Arabic language and its specification in TDL

Kais Haddar, Sirine Boukedi, Ines Zalila

Construction of an HPSG grammar for the Arabic language and its specification in TDL

Kais Haddar, Sirine Boukedi and Ines Zalila

Abstract— **The construction of an HPSG grammar (Head-driven Phrase Structure Grammar) treating Arabic specificities is not an easy task. In fact, several syntactic phenomena must be taken into account. Thus, the main objective of this work is to construct an Arabic HPSG grammar based on a proposed type hierarchy that categorizes Arabic words. In fact, some adaptations were introduced to HPSG, at the level of features and ID schemata. All linguistic resources (e.g., lexicon, type hierarchy, syntactic rules) are specified in the Type Description Language (TDL). The experimentation of the constructed grammar was done using the Linguistic Knowledge Building (LKB) platform containing generation tools. Indeed, the choice of TDL language is justified. It has syntax similar to HPSG representation and it is considered as the principal input to the LKB platform.**

Index Terms— **Arabic HPSG, relative clauses, TDL specification, LKB parser.**

I. Introduction

Natural Language Processing (NLP) covers principally four levels of treatments : lexical, syntactic, semantic and pragmatic. In NLP, syntactic analysis is fundamental for other phases such as semantic analysis. It is also necessary for several applications dealing with natural language such as human-machine dialogue systems, automatic translation and grammatical errors correction.

Despite this importance, the syntactic analysis has not been properly explored in the research domains related to the Arabic language, especially for complex phenomena like relative clauses. Thus, few works as [1], [3], [11], [17] and [21] have constructed grammars treating particular Arabic phenomena (e.g., nominal sentences, Verbal systems). In fact, in Arabic, there are several criteria to categorize words. Therefore, deciding for a hierarchical type is a difficult task. Moreover, there is a problem in the choice of the adequate grammar that can cover Arabic specificities. Indeed, there exist various types of grammars to represent different syntactic phenomena such as formal grammars which were used in syntactic domain. The construction of this type of grammars focused, particularly on the development of the syntactic rules. In fact, developers did not give importance to the lexicon. However, Arabic language has a very rich lexicon covering several types of constructions. Therefore, to represent Arabic constructions, various rules are required.

Besides, there is a problem in the grammar experimentation. In fact, there exist two approaches. The first one consists in designing and developing an individual parser. This approach supports maintenance and extensibility. Nevertheless, it requires the proposition of an adequate analysis algorithm and the description of the inputs/outputs. Thus, the proposition can influence the robustness of the results. For the second approach, it is based on parser generation tool. It allows the designer to concentrate on the grammar identification. Moreover, the inputs and outputs of the parser are well defined from the beginning. In the same way, the ergonomic of the interface is already tested. This approach is rather powerful; it makes it possible to generate reliable parsers. Indeed, there are several linguistic platforms (containing generation tools) designed for various formalisms such as the Linguistic Knowledge Building (LKB) [9].

The main objective of this work is to construct an HPSG grammar for the Arabic language based on a type hierarchy inspired from classic Arabic and respecting the Arabic language specificities. The experimentation of the established grammar is done using a linguistic platform (Linguistic Knowledge Building (LKB)). Thus, it aims to develop an adequate grammar that takes into account different phenomena of Arabic language including relative clauses. Relatives are very complex structures that are not well explored. To use LKB platform, the constructed grammar is specified in the Type Description Language (TDL). The TDL specification is original since it allows the combination of semi formal and formal modeling. Moreover, TDL integrates a set of operations and checks some concepts (e.g., inheritance, adjunction).

The present paper is organized as follows. In section 2, we review some related works focused on the syntactic analysis. In section 3, we propose a type hierarchy categorizing Arabic words. According to the proposed type hierarchy, we describe different interactions between Arabic words. Based on this study, we present in section 4, the established HPSG grammar as well as the different modifications brought to make it compatible with Arabic language. In section 5, we give the TDL specification of the conceived grammar and of the

Kais HADDAR is with the Sciences Faculty, Sfax –Tunisia. (Phone: 98657538; e-mail: kais.haddar@fss.rnu.tn).

Sirine BOUKEDI, is with the National Engineering School of Sfax - Tunisia (e-mail: serine_fss@yahoo.fr).

Ines ZALILA is with the National Engineering School of Sfax –Tunisia (e-mail : ines.zalila@yahoo.fr).

lexicon. Its experimentation was done in section 6. In fact, we experiment and evaluate the in TDL specificied HPSG grammar with LKB system. Therefore, we give an overview about this system then we describe the stages of syntactic analysis. Finally, we enclose the present paper by a conclusion and some perspectives.

II. Related works

Researchers having constructed an HPSG grammar treating Arabic specificities are not numerous; particularly those studying on complex phenomena. In fact, HPSG was used at the first time to specify French, English and Spanish languages such as [13], [16] and [22].

Indeed, in [13] Garcia constructed an HPSG grammar treating Spanish relatives, based on a proposed type hierarchy. Moreover, the author defined a lexicon including conjunctive nouns and introduced some syntactic rules treating Spanish relatives. Linguistic resources were specified in TDL and experimented on LKB platform.

In [16] and [22], authors constructed an HPSG grammar for French Language. The first one treated some constructions including coordination phenomenon. The second author studied French phrase affixes, particularly the forms “à” and “de”. Therefore, they proposed a set of syntactic rules covering these linguistic phenomena. The experimentation of the constructed grammars used also LKB platform.

For researchers treating Arabic language, few works proposed some modifications to HPSG (at the level of features and ID schemata). The adapted grammar covers Arabic specificities. In the following, we mention some works treating simple and complex phenomena.

In [3], authors studied the typology of Arabic nominal sentences and proposed an HPSG grammar generating essentially this type of sentence. Their prototype is implemented in a high language used a standard analysis algorithm. The HPSG experimentation was based on a lexicon file containing 20 entries and a rule file containing eight rules representing seven types of nominal Arabic sentences.

Moreover, in [21] authors extended an HPSG grammar to support the Arabic verbal morphology. In fact, based on a set of examples, they generated different morphological patterns representing derivation forms of Arabic verbs. Thus, to cover morphological aspect, they proposed a new feature MORPH containing three sub-features: TYPE, ROOT and MEASURE.

Besides, other Arabic works such as [6], [11] and [18] have treated complex phenomena (i.e., relatives, coordination). In [6], the author presents a study of relative clauses which shows that conjunctive nouns are not considered as determinants but as modifiers. In the same way, [11] proposed an Arabic HPSG grammar treating some simple and complex sentences. This work used a large number of production and dynamic rules.

In [18], some modifications were brought to HPSG grammar to cover Arabic coordination. In fact, the author developed a schema taking into account sentences containing joint components. To experiment the established grammars, [11] and [18] constructed an individual parser.

Referring to the related works, we relieved some problems at the level of TDL specification, grammar construction and experimentation. In fact, authors can not specify default constraints with TDL language which increases the number of syntactic rules. For Arabic works, researchers constructed grammars covering some particular phenomena. Therefore, we do not have complete grammars treating Arabic specificities which are insufficient at the lexical and syntactic levels.

