèmes morphologique et sémantique sont interconnectés. Bien que les morphologies non concaténatives constituent probablement le meilleur cas d'étude pour passer des méthodes basées sur les automates finis aux métagrammaires pour traiter l'interface morphologie-sémantique, nous montrerons que cette approche n'est pas sans mérite dans le cas des morphologies hautement concaténatives.

Dans la section 2, nous donnons les principes de la morphologie verbale en arabe. Puis, dans la section 3, nous présentons le cadre de développement que nous utilisons pour sa description. La section 4 donne les détails de la métagrammaire développée pour générer le lexique de formes verbales. Dans la section 5, nous comparons notre approche avec des travaux similaires. Enfin, la dernière section présente la conclusion de cet article et annonce les étapes suivantes de ces travaux.

2 La morphologie verbale en arabe

L'interprétation linguistique standard du processus de formation des mots dans les langages sémitiques décrit les mots comme la combinaison de deux morphèmes : une racine et un patron (parfois appelé schème), pour lesquels le premier effort de génération formelle a été réalisé par (McCarthy, 1981). Les racines sont généralement composées de trois consonnes, parfois quatre, et leur nombre est estimé à 7502, dont 2903 sont fréquemment utilisées (Altabbaa *et al.*, 2010). Un patron peut se présenter comme une séquence de lettres [1], définissant les positions des voyelles relativement aux consonnes de la racine.

Par exemple, les verbes *ma$aY*,'marcha', *ma$~Y*,'fit marcher', partagent tous le même morphème racine *m$y*, 'lié à la marche'.

(1) ma$~Y Al>abu Alwalada
marcher.CAUSE-PAST le_père le_fils
'Le père fit marcher l'enfant.'

(2) ma$aY Al>abu
marcher.SIMPLE-PAST le_père
'Le père marcha.'

La deuxième et la troisième colonne du tableau 1 présentent 9 patrons compatibles avec les racines verbales composées de trois consonnes, à l'actif et au passif (il faut également noter que toutes les racines ne sont pas compatibles avec tous les patrons). C_1, C_2 et C_3 représentent les trois consonnes de la racine.

Patron	Actif	Passif	Sémantique
1	$C_1aC_2aC_3$	$C_1uC_2iC_3$	
2	$C_1aC_2C_2aC_3$	$C_1uC_2C_2iC_3$	Causatif du transitif 1
3	$C_1aaC_2aC_3$	$C_1uuC_2iC_3$	Associatif
4	ʕaC$_1$C$_2$aC$_3$	ʕuC$_1$C$_2$iC$_3$	Causatif de 1
5	$taC_1aC_2C_2aC_3$	$tuC_1uC_2C_2ib$	Reflexif de 2 (médiopassif)
6	$taC_1aaC_2aC_3$	$tuC_1uuC_2iC_3$	Reciproque de 3
7	$nC_1aC_2aC_3$	$nC_1uC_2iC_3$	Reflexif / resultatif / passif / médiopassif
8	$C_1taC_2aC_3$	$C_1tuC_2iC_3$	Reflexif / médiopassif
10	$staC_1C_2aC_3$	$stuC_1C_2iC_3$	Requestatif

TABLE 1 – Patrons pour les racines composées de trois consonnes, et sémantiques des patrons verbaux proposées par (Ryding, 2005), extraites de (Danks, 2011)

Nous supposons, comme Doron (2003, 2013); Schneider (2010) parmi d'autres, qu'au moins une partie des patrons est associée à une contribution sémantique sous spécifiée (ces contributions sont présentées dans le tableau 1). Sous cette hypothèse, la combinaison d'une racine et d'un patron mène à la composition de leurs sémantiques. Des incompatibilités entre racines et patrons peuvent en conséquence être motivées par l'incompatibilité de leurs sémantiques respectives. Dans ce travail, nous implémentons cette idée en utilisant des représentations basées sur la théorie des *frames*, dans la tradition de Fillmore (1977) et Barsalou (1992). Dans la mesure où nous les traitons comme des structures de traits typées étendues (Petersen, 2007; Kallmeyer & Osswald, 2013; Lichte & Petitjean, to appear), la composition est vue comme l'unification. Dans la figure 1 nous présentons quelques frames préliminaires pour le deuxième patron, qui introduisent la causalité, et pour la racine *m$y*. Les frames de type *causation* et *locomotion* sont empruntées à (Kallmeyer & Osswald, 2013).

