Journal of Engg. Research Vol. 9 No. (3A) September 2021 pp. 134-142 DOI : 10.36909/jer.v9i3A.10129

A Prolog-based approach to Arabic syntax and semantics

Salah Alnajem*, A. M. Mutawa**, Hanan AlMeer*** and Aseel AlQemlas****

**Department of Arabic, College of Arts, Kuwait University*
***Department of Computer Engineering, Kuwait University*
****Department of Electrical and Electronic Engineering, University of Manchester*
*****Department of Information Science, Kuwait University*
Corresponding Author: salah.alnajem@ku.edu.kw

Submitted: 02/04/2020
Revised: 16/12/2020
Accepted: 22/12/2020

ABSTRACT

This paper introduces a computational approach to Arabic syntax. The approach uses the Lexical Functional Grammar (LFG) framework. Semantic networks and frames were used to handle computational semantics using lambda notation. This was implemented in Prolog using Definite Clause Grammar (DCG) as a formalism for analyzing and generating syntactic structure.

Keywords: Prolog; Morphology; Syntax; Semantics; LFG; DCG; Arabic; Frames; Semantic tree; Measures; Lexicon.

INTRODUCTION

In this paper, we present a Lexical Functional Grammar (LFG) approach to Arabic syntax. Semantic networks and frames were used to handle computational semantics using lambda notation. The implementation has been achieved in Prolog using Definite Clause Grammar (DCG) as a formalism for analyzing and generating syntactic structure.

The proposed system provides faster execution using a simple implementation, as shown in the results section. For Arabic morphological generation using the backtracking feature in Prolog, the system is capable of generating a lexicon containing a large number of words using a few roots and morphological templates (measures) fed to the system. The generated words are combined with all their morpho-syntactic features, taking into consideration perfective, imperfective, and imperative affixation rules and morphological generalizations. The system is also able to analyze the syntax of Arabic sentences and assign the correct syntactic structures, represented as parse trees, to those sentences. Moreover, the system can check the correctness of the semantics of Arabic sentences and generate the correct lambda notation for these sentences to represent their semantic structure.

The remainder of this paper is organized as follows. The next section provides essential background information about the subject of this paper, followed by the methodology, the results and discussion and, finally, the conclusions.

BACKGROUND

LFG's formal architecture was developed in the late 1970s. It was first described in detail in 1982 (Kaplan, *et al.*, 1982; Bresnan, 1982; Dalrymple, 1993) and further developed in subsequent works (Kaplan, *et al.*, 1995; Bresnan, *et al.*, 2015). The theory, which was motivated by psycholinguistic considerations, brought together several ideas that emerged from computational and linguistic investigations that have been carried out since the early 1970s.

In LFG, and in other theories in linguistics, phrase structure trees are used to represent phrasal groupings and word order in sentences. A constituent structure (C-structure) tree contains lexical categories such as noun (N), verb (V), and adverb (ADV). X-bar theory and its notations are widely used to describe and organize constituent structure (see Carnie, 2015, for a comprehensive introduction to the theory).

In LFG, Functional Structure (F-structure) is used to represent abstract grammatical functions like predicate, subject, and object. Morpho-syntactic features are also used, which include tense, case, person, and number. In the overall sentence structure, each lexical unit has its own F-structure that is combined with other F-structures representing other units to build an F-structure for the whole sentence.

A large amount of work has been built up in the years since the introduction of the theory of LFG. Among the advances in the theory of functional structure is the ability to characterize nonlocal relations between F-structures, allowing for a formally well-defined treatment of long-distance dependencies and constraints on anaphoric binding (see, among others, Dalrymple, 1993; Dalrymple *et al.*, 1995; and Bredenkamp *et al.*, 1996).

Another major focus of LFG study in recent years is the relation between syntactic and argument structure. Research conducted by Levin (1986) focused on the connection between thematic argument structure and grammatical functions and the generalizations that govern this connection. Argument structure is concerned with how thematic roles (such as agent, patient, and theme) in a syntactic representation relate to syntactic functions (such as subjects, objects, and adjuncts).