The originality of our work is to construct a robust and efficient HPSG grammar, covering various Arabic phenomena (simple and complex). Since the HPSG representation differs from an entry to another according to the entry’s type, we propose in the following a type hierarchy for Arabic language.

III. Proposition of an Arabic type hierarchy

As we have mentioned previously, some adaptations were required to use HPSG grammar for Arabic language. In order to avoid ambiguous cases, we define each type with a complete representation covering the appropriate linguistic knowledge. After discussion with some linguistics [2] and [10], we proposed the type hierarchy represented in Fig. 1:

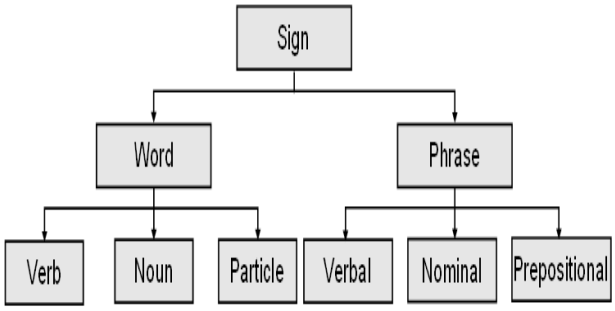


Fig. 1. Arabic word hierarchy. This figure explains the proposed categorization of an Arabic word.

Inspired from [2] and [10], we consider in our proposed type hierarchy that the Arabic type root is the linguistic sign «اللفظ, *lafZ*». This type can be a simple word «كلمة, *kalima*» or a phrase «مركب, *murakeb*». It should be noted that a phrase is composed from two or several words. Thus, to compose phrases representing Arabic phenomena, we are classified simple words based on some criteria. An Arabic word can be a verb, a noun or a particle. In the following, we detail each type of Arabic word.

A. Arabic verb

Several criteria were proposed to categories Arabic verbs. In fact, they can be subdivided according to the number of letters composing the verb or according to whether it is augmented « مزيد, *mazyd* » or denuded « مجرد, *mujarrad* ». We choose, in this paper to categorize verbs according to the first criterion. Thus, a verb can be triliteral «ثلاثي, *thulaathy*» or quadriliteral «رباعي, *rubaa’y*», as shown in Fig. 2.

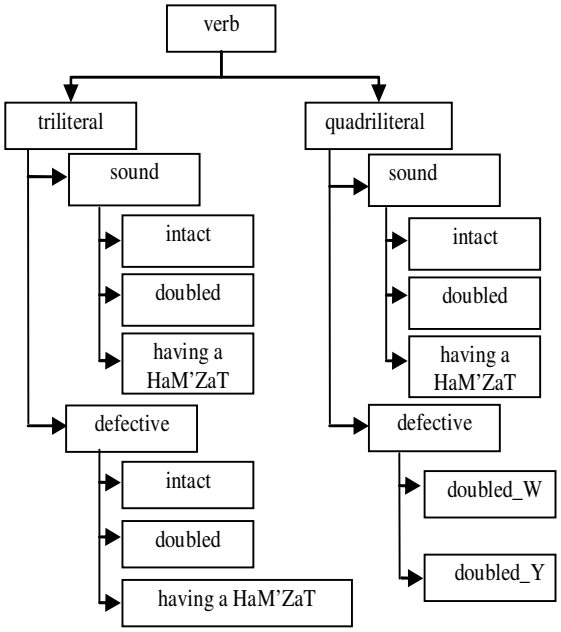


Fig. 2. Arabic verb's categories. Triliteral or quadriliteral verb can be sound or defective.

A verb is considered sound when it does not contain any defective particles (ا, و, ي). Contrary, a verb is defective when one of its particles is a defective one. Each type has different possible values what makes possible to distinguish various Arabic verbs.

The study on different verbs (triliteral or quadriliteral) showed that there exist transitive and intransitive verbs. Indeed, a transitive verb (متعدي, *muta'addy*) needs a subject and two or three complements. Most of this verbs' type can have direct and indirect complements. For the indirect one, they can be a prepositional phrase or a circumstantial object. Example (1) illustrates a transitive verb which has three objects.

(1) عاقب الأستاذ التلميذ بشدّة في القسم

The teacher punished hardly the pupil in the classroom

In this sentence, the verb 'عاقب, *'aaqaba*' is transitive. It has three objects. The first one represents a direct object complement 'التلميذ, *al-tilmydhu'*, the second and the third one represent prepositional phrases.

Moreover, intransitive verbs necessitate only a subject. But in some cases, we can add other objects to insist on a specific semantic phenomenon as shown in example (2):

(2) نام الولد نوما عميقا

The child slept deeply

At the grammatical level, the verb 'نام, *naama'* is intransitive. Whereas to express the manner of which the child slept, the verb has another object 'نوما عميقا, *deeply*'.

Moreover, an Arabic verbal sentence begins with a regular verb or a verbal phrase. In fact, in this case, the verb must be preceded by an operative particle as shown in example (3).

(3) لم يأكل الولد في المنزل

The boy did not eat at home.

Example (3) shows that an elided verb 'فعل مجزوم, *fi'l majzum*' is usually preceded by an elision particle 'حرف جزم, *harf jazm'*.

So to compose prepositional phrases, we present in the following section, the particle's categories and describe some constraints that must be taken into account to compose prepositional phrases.

B. Arabic particle

Referring to [2] and [10], an Arabic particle can be categorized on two types: operative particles and neglected particles. The first type operates on the associated compound (noun or verb) and the second type does not have any influence. Fig. 3 illustrates the two distinguished categories of Arabic particles

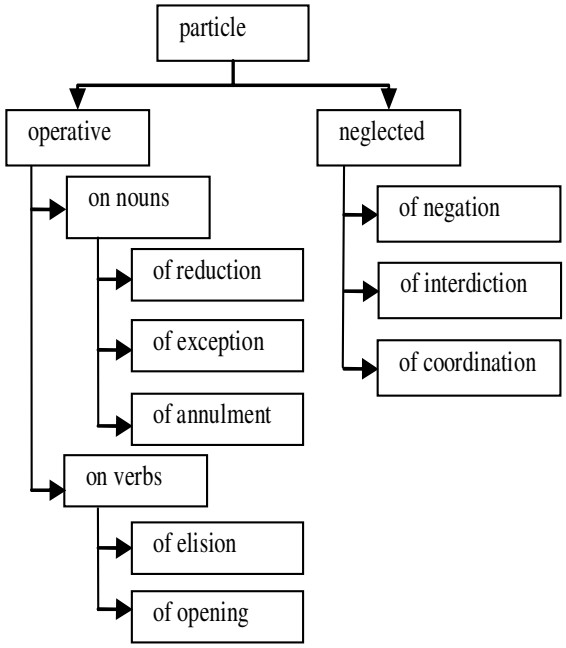


Fig. 3. Arabic particle's categories. There are two different categories: operative particles and neglected particles.

In fact, as represented in Fig. 3, neglected particles such as coordination particles (حروف العطف, *huruf al-'atf*), negation particles (حروف النفي, *huruf al-nafiy*) and interrogation ones (حروف الاستفهام, *huruf al-istifhaam*) do not influence on the declination of associated compound.