Notons que dans la figure 1 la frame de la racine est unifiée avec la valeur du trait EFFECT de la frame du patron (toutes les deux étiquetées IN). L'unification est donc parfois effectuée entre des sous parties des frames. L'unification des types est déterminée par une hiérarchie de types comme celle de la figure 2. C'est pourquoi l'unification des types *activity* et *locomotion* dans la figure 1 produits *locomotion*, qui est leur sous-type commun le plus spécifique.

3 eXtensible MetaGrammar

eXtensible MetaGrammar, ou XMG (Crabbé *et al.*, 2013) [2], désigne à la fois un formalisme métagrammatical et l'outil utilisé pour traiter les descriptions reposant sur ce formalisme. L'outil en question, appelé compilateur, permet de générer une ressource linguistique à partir d'une description abstraite et plus compacte de celle-ci (la métagrammaire). Les

1. Nous utilisons dans cet article la méthode de translittération de Buckwalter (Buckwalter, 2004) : `www.qamus.org/transliteration.htm`.
2. La nouvelle implémentation du compilateur, XMG-2, est disponible librement à l'adresse suivante : `https://sourcesup.cru.fr/xmg`

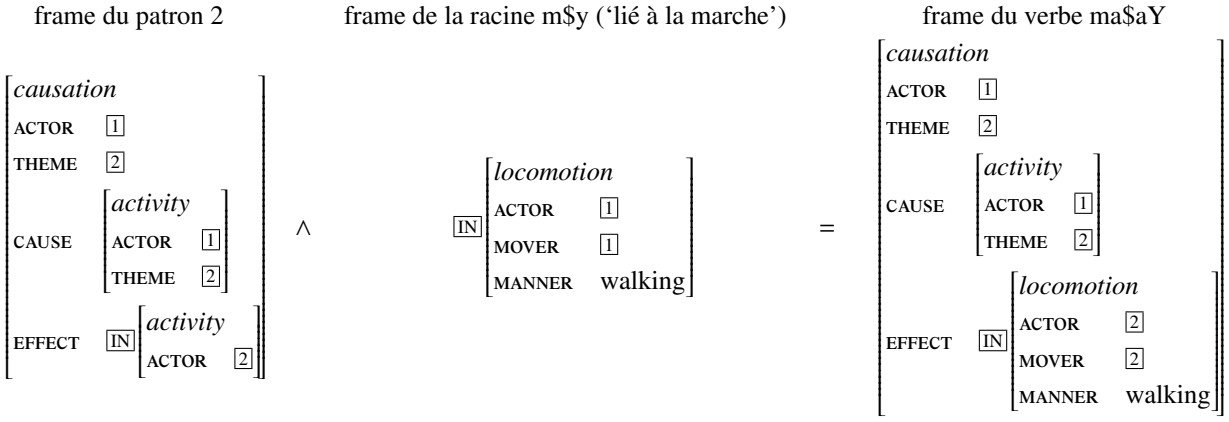

Figure 1 – Représentations sémantiques et composition du patron 2 et de la racine m$y (‘lié à la marche’)

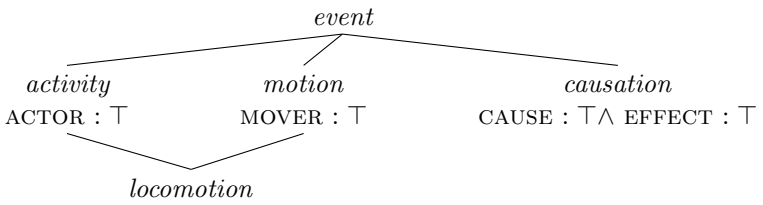

Figure 2 – Hiérarchie de types avec contraintes de type extraite de Kallmeyer & Osswald (2013)