Another major focus in LFG from the beginning was lexical integrity and the various levels at which word-hood can be defined (Bresnan *et al.*, 1995; Cho *et al*, 1995). Another line of investigation in LFG is with the phenomena of agreement (Kibort, 2006; Beavers *et al.*, 2004; Kibort, 2008; Mahowald *et al.*, 2011). There has also been work exploring constituent structure and its relationship with the functional structure of LFG (Dalrymple, 1995; Kaplan *et al.*, 1995; Falk, 2001). Recent work in LFG explored semantic composition and the syntax–semantics interface (Pylkkänen *et al.*, 2006; Copestake, 2007; Van Valin Jr., 2013). There has also been work on using the deductive approach for the assembly of meanings that relies on the projection architecture of LFG to specify the correspondence between the F-structure of LFG and its meaning (Crouch et al., 1997).

The use of linear logic as a glue for assembling meanings allows for the proper treatment of a range of phenomena (Andrews, 2010), quantifier scoping, and bound anaphora and their interactions with intentionality (Migdalski, 2010). LFG has also been used for stem processing (Al Ajeeli, 2016), to handle Arabic syntax (Salloum *et al.*, 2016; Zaki *et al.*, 2016; Camilleri *et al.*, 2018), and for managing morphological and syntactic ambiguity (Attia, 2008). LFG has also been used for enhancing Arabic named-entity recognition (Aotaiwe, 2019). It has also been used for information retrieval in Arabic (Gashaw *et al.*, 2019).

Some studies have used Head-Driven Phrase Structure Grammar (HPSG) for Arabic verbs (Bhuyan *et al.*, 2008) and nominal sentences (Mutawa *et al.*, 2008). Lately, attention has been paid to using LFG to handle Arabic grammar (Attia, 2006; Attia, 2007; Attia *et al.*, 2010, and Arabic morphological and syntactic ambiguity (Attia, 2012).

ARABIC SENTENCE STRUCTURE IN LFG

To see how C-structure and F-structure are related, consider Fig. 1, which shows the C-structure of the Arabic sentence زيد أكل تفاحة (“Zaid ate an apple”). In this figure, each component in the F-structure corresponds to a C-structure node.

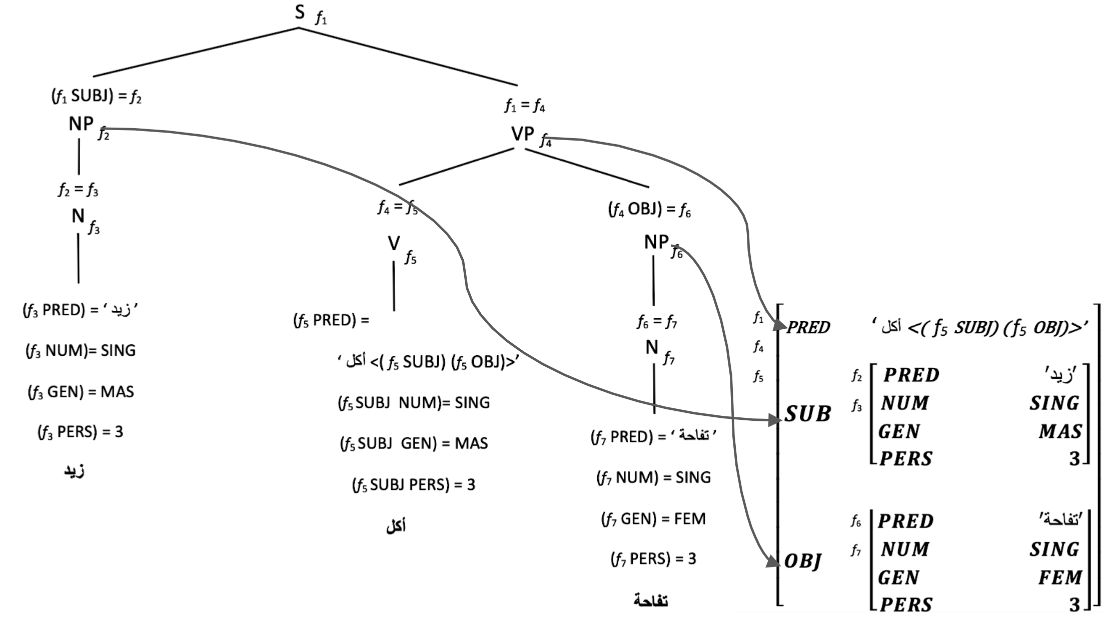


Figure 1. Relationship between C-structure and F-structure.