However, an operative particle changes the declination of associated compound (verb or noun). Therefore, we subdivide this category in two different classes: particles operating on nouns and particles operating on verbs. As example of noun operative particle, we can mention reduction particles 'حروف الجر, *huruf al-jar*' and annulment particles 'حروف النسخ, *huruf al-naskh*' as shown in examples (4) and (5):

(4) في المنزل, *At home,*

(5) كأنّ الولد مريض, *The boy seems ill,*

In those examples, each particle must be associated with a noun having a determined declination. In fact, the reduction particle must be followed by a reduced noun 'مجرور, *majrur*' and the annulment one must be followed by an open ending noun 'منصوب, *mansub*'.

C. Arabic noun

For nouns «الأسماء, *al-asmaa* », we choose to subdivide them according to their declination «الإعراب, *al-i'raab*». Thus, we find declined nouns «الأسماء المعربة, *al-asmaa al-mu'raba*» and indeclinable nouns «الأسماء المبنية, *al-asmaa al-mabniyya* », as shown in Fig. 4.

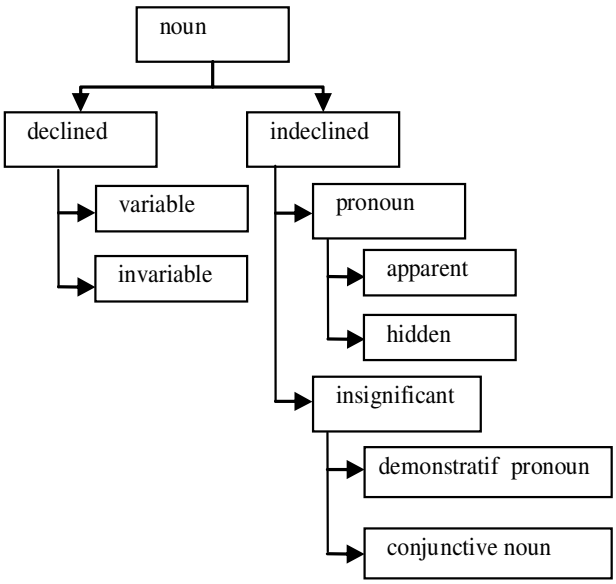


Fig. 4. Arabic noun's categories. A noun is declined when its ending varies according to its grammatical function in the sentence.

Fig. 4 shows that there are two categories of declined nouns: variable "متصرف, *mutaSarrif*" and invariable "غيرمتصرف, *ghayr mutaSarrif*". A noun is variable when it can be modulated (منوّن, *munawwn*) in a sentence for example (a man, رجلٌ). Moreover, a noun is invariable when it can not be modulated. This category covers essentially the proper names, "أسماء العلم, *asmaa al-'alam*" (simple or compound). Contrary to the indeclinable nouns, the declined ones can have several grammatical functions in an Arabic sentence.

In fact, indeclinable nouns remain invariable whatever their position and grammatical function in a sentence [2]. This category covers pronouns (الضمائر, *al-Damaair*) and insignificant nouns (الأسماء المهملة, *al-asmaa al-muhmala*). In fact, an insignificant noun has a meaning only when it is associated to another declined noun. Among this type of nouns, we can mention demonstrative pronouns "أسماء الإشارة, *asmaa al-ishaara* " and relative pronouns "الأسماء الموصولة, *al-asmaa al-mawSuwla*".

It should be noted that some constraints must be respected to compose different nominal phrases and some prepositional ones (reduction phrase, مركب الجر). Indeed, a nominal phrase is composed of two constituents: a noun representing the head daughter and another specifying the first one. There are various types of nominal phrases including annexation phrase 'مركب إضافي, *murakkab iDaafy* ', descriptive phrase 'مركب نعتي, *murakkab na'ty*' and substitution phrase 'مركب بدلي, *murakkab badaly*'.

(6) ولد الجار *The neighbor's son,*

(7) بيت الجار العجوز *The old neighbor's house,*

As represented in example (6), an annexation phrase is composed from two nouns. The first one is declined and undefined and the second one must be definite and reduced. Indeed, annexed compound can be a succession of nouns (Example 7).

Based on the proposed type hierarchy for Arabic language, it is necessary to add new criteria to specify an Arabic word. Besides, the studied sub-categorization of different linguistic words will be taken into account during the construction of an Arabic grammar. Thus, we have to establish appropriate HPSG representation for Arabic entries and syntactic phenomena.

IV. HPSG FOR THE ARABIC LANGUAGE

Head-driven Phrases Grammar Structure (HPSG) is a unification grammar proposed by Pollard and Sag [20]. It is considered among the best grammars to model universal grammatical principles and to give a complete representation of linguistic knowledge. In the following, we present an overview on HPSG and we propose some modifications to cover Arabic specificities.

A. Overview on HPSG

Contrary to other grammars, HPSG gives an importance to the lexicon. In fact, it represents not only the syntactic rules representing linguistic phenomena but also lexical entries with a very complete representation covering phonological, morphological, syntactic and semantic information. This allows taking into account a great number of linguistic phenomena and describing linguistic constructions with a limited number of operators. Indeed, HPSG formalism is based on the unification of AVM's. This operation modifies two structures to a common form.

1) HPSG components:

HPSG grammar is based on two essential components: a set of Attribute Value Matrix (AVM), to represent lexical entries and a set of Immediate Domination schemata (ID schemata), to describe syntactic phenomena.

An AVM is composed from a set of features. To each feature, a determined value is associated. Fig. 5, represents the general structure of an AVM:

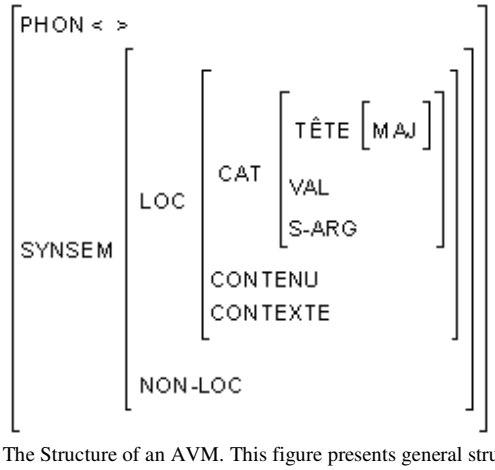


Fig. 5. The Structure of an AVM. This figure presents general structure of an AVM.

In an AVM, each feature represents a determined type of information. In fact, the feature PHON represents phonetic information and the feature SYNSEM collects syntactic and semantic information. This feature is subdivided in two features: LOC and NON-LOC. The first feature LOC covers others features such as TETE and VAL. It represents intrinsic information of the represented compound. In fact, characteristics describing the represented entry are gathered at the level of TETE feature and the compounds categorized by the represented entry are introduced in VAL feature. For the second feature NON-LOC, it describes the relation between the represented compound and other compounds.

For the ID schemata, HPSG grammar is based on six different schemata representing syntactic rules (i.e., specification rules). These rules are applied to compose various phrases. It should be noted that phrase composition requires a checking in a set of principles (i.e., HFP Head Feature Principle). In the following paragraph, we present the most important principles of HPSG grammar.