descriptions métagrammaticales utilisant ce formalisme sont totalement déclaratives, et peuvent être formalisées de la manière suivante :

$$\begin{aligned} \mathit{Classe} &:= \mathit{Nom} \rightarrow \mathit{Contenu} \\ \mathit{Contenu} &:= \langle \mathit{dim} \rangle \{\mathit{Contribution}\} \mid \mathit{Nom} \mid \\ &\quad \mathit{Contenu} \vee \mathit{Contenu} \mid \mathit{Contenu} \wedge \mathit{Contenu} \end{aligned}$$

La première règle correspond à la notion d’abstraction : une classe permet d’associer un contenu à un identifiant. La seconde règle fait intervenir les trois autres concepts centraux de XMG, ceux de dimension, de conjonction et de disjonction. Ainsi, le contenu d’une abstraction peut être la contribution d’une description, l’utilisation d’une autre abstraction, la combinaison de contenus, ou bien l’expression d’une alternative entre contenus. Les contributions sont effectuées en spécifiant une dimension cible, une dimension étant un accumulateur correspondant à un niveau de description linguistique. Chaque accumulateur étant indépendant des autres, le concept de dimension permet aux métagrammaires utilisant le formalisme XMG de modéliser l’interface entre les niveaux de description linguistique (par exemple l’interface entre syntaxe et sémantique), puisqu’il est possible de partager de l’information explicitement entre dimensions au moyen de variables d’unification.

Si XMG a principalement été utilisé pour décrire la syntaxe des langues, la modularité du compilateur lui permet d’être facilement adapté à de nouvelles tâches de description, par la création de nouveaux modules. La création d’un nouveau compilateur, pour un nouveau langage métagrammatical, est réalisée par un assemblage de ces modules, appelés briques (voir Petitjean 2014). Chaque module définit un langage de description, dédié à de nouvelles structures.

Certains travaux utilisant les nouvelles extensions de XMG ont prouvé que l’approche métagrammaticale pouvait se révéler utile pour la description de différents autres niveaux de description linguistique. Les travaux en question s’intéressent notamment à la représentation de la sémantique au moyen de structures de traits typées (Lichte *et al.*, 2013) et à celle de la morphologie verbale de l’ikota, langue bantoue (Duchier *et al.*, 2012). C’est la dimension morphologique de XMG créée pour cette dernière tâche que nous utilisons pour modéliser la morphologie verbale de l’arabe. Nous formulons donc l’hypothèse que l’outil n’est pas seulement adapté à la description de langues agglutinantes (telles que l’ikota), mais également à celle de langues sémitiques.

4 Metagrammaire de la morphologie verbale de l'arabe

La méthode utilisée pour la description de l'ikota dans (Duchier *et al.*, 2012) consiste à contribuer des morphèmes dans des champs topologiques ordonnés. Le fait que l'ordre de ces champs soit fixe différencie cette tâche de la nôtre. Le nombre et les contraintes sur l'ordre des contributions à la dimension morphologique diffèrent selon la racine et le patron utilisés.

Nos descriptions contiennent donc trois types d'information :

(i) des contraintes sur le nombre et l'ordre des champs,
(ii) des instructions affectant un contenu (soit une chaîne de caractères) dans un champ,
(iii) des informations morphosyntaxiques sous la forme de structures de traits,
(iv) des descriptions de frames.

En comparaison, la métagrammaire de l'ikota ne contient que les types d'information (ii) et (iii). La métagrammaire peut être vue comme un assemblage de blocs élémentaires (notation empruntée à (Duchier *et al.*, 2012)), chacun de ces blocs pouvant contenir ces quatre types d'information. L'exemple de bloc élémentaire de la figure 3 définit deux champs topologiques nommés `C1` et `C2`, et contraignant leur ordre grâce à l'opérateur de précédence linéaire `>>`. La deuxième partie du bloc indique la contribution de la lettre /k/ dans le champ `C1`, et la troisième ajoute un trait morphosyntaxique précisant le patron utilisé dans la dérivation. La quatrième partie contient la description d'une frame très générale de type *activity*. Le langage de description utilisé est celui développé pour XMG dans (Lichte & Petitjean, to appear).

```
champ C1
champ C2
C1 >> C2
---
C1 <- k
---
patron = p1
---
[activity,
 actor:?X1 ]
```