METHODOLOGY

In this section, we introduce our LFG approach to Arabic syntax and semantics using Prolog. Prolog is a declarative programming language based on first-order logic; it uses facts and rules to handle relations. Before we can describe our Prolog approach, we must explain DCG notation, semantic networks, frames, and computational semantics.

DCG Notation

DCG is a formalism used to express language grammar. It was originally introduced by Colmerauer (Alain, 1975). A Prolog program can be used to execute grammar defined using the DCG notation in order to verify if a list of symbols correctly represents a valid sentence in the grammar being defined. In this respect, a DCG specification defines a linguistic grammar and a parser for that grammar. Some DCG specifications can be used "in reverse". This means that those specifications can be used to generate valid sentences according to a specific grammar.

Semantic Networks

To achieve semantic knowledge representation, semantic networks are used instead of predicate logic. In this type of representation, knowledge is stored in the form of a graph in which objects in the world are represented by nodes and arcs are used to represent relationships between those objects. This semantic representation resembles the way humans structure knowledge, and it is analogous to the mental links between objects in the human mind. Each node representing an object contains all the information about that object.

In the representation of nodes, a distinction is made between individual nodes (instance nodes) and nodes representing classes. To indicate that an object belongs to a class, the "is_a" link is used. The label "a_kind_of", or "ako" for short, is used to label a link that represents the fact that one class is a subset of another.

Semantic networks can be used to apply a form of inference known as inheritance. Inheritance indicates that if an object belongs to a subclass that is connected to another class by "a_kind_of" link, then the subclass will inherit all the properties of that class. Inheritance also applies across the "is_a" links between instances and classes.

Fig. 2 shows the general class “animate”. Most animates have the attributes of being sensible, moving, and “can_eat”. The values of all these attributes are Boolean: the sensible attribute is fixed to a false value, while moving and “can_eat” are fixed to true at the level of “animate”. Two subclasses of animate—human and plants—are defined. Human class overrides the sensible attribute, making it fixed to true. Human class inherits the moving and “can_eat” attributes from animate, but adds an additional gender attribute that is fixed to masculine (mas). Plant class overrides both the moving and “can_eat” attributes inherited from the animate class and makes them fixed to false. Plant class inherits the sensible attribute from animate, and an additional attribute is added—green—that is fixed to true. Two subclasses—boy and girl—inherit all attributes of the class human, except that the subclass girl overrides the gender attribute and makes it fixed at feminine (fem). There is a subclass called trees that inherits all the plant class attributes (nothing is overridden). Finally, the instance “ولد” inherits all the boy class attributes, instance “بنت” inherits all the girl class attributes, and instance “شجرة” inherits all the tree class attributes.

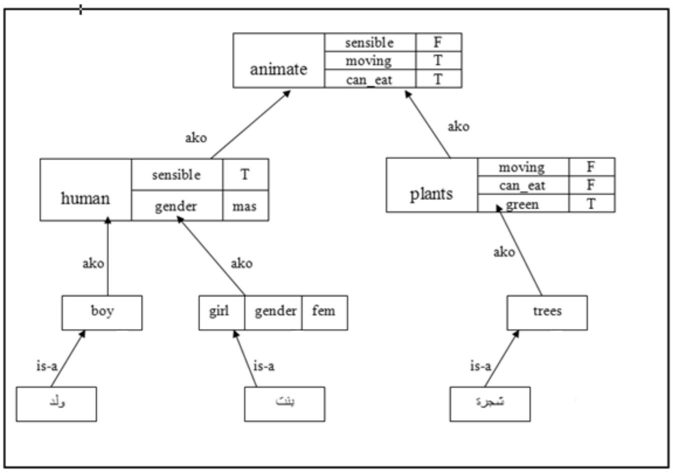


Figure 2. Semantic network example showing different inheritance and override attributes.