2) HPSG principles

In HPSG, feature propagation represents a fundamental mechanism. It describes syntactic relations between the different components. This task requires checking of some HPSG principles.

Among HPSG principles, we can mention Head Feature Principle, Valence, SPEC and Marker ones. *The Head Feature Principle* (HFP) identifies the HEAD value of any headed phrases with that of its HEAD-DTRS (Fig. 6).

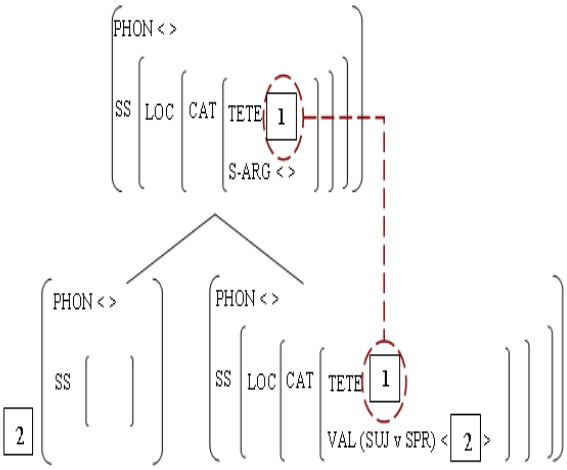


Fig. 6. Head Feature Principle. The HEAD of HEAD-DTRS, indexed 1, is similar to phrase head.

It must be noted *HFP* must be respected in the construction of all phrases.

The Valence Principle (VALP) requires that in each phrase the head daughter's relevant valence feature (COMPS, SUJ or SPR) specifies an element that is identified with the appropriate non-head daughter. In fact, this specification is mentioned at the level of VAL feature.

Another HPSG principle allows sharing of the marker daughter's SPEC value with the head daughter's SYNSEM value. It is *Specification Principle (SPECP).*

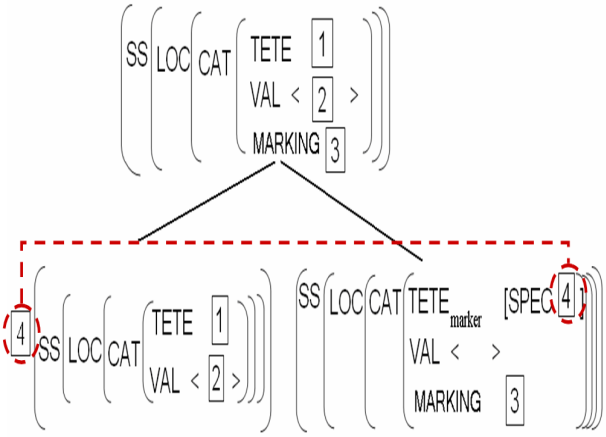


Fig. 7. Specification Principle. The MARKER-DTRS's SPEC value, indexed 4, represents the HEAD-DTRS's SYNSEM value.

As represented in Fig. 7, the HEAD-DTRS's SYNSEM value, indexed 4, is specified by MARKER-DTRS's SPEC value.

The Marking principle states that the HEAD-MARKER phrase takes its MARKING value from the marker daughter (In contrast to all other headed phrases, which share their head daughter's MARKING value) as shown in Fig. 8.

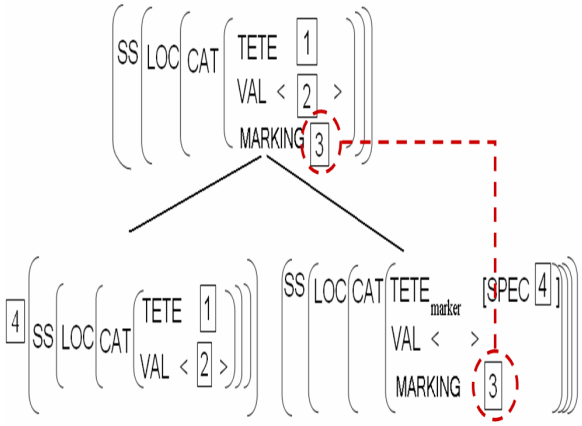


Fig. 8. Marking Principle. The MARKING feature of MARKER-DTRS, indexed 3, is similar to phrase head.

In Fig. 8, the MARKING feature is propagated to the head phrase using this principle.

All HPSG schemata are useful to represent Arabic phenomena. However we have to know how to use them and choose the adequate schema for each syntactic phenomenon. Moreover, features defined in HPSG are insufficient to represent Arabic entries. In fact, the values of some features differ in Arabic and others features must be added to represent other characteristics. In the following sections, we present Arabic language features and Arabic schemata.

B. *Arabic item features*

Referring to previous projects such as [1], [4], [11] and [18], we have kept some features and have added some others according to the proposed type's hierarchy.

As we have already seen, a linguistic sign (word or phrase) can be characterized by its declination (الإعراب, *al-i'raab*). Therefore a new feature called "DEC" is necessary to specify

if it is a declined sign (معرب, *mu'rab*) or not (غير معرب, *ghayr mu'rab*).

1) Features for verbs: According to Fig. 2, a triliteral or quadriliteral verb can be sound (سالم, *saalam*) or defective (معتل, *mu'tal*). Thus, features characterizing Arabic verbs are represented in Table1.

TABLE I
ARABIC VERB FEATURES

FEATURES	POSSIBLE VALUES	
RADICAL	- triliteral	ثلاثي
	- quadriliteral	رباعي
VFORM	- sound	صحيح
	- defective	معتل
TYPE	- intact	سالم
	- doubled	مضعف
VOICE	- Passive	مبني للمجهول
	- Active	مبني للمعلوم
ASPECT	- accomplished	ماضي
	- unaccomplished	مضارع
	-Imperative	أمر
ROOT	- the verb's root	(جذر)

Most of these features have been modified according to the proposed type's hierarchy.

The different features presented in Table I allows to represent adequately an Arabic verb. In fact, it covers all verbs' specificities and then reduces the ambiguity cases. Fig. 9 bellow is an example representing an Arabic verb.

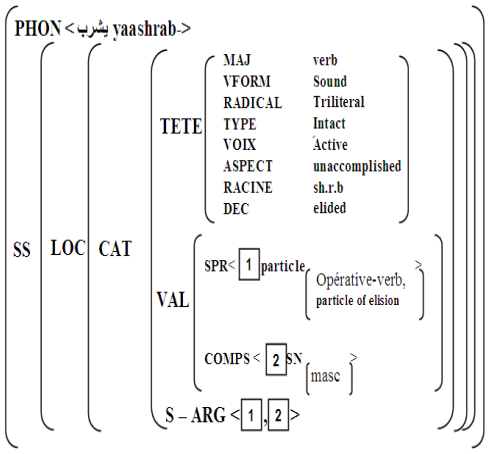


Fig. 9. An AVM modeling 'يشرب, yaashrab'. This figure presents an example of an Arabic verb. It can be accepted only when it is preceded by an elision particle.

As shown in Fig. 9, the verb 'يشربْ, yachrab-' has a complete representation. This verb is in an elided form. It indicates in the valence's feature different objects. In fact, an elided verb (مجزوم, *majzum*) must be preceded by an elision particle (حرف جزم, *harf jazm*), (referred by SPR feature) and followed by a masculine noun (referred by COMPS feature). Note that the order of these two components is respected by the S-ARG feature.