FIGURE 3 – Example de description morpho-sémantique dans XMG

La figure 4 présente la métagrammaire que nous proposons. La classe *Forme* est l'unique axiome de cette métagrammaire, ce qui signifie que ce sont les modèles de cette classe que le compilateur doit calculer. Une *Forme* est obtenue par l'assemblage de quatre abstractions. La première, *Consonnes* est utilisée pour déclarer les trois champs qui contiennent les consonnes de la racine, ainsi que pour insérer ces dernières dans les champs. Les consonnes en question sont contenues dans des variables, les valeurs de ces variables étant obtenues par unification.

Un *Patron* est obtenu en combinant deux blocs élémentaires. Le premier réalise le même travail que le bloc *Consonnes* pour deux voyelles (utilisées dans chaque patron), soit la déclaration des champs, et l'insertion du contenu obtenu par unification. Le second bloc élémentaire est spécifique au patron, et est donc choisi parmi un ensemble de blocs (un par patron). La deuxième de ces alternatives, correspondant au patron 2 ($C_1aC_2C_2aC_3$), déclare dans un premier temps un nouveau champ topologique (`C21`), pour recevoir la consonne géminée, et ordonne la totalité des champs. Dans un second temps, on place dans le nouveau champ la deuxième consonne de la racine. Enfin, on ajoute l'information que le patron utilisé pour construire cette forme est le deuxième.

L'abstraction *Voix* permet d'exprimer l'alternative entre les blocs élémentaires associés aux voix active et passive. Chacun d'entre eux donne simplement la valeur des deux voyelles utilisées dans les patrons (/a/ et /a/ pour l'actif, /u/ et /i/ pour le passif). Enfin, l'abstraction *Racine* exprime le choix de la racine verbale (ici, par exemple, écrire, étudier et marcher, respectivement /ktb/, /drs/, et /m$y/).

Le résultat de la combinaison des blocs pour le deuxième patron, la voix active et la racine /m$y/ est montré dans la figure 5. Cette accumulation, après résolution (c'est à dire ordonnement des champs et concaténation de leurs contenus) produit une forme intermédiaire de l'entrée lexicale *ma$$ay*. Pour créer la forme finale *ma$~Y*, des règles morphophonémiques ulterieures doivent être appliquées, qui ne sont pas montrées ici.

5 Autres travaux

Comme annoncé précédemment, la motivation initiale pour notre travail était de fournir une approche pour la morphologie en arabe plus modulaire que celles utilisant des méthodes basées sur les automates finis. XMG offre un cadre à la fois déclaratif, flexible et multi-dimentionnel. Par conséquent il semble particulièrement adapté à la modélisation de morphologies non-concaténatives et de l'interface morpho-sémantique. Cependant, une limitation cruciale est qu'XMG ne peut jusqu'ici être utilisé que pour la génération.

(Bhuyan & Ahmed, 2008) réalisent l'une des rares propositions pour intégrer une morphologie basée sur les racines et les patrons à une grammaire de précision. Ils proposent d'étendre l'architecture basée sur les traites de HPSG avec un trait