Computational Semantics

We must mention here that in computational semantics, there are a number of problems that do not yet have accepted solutions. In principle, an entirely adequate theory of semantics would require a complete theory of human thought.

Table 1 shows logical formulas representing English words with the corresponding representation of those formulas in Prolog. Lambda is represented as ^, which is right associative so that X ^ Y ^ formula = X ^ (Y ^ formula).

Table 1. Logical formulas representing English words and phrases.

Type of constituent	**Logical representation**	**As written in Prolog**
Proper noun John	Logical constant john	john
Common noun dog	1-place predicate (λx)dog(x)	X^dog(X)
Transitive verb chased	2-place predicate (λy)(λx)chased(x,y)	Y^X^chased(X,Y)

Implementation using Prolog

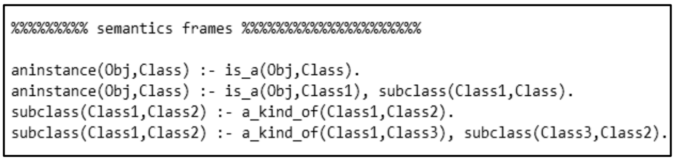
Arabic roots were fed into the system by saving them in a text file called root.pl for trilateral roots and root4.pl for quadrilateral roots. Once the Prolog code was compiled, it consulted these two files and added all these roots to the Prolog knowledge base. Next, the system generated all stems from the roots using measures. These stems were used to form a lexicon. Table 2 shows part of the lexicon generated using the stem كاتَب (corresponded in writing).

Table 2. Sample of the console log showing part of the lexicon generated using the stem كاتَب.

Verb	Measure	Person	Number	Gender	Tense
v(كاتَبتُ)	فاعَلتُ	First	Singular	Mas	Past
v(كاتَبتُ)	فاعَلتُ	First	Singular	Fem	Past
v(كاتَبنا)	فاعَلنا	First	Dual	Mas	Past
v(كاتَبنا)	فاعَلنا	First	Dual	Fem	Past
v(كاتَبنا)	فاعَلنا	First	Plural	Mas	Past
v(كاتَبنا)	فاعَلنا	First	Plural	Fem	Past
v(كاتَبتَ)	فاعَلتَ	Second	Singular	Mas	Past
v(كاتَبتِ)	فاعَلتِ	Second	Singular	Fem	Past
v(كاتَبتُما)	فاعَلتُما	Second	Dual	Mas	Past
v(كاتَبتُما)	فاعَلتُما	Second	Dual	Fem	Past
v(كاتَبتُم)	فاعَلتُم	Second	Plural	Mas	Past
v(كاتَبتُنَّ)	فاعَلتُنَّ	Second	plural	Fem	Past

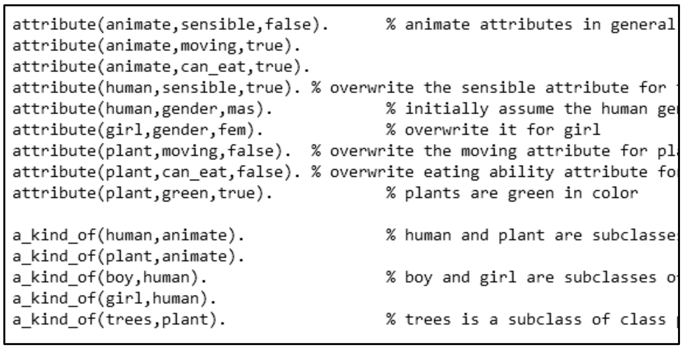
The six fields (verb, measure, person, number, gender, and tense) for every generated word were automatically filled. It is worth mentioning that the system is capable of generating a large amount of words from the few roots fed to it along with all their morpho-syntactic features using the backtracking feature in Prolog, which takes into consideration perfective, imperfective, and imperative affixation rules.