2) Features for nouns: According to Fig. 3, a declined noun can be variable (متصرف, *mutaSarrif*) as common nouns or invariable (غير متصرف, *ghayr mutaSarif*) as proper names (أسماء علم, *asmaa 'alam*). Indeclinable nouns are composed of personal pronouns (الضمائر, *al-Damaair*), conjunctive nouns (relative pronouns) (الأسماء الموصولة, *al-asmaa al-mawSuwla*) and demonstrative nouns (أسماء الإشارة, *asmaa al-ishaara*). Thus, features characterizing Arabic noun are represented in Table II.

TABLE II
ARABIC NOUN FEATURES

FEATURES	POSSIBLE VALUES	
NFORM	- Declined	معرب
	- Indeclinable	مبني
DEFINITE	- yes if it is defined	معرف
	- no otherwise	
NAT	- demonstrative nouns	اسم إشارة
	- conjunctive nouns	اسم موصول
ADJ	*- Yes if it can be an adjective*	
	- no otherwise	

Most of these features have been modified according to the proposed type's hierarchy as NFORM and NAT features.

In this context, conjunctive nouns are considered as insignificant indeclinable nouns which require the addition of some new features as summarized in Table III.

TABLE III
ARABIC CONJUNCTIVE NOUN FEATURES

FEATURES	POSSIBLE VALUES	
RFORM	- nominal	اسمي
	- prepositional	حرفي
RTYPE	- common	مشترك
	-specific	خاص

This table gathers various features characterizing Arabic conjunctive nouns. For example, RFORM feature distinguishes between nominal and propositional conjunctive nouns.

Fig. 10 represents an example of an AVM modeling an Arabic conjunctive noun and covering features described in Table III.

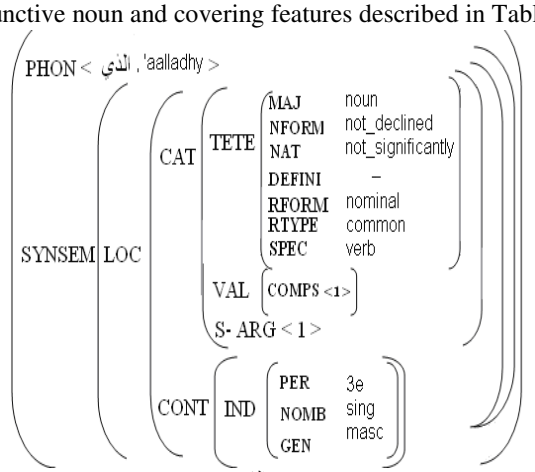


Fig. 10. An AVM modeling 'الذي, 'aalladhy'. This figure presents an example of a nominal conjunctive noun.

The conjunctive noun « الذي, ‘alladhy » is not a significant declined noun. This information is expressed by MAJ, NFORM and NAT features. Besides, INDEX feature shows that « الذي » is a singular masculine noun.

3) Features for particles: Arabic particle, presented in Fig. 4, can be categorized in operative and non operative particles. Thus, the features characterizing the particle type are presented in table IV.

TABLE IV
ARABIC PARTICLE FEATURES

FEATURES	POSSIBLE VALUES
PFORM	*- Non operative* مهمل
	- Operative عامل
NATP	*- elision particle* حرف جر
	- Subjunctive particle حرف نصب

NATP feature has an important function at the syntactic level. According to NATP value, we can specify the adjunct word (verb or noun).

As example, an AVM representing an elision particle (حرف جزم, *harf jazm*) is shown in Fig. 11.

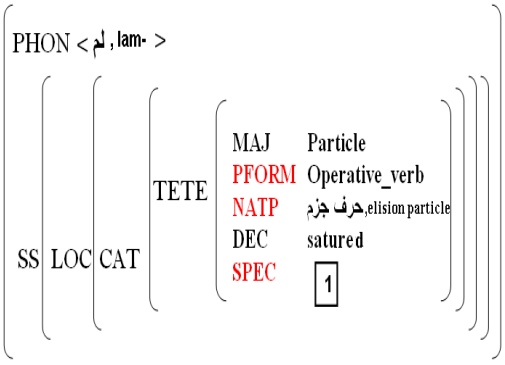


Fig. 11. An AVM modeling ‘لم, lam-’. This figure presents an example of an Arabic elision particle.

As represented in figure above, we defined different information characterizing this particle, at the level of TETE feature. Besides, the feature SPEC has an important role. It specifies an object indexed 1. In fact, according to features PFORM and NATP, the object must be an elided verb.

Modifications brought to HPSG formalism cover not only features but also different schemata of this grammar. In the following paragraph, we are going to present different modifications brought to ID schemata.

C. Arabic HPSG schemata

According to Abdelwahed and Dahdeh [2, 10], there are three types of Arabic phrases (nominal, prepositional and verbal) and two types of sentences (nominal and verbal). As we have mentioned previously, all HPSG schemata are useful to represent Arabic phenomena. In the following, we detail the exploitation of HPSG schemata in Arabic language.

1) Schemata representing specification rule:

Based on [7], we concluded that there is two schemata (schema 1 and 2) representing the specification rule. In fact, schema 1 was used to represent saturated phrases, where the HEAD-DTR sub-categorizes a specifier or a subject. In Arabic, we do not have the notion of subject. Therefore, we kept the schema 1 to represent compounds sub-categorizing only a specifier (SPR). In our work, we represented nominal phrases with this schema where the HEAD-DTR is a noun preceded by a demonstrative one. We present in Fig. 12, the HPSG representation of the sentence: ‘*This boy,* هذا الولد’. The construction of this sentence used schema 1.

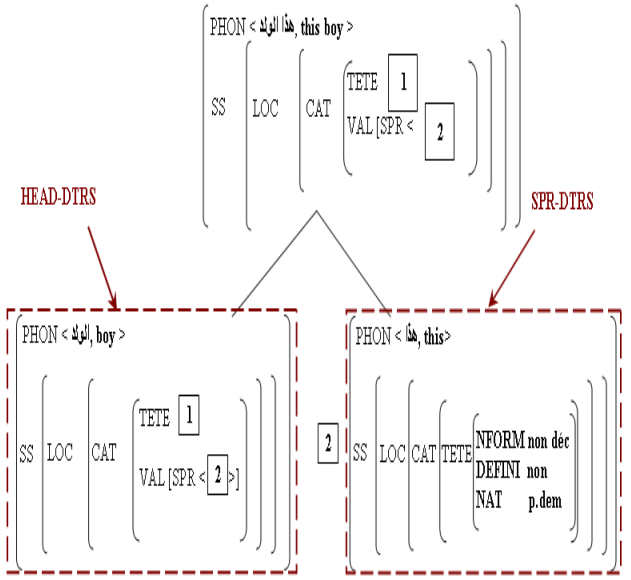


Fig. 12. An illustrative example of schema 1. The SPR-DTRS is specified at VAL feature level.