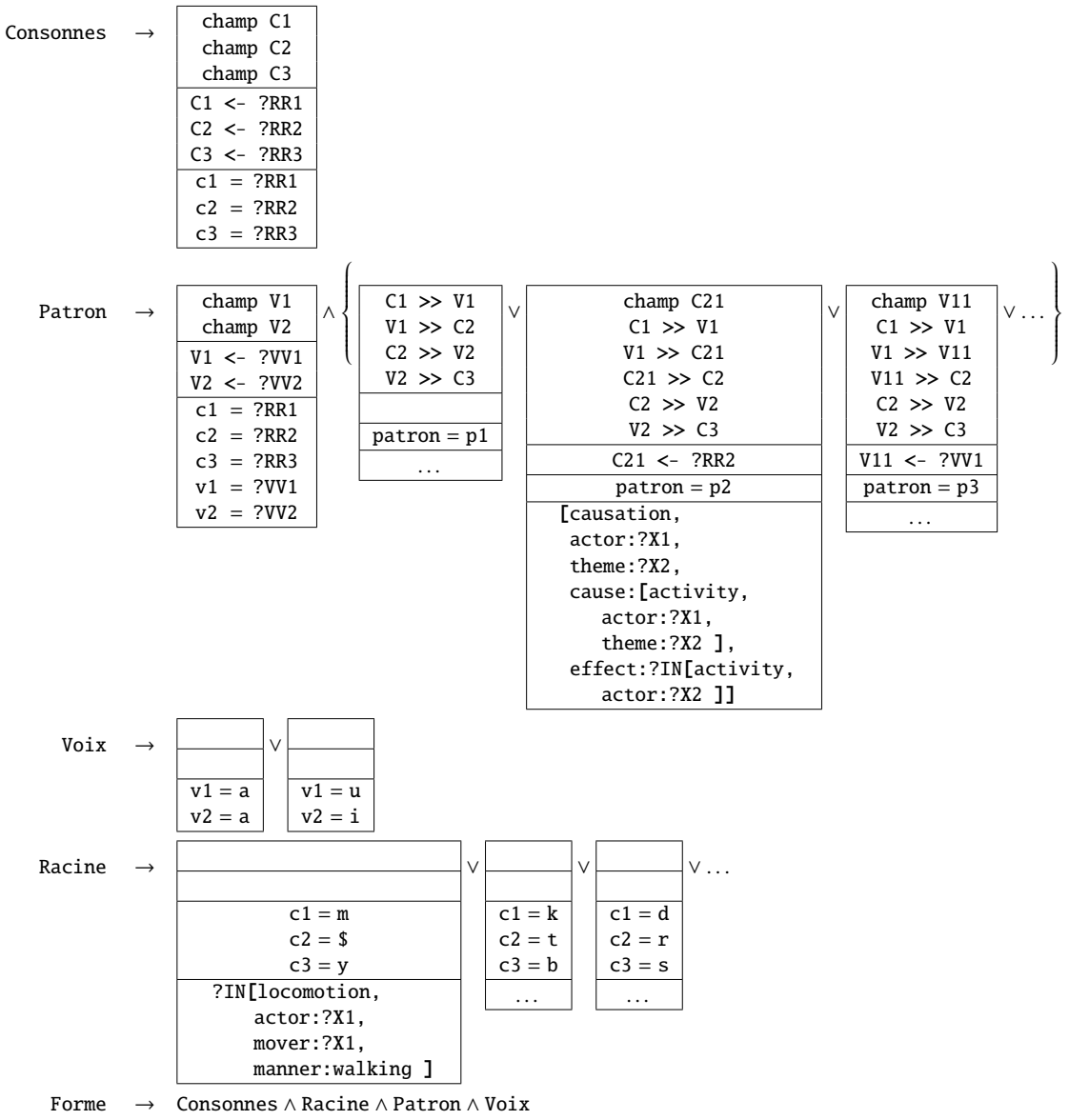

FIGURE 4 – Métagrammaire de la morphologie verbale de l'arabe

MORPH contenant une représentation richement structurée des racines et des patrons. Toutefois, Bhuyan et Ahmed abordent les combinaisons de racines et de patrons sans les décomposer sémantiquement.

Notre approche partage d'avantage de similarités avec des travaux relativement récents sur la morphologie constructionnelle reportés dans (Schneider, 2010). Dans le cadre de l'étude de l'hébreu moderne, Schneider propose d'augmenter la morphologie racine-patron de l'hébreu avec une sémantique compositionnelle, en mettant de côté les problèmes phonologiques. Néanmoins, il utilise les conventions de notation propres à la *Embodied Construction Grammar* (Bergen & Chang, 2005), qui sont très différentes des nôtres, au moins en surface. Un outil d'implémentation et un parser pour les grammaires ECG est disponible, [3] mais aucun outil pour la génération ne semble exister.