The code used to generate the lexicon consists of around 500 lines. Semantic frames were implemented in Prolog as rules representing the "is_a" and "kind_of" relations and connecting them to instances of objects, as depicted in Fig. 3. Objects and their attributes were encoded as frames using relation tuples, as shown in Fig. 4. Notice how girl gender value "fem" overrides the default "mas" value of human. Semantics were asserted into the working memory in the form of rules, as shown in Fig. 5.



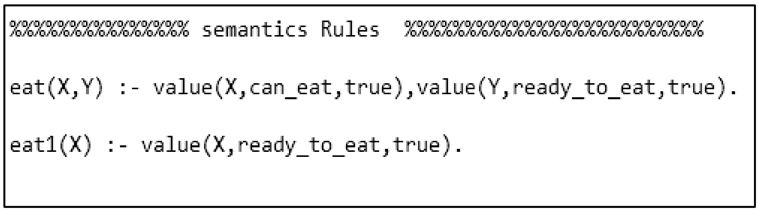
```
%%%%%%%%% semantics frames %%%%%%%%%%%%%%%%%%%%%

aninstance(Obj,Class) :- is_a(Obj,Class).
aninstance(Obj,Class) :- is_a(Obj,Class1), subclass(Class1,Class).
subclass(Class1,Class2) :- a_kind_of(Class1,Class2).
subclass(Class1,Class2) :- a_kind_of(Class1,Class3), subclass(Class3,Class2).
```

Figure 3. Sample of the console log showing the semantic frames representation in Prolog.



```
attribute(animate,sensible,false).      % animate attributes in general
attribute(animate,moving,true).
attribute(animate,can_eat,true).
attribute(human,sensible,true). % overwrite the sensible attribute for
attribute(human,gender,mas).            % initially assume the human ge
attribute(girl,gender,fem).             % overwrite it for girl
attribute(plant,moving,false).  % overwrite the moving attribute for pl
attribute(plant,can_eat,false). % overwrite eating ability attribute fo
attribute(plant,green,true).            % plants are green in color

a_kind_of(human,animate).               % human and plant are subclasse
a_kind_of(plant,animate).
a_kind_of(boy,human).                   % boy and girl are subclasses o
a_kind_of(girl,human).
a_kind_of(trees,plant).                 % trees is a subclass of class
```

Figure 4. Objects and their attributes encoded as frames using relation tuples.



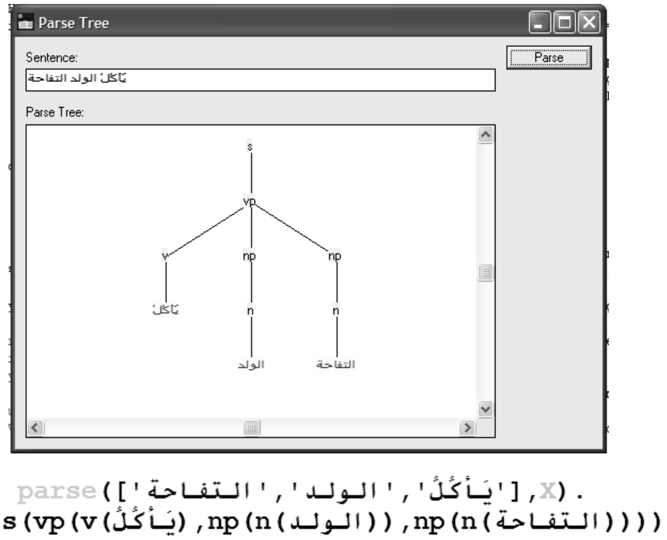
```
%%%%%%%%%%%%%% semantics Rules  %%%%%%%%%%%%%%%%%%%%%%%

eat(X,Y) :- value(X,can_eat,true),value(Y,ready_to_eat,true).

eat1(X) :- value(X,ready_to_eat,true).
```

Figure 5. Semantic rules for the example of the verb “eat” represented as unary and binary relations.