As represented in Fig. 12, the noun “الولد, boy” represents the head-daughter (HEAD-DTRS) of the nominal phrase “ هذا الولد, this boy”. It categorizes as specifier (Head-SPR) the demonstrative noun “هذا, this”, indexed 2.

For the schema 2 (rule of specification 2), according to [7], it represents verbal phrases (VP) where the verb categorizes a subject. In Arabic, we modified this schema to represent nominal sentences where the attribute categorizes a topic. Fig. 13 is a representation of the sentence "The boy is handsome, الولد جميلٌ" with this schema.

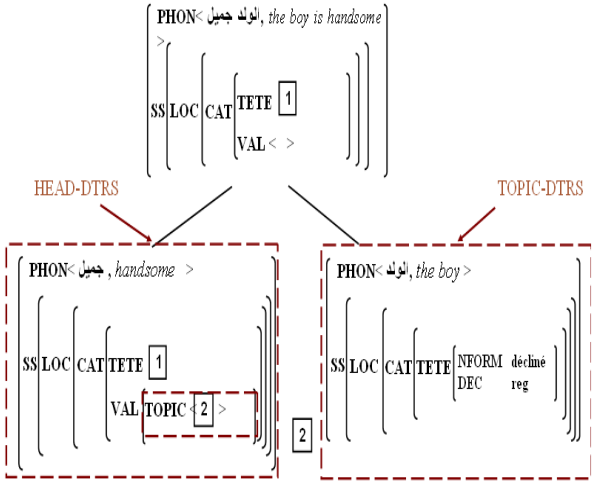


Fig. 13. An illustrative example of schema 2. The SPR of the HEAD-DTRS is the same one of the nominal phrase.

The declined noun "جميلٌ, handsome" represents the head-daughter (HEAD-DTRS) of the nominal sentence "The boy is handsome, الولد جميلٌ". It categorizes as topic (HEAD-TOPIC) the noun "الولد, the boy", indexed 2.

As represented in both Fig. 12 and Fig. 13, the HFP principle was respected. The HEAD value of the phrase is similar to the value of the HEAD-DTRS.

2) Schemata representing complementation rule:

According to [7], schema 3 represents a phrase where the HEAD-DTRS sub-categorizes one or several objects. Thus, we used this schema to model Arabic verbal sentences (VP +SUBJECT) or (VP +SUBJECT+COMPS). Besides, schema 3 represents various NP (i.e., annexed phrase, substitution phrase). Fig. 14 represents the annexed phrase 'ولد الجار, the neighbor's son'.

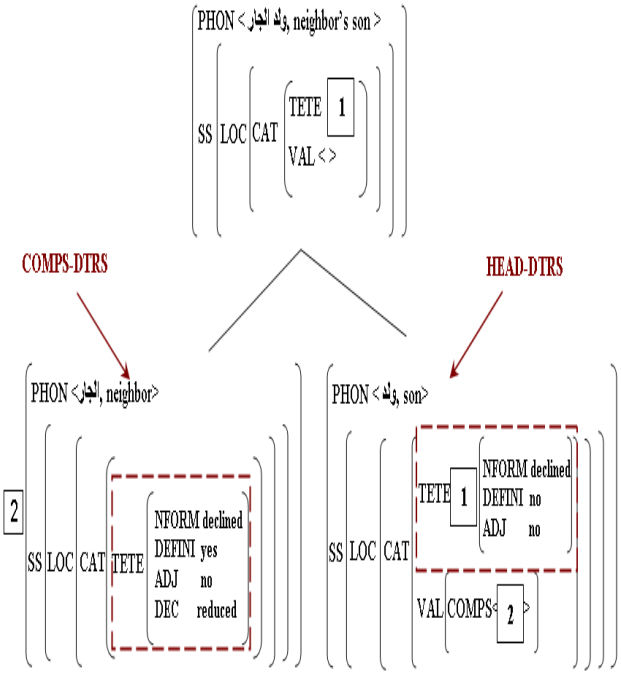


Fig. 14. An illustrative example of schema 3. The HEAD-DTRS categorizes an object indexed 2.

According to Fig. 14, the HEAD-DTRS sub-categorizes a defined declined noun 'الجار, neighbor'. This object must be reduced (مجرور, *majruwr*).

3) Schemata representing marking rule

Schema 4 takes into account the phenomenon of relatives. The HEAD-DTRS does not have an unlimited dependency during the propagation and the MARKER-DTRS has a MARKING feature. In fact, this schema used functional words. These words inherit from HEAD type to which we add a SPEC and a MARKING features. The SPEC feature allows an object to select the head type's with which it combines. The MARKING feature distinguishes words with or without marker. In fact, markers are associated with the SYNSEM | LOC | CAT| MARKING feature. Fig. 15 presents the relative phrase "who succeeded in the exam, الذي نجح في الامتحان".

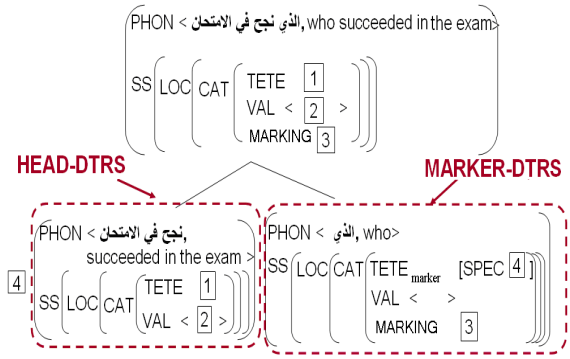


Fig. 15. An illustrative example of schema 4. The SPEC of the HEAD-DTRS specifies the MARKER-DTRS.

As represented in Fig. 15, the relative phrase "who succeeded in the exam, الذي نجح في الامتحان" has as marker the conjunctive noun "الذي, who". This last is followed by a verbal phrase "نجح في الامتحان, succeeded in the exam".

4) Schemata representing modification rule

Schema 5 represents the modification rule. It is very particular. In fact, the HEAD-DTRS is selected by the ADJUNCT-DTRS via MOD feature. This schema is used essentially for descriptive phrases. In Fig 16, we present the example: "فتاة جميلة, a pretty girl".

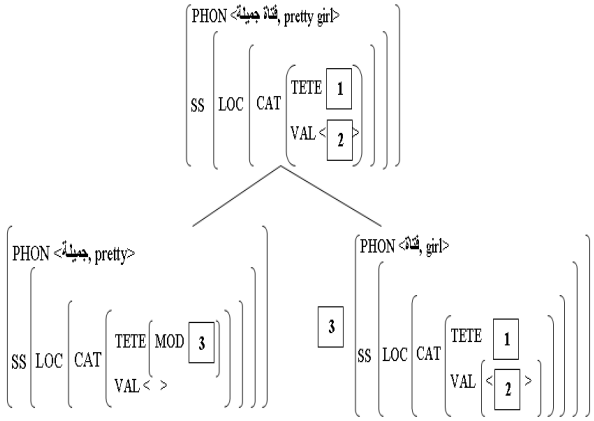


Fig. 16. An illustrative example of schema 5. The MOD of the ADJUNCT-DTRS selects the HEAD-DTRS.