3. Voir `http://www1.icsi.berkeley.edu/~lucag/`.

```
champ C1
champ C2
champ C3
champ C21
champ V1
champ V2

C1 >> V1
V1 >> C21
C21 >> C2
C2 >> V2
V2 >> C3

C1 <- m
C2 <- $
C3 <- y

V1 <- a
V2 <- a
C21 <- $

patron = p2
  v1 = a
  v2 = a

  c1 = m
  c2 = $
  c3 = y

[causation,
 actor:?X1,
 theme:?X2,
 cause:[activity,
    actor:?X1,
    theme:?X2 ]
 effect:?IN[activity,
    actor:?X2 ]]

?IN[locomotion,
    actor:?X1,
    mover:?X1,
    manner:walking ]
```

FIGURE 5 – Une accumulation pour la classe `Forme` de la figure 4 incluant le patron 2, la voix active et la racine /m$y/, menant à la solution *ma$$ay*

6 Conclusion et perspectives

Nous avons présenté une formalisation de la morphologie verbale de l'arabe sous la forme d'une métagrammaire. À partir de cette description, le compilateur XMG génère un lexique de formes verbales non fléchies. Les travaux présentés dans cet article sont la première étape d'un projet plus ambitieux : l'objectif est d'enrichir ce lexique en intégrant l'interface morpho-sémantique. Nous utiliserons pour ceci des structures de traits typées, que nous décrirons au moyen de la dimension sémantique proposée dans (Lichte & Petitjean, to appear). La dimension morphologique de XMG utilise des séquences ordonnées de champs topologiques, l'ordre de ces champs étant défini dans la métagrammaire, ce qui constitue une extension de la dimension utilisée dans (Duchier *et al.*, 2012), apportant la flexibilité nécessaire au traitement de l'arabe. Nous prévoyons à terme d'intégrer des lexiques générés de cette manière à des chaînes de traitement plus importantes, par exemple dans le cadre d'une analyse syntaxique.

Ce travail pourrait évoluer par la suite dans différentes directions. D'un point de vue technique, la dimension morphologique de XMG pourrait être enrichie pour permettre des descriptions plus compactes (par exemple une notation concaténative telle que `C1 >> V1 >> C2 >> V2 >> C3` pourrait remplacer un ensemble de contraintes binaires de précédence linéaire) ainsi qu'un opérateur de précédence non immédiate. De plus, une dimension phonologique pourrait être ajoutée pour la prise en compte de règles morphophonémiques. En ce qui concerne la couverture, d'autres racines et patrons doivent être pris en compte, ainsi que l'affixation et l'attachement de clitiques. Enfin, nous pourrions étudier plus en détail les analyses obtenues en utilisant la morphologie constructionnelle, et éventuellement réimplémentées.

Remerciements

Les travaux présentés dans cet article ont été financés par la fondation allemande pour la recherche (Deutsche Forschungsgemeinschaft, DFG), par l'intermédiaire du SFB 991. Nous remercions également les trois relecteurs de TALN pour leurs précieux commentaires.

Références

ALTABBAA M., AL-ZARAEE A. & ARIF SHUKAIRY M. (2010). *An Arabic Morphological Analyzer and Part-Of-Speech Tagger*. PhD thesis, Arab International University, Damascus, Syria. Thèse.

Barsalou L. (1992). Frames, concepts, and conceptual fields. In A. Lehrer & E. F. Kittey, Eds., *Frames, fields, and contrasts : New essays in semantic and lexical organization*, p. 21–74. Hillsdale : Lawrence Erlbaum Associates.

Beesley K. (1998). Arabic morphological analysis on the internet. In *Proceedings of the International Conference on Multi-Lingual Computing*.

Bergen B. & Chang N. (2005). Embodied Construction Grammar in simulation-based language understanding. In J.-O. Östman & M. Fried, Eds., *Construction Grammars : Cognitive grounding and theoretical extensions*, p. 147–190. Amsterdam : John Benjamins.