RESULTS AND DISCUSSION

The code was implemented and compiled using Logic Programming Associates (LPA) Win-Prolog 5.0 (L. P. Associates, 2014) on an Intel Core i7 @3.5GHz CPU equipped with 16 GB RAM running on a 64-bit Windows 8.1 Pro OS. The code takes only 1 ms to generate all stems per root and 13 ms to generate all lexical entries generated from a root, on average.

The system was tested using the sentence يأكُلُ الولد التفاحة (The boy eats the apple). It gave a positive result represented by a parse tree, as shown in Fig 6. The positive result indicates a semantically and syntactically correct sentence.

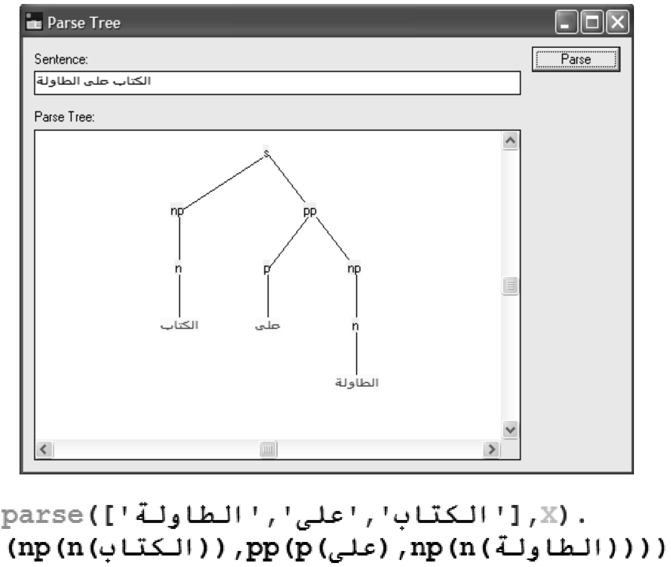


Sentence: يَأكُلُ الولد التفاحة

```
| ?- parse(['التفاحة','الولد','يَأكُلُ'],X).
X = s(vp(v(يَأكُلُ),np(n(الولد)),np(n(التفاحة))))
```

Figure 6. A parse tree and parse text of the sentence يأكُلُ الولد التفاحة (The boy eats the apple).

The system was also tested using the sentence الكتاب على الطاولة (The book is on the table), which also gave a positive result, as shown in Fig. 7.

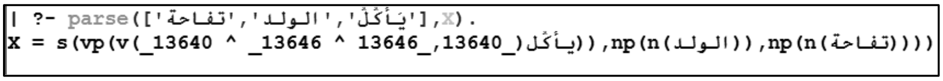


Sentence: الكتاب على الطاولة

```
| ?- parse(['الكتاب','على','الطاولة'],X).
X = s(np(n(الكتاب)),pp(p(على),np(n(الطاولة))))
```

Figure 7. A parse tree and parse text of the sentence الكتاب على الطاولة (The book is on the table).

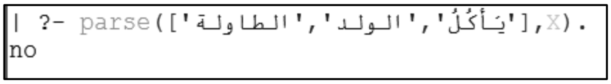
Another sentence was tested with an indefinite noun: يأكل الولد تفاحة (The boy eats an apple). The system generated the correct lambda notation for this sentence, as shown in Fig. 8.



```
| ?- parse(['يَأكُلُ','الولد','تفاحة'],X).
X = s(vp(v(_13640 ^ _13646 ^ يأكل(_13646,_13640))),np(n(الولد)),np(n(تفاحة))))
```

Figure 8. A parse text of the sentence يأكل الولد تفاحة (The boy eats an apple).

The system was also challenged with a semantically unacceptable but syntactically correct sentence. The sentence يأكل الولد الطاولة (The boy eats the table) was entered into the system, but it was rejected due to the semantic conflict of having a human eating an object that is not edible, as shown in Fig. 9.



```
| ?- parse(['يَأكُلُ','الولد','الطاولة'],X).
no
```

Figure 9. Rejecting the semantically unacceptable sentence يأكل الولد الطاولة (The boy eats the table).