The AVM containing MOD feature represents the ADJUNCT-DTRS 'جميلة, pretty'. According to Fig 16 above, the adjunct component selects the HEAD-DTRS indexed 3. This selection is associated in SS | LOC | CAT | HEAD | MOD feature of modifiers. In addition, the MOD feature has as values a SYNSEM structure.

Besides, modification rule allows some conjunctive nouns to select the modified category. Consequently, some conjunctive nouns are considered at the same time as modifiers and specifiers [6].

According to the proposed modifications for HPSG features and ID schemata, we specify an Arabic HPSG. This grammar is experimented on LKB platform and specified in Type Description Language (TDL). In the following paragraph, we start by an overview on the TDL syntax. Then we give an idea about the grammar's specification.

V. SPECIFICATION OF THE HPSG GRAMMAR IN TDL

TDL language is designed to support highly lexicalized grammar theories like HPSG. Work on TDL has started within DISCO project of the DFKI [15]. In the following sections, we give an overview of TDL. After that, we present TDL specification of linguistic resources composing the constructed grammar (i.e., type hierarchy, lexicon, syntactic rules).

A. Overview on TDL language

TDL language is considered the most adequate to specify HPSG formalism. Indeed, there exist a great similarity between TDL syntax and HPSG representation as shown in Table V.

TABLE V
IDEA ON TDL SYNTAX

Operator	Function
&	The constraints addition allows on types.
# [a..z]	For structures indexation and labeling.
;	For comments addition on the same line.
#|…|#	For comments addition of several lines.
: =	Element on the left is defined like constraints by element on the right.
[ ]	To define a feature structure: Attribute Value Matrix (AVM).
< >	To define a list.
,	To separate attribute-value couples in a AVM.
.	To indicate the end of structure totality: end of type description. Also equivalent of [ ].

This Table explains the meanings of same symbols used in TDL syntax.

According to Krieger and Schafer [15], TDL define different types based on lists notion. In TDL, lists are represented as first-rest structures with distinguished attributes FIRST and REST, where the sort *null* at the last REST attribute indicates the end of a list (and, of course, the empty list). The input of lists can be abbreviated by using the < > syntax as follows:

```
*diff-list* := chaine &
        [LIST *liste*,
         LAST *liste*].
dlist-phon := *diff-list*.
dlist-ind := *diff-list*.
```

Moreover, there are other types of lists: difference lists. They are first-rest structures with distinguished attributes FIRST, and a special LAST attribute at the top level, which shares the value with the last REST. In TDL, the elements of difference lists may be enclosed in <! !>.

Since, features differ from an AVM to another referring to the word's type; we have to specify the proposed hierarchy, the lexicon and the different syntactic rules. In the following, we present TDL specification of these linguistic resources.

B. TDL specification of type hierarchy

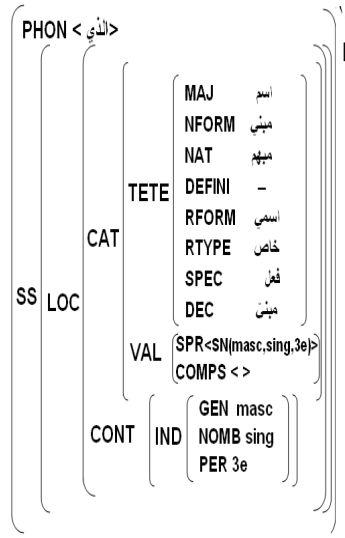
Types can be arranged hierarchically where subtype inherits all information from its super types. This leads to multiple inheritances in the description of linguistic entities. In addition, recursive types are necessary to describe at least phrase structure recursion. Note that, recursion is based on difference lists. Below, we present an extract from "type.tdl" file containing TDL specification of proposed type hierarchy:

```
signe := *top* &
    [PHON dlist-phon,
     SS synsem-canon].
tete := valeur &
   [MAJ string, DEC dec].
dec := valeur.
 ouverte := dec.
 reduite := dec.
 elidee := dec.
verbe := tete &
     [MAJ "verbe",
      RADICAL radical,
      VFORM vform,
      TYPE type,
      RACINE string,
      ASPECT aspect,
      VOIX voix].
```

In the code represented above, type's definition is based on the inheritance notion. For example, to represent the verb type "*verbe*", some constraints (i.e., *RADICAL*, *VFORM*) are introduced at the level of verb type and others are inherited from subtypes. Indeed, verb type inherits DEC feature from the subtype *tete*.

C. TDL specification of lexicon

As we have mentioned previously, HPSG represents lexical entries with AVM structures. This representation is based also on multiple inheritances. Fig. 17 shows the AVM "who", (الذي) as well as its TDL specification:




```
'aalladhy := lex-nom-non-decline &
[PHON <! "'aalladhy" !>,
 SS [LOC
 [CAT.
 TETE[
   MAJ noun,
   NFORM indeclined ,
   NAT  insignificant  ,
   DEFINI not ,
   RFORM nominal,
   RTYPE specific,
   SPEC < SS.LOC.TETE verb  >,
     DECsatured],
  VAL [SPR <SS.LOC.TETE noun >,
       COMPS < [ ] >],
 CONT [ GEN masc,
        NOMB sing,
        PER 3e]]].
```

Fig. 17. Implementation TDL of "الذي". This figure shows the great similarities between HPSG and TDL syntax.

In Fig. 17, we conclude that it is very simple to specify in TDL a HPSG representation. In fact, the symbol ":=" means that this entry represents an instance of indeclinable nouns. Constraints are added by the symbol &. Different features composing an AVM are separated in HPSG with a set of matrixes. In TDL, we have to replace them by simple brackets. The various attribute-value couples are separated by comas and the full stop designates the AVM end. The different lexical entries are specified in a TDL file: "lexique.tdl".

As shown in the figure above, the lexical entry "الذي, *'aalladhy*" is an instance of the type "*lex-nom-non-decline*". In the following, we present TDL specification of this type.

```
lex-nom-non-decline := lex-nom &
  [SS [LOC [CAT [TETE relatif &
                    [ NAT pronom_relatif ,
                      ADJ non,
                      MOD [LOC [CAT [TETE nom]],
                               NONLOC [REL #rel,
                                       SLASH #slash]],
                      SPEC.LOC.CAT.TETE tete_mot],
                    VAL[COMPS < >]]],
       NONLOC [REL #rel,
               SLASH #slash & <! !>]]].
```

Note that, the type "*lex-nom-non-decline*" is specified in "type-lex.tdl" file which introduces a set of constraints that must be specified for each type of words. In fact, features representing a lexical entry are defined in the "type.tdl" file and constraints were defined in "type-lex.tdl".

D. TDL specification of syntactic rules

For different schemata mentioned in part V, they are specified in another TDL file: "rsynt.tdl". In fact, in this file we have specified different Arabic phrases and so various structures of Arabic sentences.