Bhuyan M. S. I. & Ahmed R. (2008). An HPSG analysis of Arabic verb. In *The International Arab Conference on Information Technology (ACIT 2008)*.

Buckwalter T. (2004). Issues in arabic orthography and morphology analysis. In *Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages*, Semitic '04, p. 31–34, Stroudsburg, PA, USA : Association for Computational Linguistics.

Crabbé B., Duchier D., Gardent C., Le Roux J. & Parmentier Y. (2013). XMG : eXtensible MetaGrammar. *Computational Linguistics*, **39**(3), 1–66.

Danks W. (2011). *The Arabic Verb : Form and Meaning in the Vowel-lengthening Patterns*. Number 63 in Studies in functional and structural linguistics. Amsterdam : John Benjamins.

Doron E. (2003). Agency and voice : The semantics of the Semitic templates. *Natural Language Semantics*, **11**(1), 1–67.

Doron E. (2013). Binyanim : Modern Hebrew. In G. Khan, Ed., *Encyclopedia of Hebrew Language and Linguistics*. Brill Online. Available online at `http://referenceworks.brillonline.com/entries/encyclopedia-of-hebrew-language-and-linguistics/binyanim-modern-hebrew-EHLL_COM_00000247`.

Duchier D., Magnana Ekoukou B., Parmentier Y., Petitjean S. & Schang E. (2012). Décrire la morphologie des verbes en ikota au moyen d'une métagrammaire. In *19e conférence sur le Traitement Automatique des Langues Naturelles (TALN 2012) – Atelier sur le traitement automatique des langues africaines (TALAf 2012)*, p. 97–106, Grenoble, France.

Fillmore C. J. (1977). The case for case reopened. In P. Cole & J. M. Sadock, Eds., *Grammatical Relations*, volume 8 of *Syntax and Semantics*, p. 59–81. New York : Academic Press.

Habash N. & Rambow O. (2006). MAGEAD : A morphological analyzer and generator for the arabic dialects. In *ACL 2006, 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, Sydney, Australia, 17-21 July 2006*.

Kallmeyer L. & Osswald R. (2013). Syntax-driven semantic frame composition in Lexicalized Tree Adjoining Grammar. *Journal of Language Modelling*, **1**, 267–330.

Kiraz G. A. (2001). *Computational Nonlinear Morphology : With Emphasis on Semitic Languages*. New York, NY, USA : Cambridge University Press.

Lichte T., Diez A. & Petitjean S. (2013). Coupling trees, words and frames through XMG. In *Proceedings of the ESSLLI 2013 workshop on High-level Methodologies for Grammar Engineering*.

Lichte T. & Petitjean S. (to appear). Implementing semantic frames as typed feature structures with XMG. *Journal of Language Modelling*.

McCarthy J. J. (1981). A prosodic theory of nonconcatenative morphology. *Linguistic inquiry*, p. 373–418.

Petersen W. (2007). Representation of concepts as frames. *The Baltic International Yearbook of Cognition, Logic and Communication*, **2**, 151–170.

Petitjean S. (2014). *Génération Modulaire de Grammaires Formelles*. PhD thesis, Université d'Orléans. Thèse de Doctorat.

Ryding K. (2005). *A Reference Grammar of Modern Standard Arabic*. A Reference Grammar of Modern Standard Arabic. Cambridge University Press.

Schneider N. (2010). Computational cognitive morphosemantics : Modeling morphological compositionality in Hebrew verbs with Embodied Construction Grammar. In *Proceedings of the 36th Annual Meeting of the Berkeley Linguistics Society (BLS)*. Available online at `http://www.cs.cmu.edu/~nschneid/bls36.pdf`.

Shaalan K. F., Samih Y., Attia M., Pecina P. & van Genabith J. (2012). Arabic word generation and modelling for spell checking. In *LREC*, p. 719–725.

Yona S. & Wintner S. (2005). A finite-state morphological grammar of Hebrew. In *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, p. 9–16, Ann Arbor, Michigan : Association for Computational Linguistics.