CONCLUSION

This paper introduced the design and implementation of an approach to Arabic syntax and semantics using LFG, semantic networks, frames, and lambda notation in Prolog programming language. The paper has shown how syntactic generalizations and rules can be computationally implemented using LFG formalism through C-structures and F-structures. Thus, we have implemented a morphological generator that generates a lexicon consisting of a large number of Arabic words and their morphosyntactic features from trilateral and quadrilateral roots using measures, taking into consideration perfective, imperfective, and imperative affixation rules and morphological generalizations. A dictionary file was used to map concept items into the semantic network using an "is_a" relation. The proposed system provides a fast execution using a simple implementation that allows for the syntactic and semantic checking of Arabic sentences in a timely manner and with low overhead. Finally, our system can be used for implementing a semantic-based translator. The same system can be adapted to handle Arabic syntax and semantics using HPSG formalism.

REFERENCES

Al Ajeeli, A.T. 2016. An intelligent framework for natural language stems processing. Global Journal of Computer Science and Technology, **16**(1): 22-38.

Alain, C. 1975. Les grammaires de métamorphose GIA. Natural Language Communication with Computers, **63**: 133-189.

Andrews, A.D. 2010. Propositional glue and the projection architecture of LFG. Linguistics and Philosophy, **33**: 141-170.

Aotaiwe, A. 2019. Enhancing Arabic named entity recognition using parallel techniques. Journal of Theoretical and Applied Information Technology, **97**(6): 1775-1787.

Attia, M.A. 2006. Accommodating multiword expressions in an Arabic LFG grammar. In Salakoski T., Ginter F., & Pyysalo S. (eds.). Advances in Natural Language Processing, 5th International Conference on NLP, FinTAL 2006. Springer-Verlag, Berlin, Pp. 87-98.

Attia, M. 2007. Arabic tokenization system. Proceedings of the 2007 workshop on computational approaches to Semitic languages: Common issues and resources, pp. 65-72, Czech Republic.

Attia, M.A. 2008. Handling Arabic morphological and syntactic ambiguity within the LFG framework with a view to machine translation. PhD Dissertation, University of Manchester, UK.

Attia, M. 2012. Ambiguity in Arabic computational morphology and syntax: a study within the lexical functional grammar framework. Saarbrücken, LAP Lambert Academic Publishing.

Attia, M., Tounsi, L., Pecina, P., Van Genabith, J., & Toral, A. 2010. Automatic extraction of Arabic multiword expressions. Proceedings of the Multiword Expressions: From Theory to Applications (MWE 2010), pp.19–27, China.

Beavers, J., & Sag, I.A. 2004. Coordinate ellipsis and apparent non-constituent coordination. In Stefan Müller (ed.). Proceedings of the 11th International Conference on Head-Driven Phrase Structure Grammar. Centre for Computational Linguistics, Katholieke Universiteit Leuven, pp. 48-69, Belgium.

Bhuyan, M., S. I. & Ahmed, R. 2008. An HPSG analysis of arabic verb. The International Arab Conference on Information Technology.

Börjars, K. 2020. Lexical-functional grammar: an overview. Annual Review of Linguistics, **6**: 155-172.

Bredenkamp, A., Fouvry, F., Declerck, T., & Music, B. 1996. Efficient integrated tagging of word constructs. Proceedings of the 16th conference on Computational linguistics, 2:1028-1031, Denmark.

Bresnan, J. 1982. The mental representation of grammatical relations, The MIT Press, Cambridge.

Bresnan, J., & Mchombo, S.A. 1995. The lexical integrity principle: Evidence from Bantu. Natural Language & Linguistic Theory, **13**: 181-254.

Bresnan, J., Asudeh, A., Toivonen, I., & Wechsler, S. 2015. Lexical-functional syntax. Blackwell, Oxford.

Camilleri, M., & Sadler, L. 2018. Schematising (morpho) syntactic change in LFG: Insights from grammaticalisation in Arabic. In Butt, M., & King, T. H. (eds.). Proceedings of the LFG'18 Conference. Pp, 129-149. CSLI Publications, Stanford.

Carnie, A. 2015. Syntax: a generative introduction. Blackwell, Oxford.

Cho, Y., & Sells, P. 1995. A lexical account of inflectional suffixes in Korean. Journal of East Asian Linguistics, **4**:119-174.

Copestake, A. 2007. Semantic composition with (robust) minimal recursion semantics. Proceedings of the Workshop on Deep Linguistic Processing, **1**: 73-80, Czech Republic.

Crouch, J.V.G.R., Butt, M., & King, T.H. 1997. On comparing dynamic and underspecified semantics for LFG. Proceedings of the LFG97 Conference, U.S.A.

Dalrymple, M. 1993. The syntax of anaphoric binding. CSLI Lecture Notes No. 36. Centre for the Study of Language and

Information, Stanford.

Dalrymple, M., Kaplan, R.M., Maxwell, J.T., III & Zaenen, A. (eds.) 1995. Formal issues in lexical-functional grammar. CSLI Publications, Stanford.

Dalrymple, M. 2001. Lexical functional grammar. Academic Press, New York.

Falk, Y.N. 2001. Lexical-functional grammar: An introduction to parallel constraint-based syntax. CSLI Publications, Stanford.

Gashaw, I., & Shashirekha, H.L. 2019. Enhanced Amharic-Arabic Cross-Language Information Retrieval System using Part of Speech Tagging. 2019 International Conference on Advances in Computing, Communication and Control (ICAC3), pp. 1-7, India.

Kaplan, R.M. & Bresnan, J. 1982. Lexical-functional grammar: A formal system for grammatical representation. In: Bresnan J., (ed.). The Mental Representation of Grammatical Relations. Pp. 173-281. MIT Press, Cambridge.

Kaplan, R.M. & Zaenen, A. 1995. Long-distance dependencies, constituent structure, and functional uncertainty. In: Dalrymple, M., Kaplan R., Maxwell J., & Zaenen A., (eds.). Formal Issues in Lexical-Functional Grammar. Pp. 137-165. CSLI Publications, Stanford.

Kibort, A. 2006. On three different types of subjectlessness and how to model them in LFG. Proceedings of the LFG06 Conference, 289-309. University of Konstanz, Germany.

Kibort, A. 2008. Impersonals in Polish: an LFG perspective. Transactions of the Philological Society, **106**(2): 246-289.

Levin, L.S. 1986. Operations on lexical forms: Unaccusative rules in germanic languages. PhD Dissertation, MIT, U.S.A.

L. P. Associates. 2014. Win-Prolog 5.0. Available: http://www.lpa.co.uk.

Mahowald, K., Butt, M., & King, T.H. 2011. An LFG approach to word order freezing. In: Butt, M. & King, T. H. (eds.). Proceedings of LFG11. Pp.381-400. CSLI Publications, Stanford.

Migdalski, K. 2010. Studies in formal Slavic linguistics (review). Journal of Slavic Linguistics, **18**(2): 339-354.

Mutawa, A., Alnajem, S. & Alzhouri, F. 2008. An HPSG approach to Arabic nominal sentences. Journal of the American Society for Information Science and Technology, **59**(3): 422-434.

Pylkkänen, L. & McElree, B. 2006. The syntax–semantics interface: on-line composition of sentence meaning, In: Traxler, M., & Gernsbacher, M. A. (eds.). Handbook of Psycholinguistics, 2nd edition. Pp. 537-577. Elsevier, New York.

Salloum, S.A., Al-Emran, M., & Shaalan, K. 2016. A survey of lexical functional grammar in the Arabic context. International Journal of Computing and Network Technology, **4**(3): 141-147.

Van Valin Jr., R.D. 2013. Lexical representation, co-composition, and linking syntax and semantics. Advances in Generative Lexicon Theory. Text, Speech and Language Technology. Springer-Verlag, Berlin, Pp. 67-107.

Zaki, Y., Hajjar, H., Hajjar, M., & Bernard, G. 2016. A survey of syntactic parsers of Arabic language. Proceedings of the International Conference on Big Data and Advanced Wireless Technologies, 1-10.