According to [2], we studied different phrases (NP, VP or PP). Each type of phrase has some constraints to take into account. For example, annexation phrase "المركب الإضافي,*al-murakkab al-iDaafy* " has different forms. We give below some examples of this NP type:

(8) *The neighbor's house,* بيت الجار
(9) *The old's child,* صغير العجوز
(10) *Between lines,* بين الأسطر
(11) *His child,* ولده

As represented in these sentences, the annex "مضاف, *muDaaf*" can be a variable (descriptive or not) or an invariable noun. In fact, in sentence (8) and (11) the annex is a variable noun. Contrary, in (10) it is an insignificant noun. Moreover, the annexed noun can be an attached pronoun (sentence 11). All these constraints are taken into account in the TDL specification of the complementation rule. We give above, an extract of the "rsynt.tdl" file:

```
regle_annexion := regle-bin-t-init &
[SS [LOC [CAT [tete tete-annexant ,
          VAL [COMPS <#nontete >]]]],
 BRS [BR-TETE [SS [LOC [CAT [tete tete-annexant
                             & [DEC dec_simple],
                      VAL [SPR< >, COMPS <#nontete >]]]]],
      BRS-NTETE < [SS #nontete
                   & [LOC [CAT [TETE tete-annexe
                                & [DECred]]]] >]].
```

This TDL specification shows that this rule is an instance of "*regle-bin-t-int*". In fact, this type of rule composes binary phrases where Head-DTRS is in the beginning of phrase. It should be noted that rules types are specified in the TDL file "*types-regles.tdl*". Besides, "BRS" represents the two phrase compounds: BR-TETE (HEAD-DTRS, tete-annexant) and BRS-NTETE (tete-annexe). In fact, these two types were specified in "*types.tdl*". The first type regroups annex's possible forms. The second one regroups all annexed forms.

In the same way, based on [2] we have specified different structures of verbal phrases (VP). In fact, this type of phrase is composed of a particle and a verb. Each particle must be associated to a very determined verb. To detail this idea, we give in the following some VP examples:

(12) *He didn't slept,* لم ينم
(13) *He didn't slept,* ما نام

As we can conclude from these sentences, an elision particle must be associated to an elided verb (sentence 12). Contrary to sentence (13), we note that this type of particle must be associated with an accomplished verb. In the following, we present TDL specification of the prepositional rule which represents verbal phrases:

```
regle_specification_3 := regle-bin-t-fin &
[SS [LOC [CAT [TETE verbe,
          VAL.SPR < [LOC.CAT.TETE particule]>]]],
BRS [BR-TETE [SS #tete & [LOC [CAT [TETE verbe
                                    & [DEC dec_sv],
                             VAL [COMPS < >]]]]],
     BRS-NTETE < [SS [LOC [CAT [TETE particule
                                & [SPEC #tete]]]] >]].
```

Besides, we have specified nominal and verbal sentences for Arabic language having the following structures:

- Nominal sentences: (NP + NP), (NP + VP) or (NP + PP).
- Verbal sentences: (VP + NP), (VP + NP + COMPS) where COMPS can be NP or PP.

In fact, according to [2], an Arabic verb can be transitive or intransitive. For the transitive verbs, we specified a binary syntactic rule where the HEAD-DTRS is a VP and the object is a regular noun. This rule represents verbal sentences compound from a VP and a subject. Its TDL specification is presented below:

```
regle_specification_2_S:= regle-bin-t-init &
[SS [LOC [CAT [TETE verbe & [DEC init],
                VAL [COMPS <#nontete >]]],
BRS [BR-TETE [SS [LOC [CAT [TETE verbe
                                & [DEC init],
                          VAL [COMPS <#nontete >]],
                         CONT.IND.GEN #ind]]],
     BRS-NTETE < [SS #nontete &
          [LOC [CAT [TETE nom & [DEC reguliere],
                     VAL [COMPS < >]],
                CONT.IND.GEN #ind]]>]].
```

In fact, this TDL specification represents verbal sentences that can start with an intransitive verb. This type of verb has one object representing the sentence subject.

For the second type of verbs (transitive ones), we have specified another ternary rule which represents Arabic sentences having the following structure: VP + NP + COMPS. Since objects number is undefined, we have specified another rule regrouping verb's objects.

The specified linguistic resources (proposed type hierarchy, lexicon and syntactic rules) are used as an input to LKB platform in order to experiment the constructed HPSG grammar. In the next paragraph, we give an idea about LKB platform. Then, we experiment and evaluate the established Arabic grammar.

VI. Experimentation and Evaluation

Linguistic Knowledge Building (LKB) system is a generation tool, proposed by [9]. It is based on two types of files: TDL files and LISP files. The first type represents the grammar's files. In fact, this grammar is based on seven TDL files: lexicon, type, type-lex, type-rules, rsynt, noeuds and roots. The file "nœuds.tdl" allows labels specification to be posted during the parsing. The file "roots.tdl" delimits the structure to be analyzed by the parser. The other files are detailed later.

The second type represents files to parameterize LKB system. It is based on five LISP files. Among these files, we can especially mention the file: "script.lsp'. It is a very important file. In fact, it indicates the name and the repertory of each grammar file.

It should be noted that there exist several versions of LKB system. In our work, we have used windows version. In the following paragraph, we describe the stages of syntactic analysis. Then, we present an experimentation of the established grammar.

A. *Stages of syntactic analysis*

To analyze the constructed HPSG grammar, we have to load it on LKB platform by giving the path of "script.lsp" file. Thus, the LKB system compiles different grammar files. If there isn't any error message, a parser is generated. In fact, LKB offers two different types of analysis: parsing a simple sentence or a corpus of sentences.

To start the analysis stage, the generated parser segments the tested sentence. Then, it checks the existence of all entries in the lexical database *"lexique.tdl"*. Once this phase is completed successfully, a verification of the compatibility between lexical constraints with those of syntactic rules is done. After that, the parser analyzes syntactic relations and assigns labels for lexical entries and built phrases. It should be noted that the obtained result is represented as a derivation tree like in Fig 18.

(14) الولد الذي شرب الماء نام

'alwaladu 'alladhy chariba 'almaa naama

The child who drank the water has slept

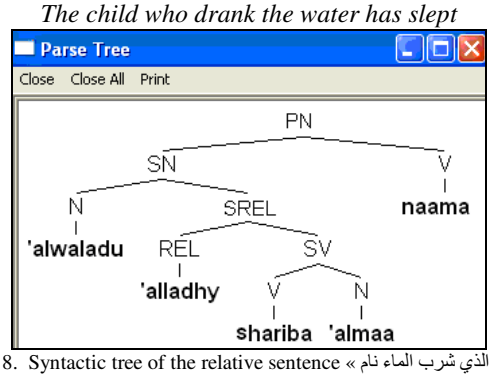


Fig. 18. Syntactic tree of the relative sentence « الولد الذي شرب الماء نام ».

Relative sentences can contain other type of phrases (i.e., prepositional phrase, verbal phrase). Sentence (14) includes a special nominal conjunctive noun « الذي, *'alladhy*» associated to the verbal phrase (VP) «شرب الماء, *shariba 'almaa* ».

As we have mentioned, the generated parser can experiment the constructed grammar on a corpus of sentences. Since, the LKB system (version system) does not support Arabic letters and lacks a fragmentation module, the tested corpus must be fragmented in transliterated sentences. Therefore, we have to present two files. The first file (corpus.txt) contains sentences composing the corpus. The second file (results.txt) covers the obtained results. In fact, in this file, LKB presents the number of tree's derivation and the number of nodes in the creation graph of the derivation's tree as shown in Fig. 19.

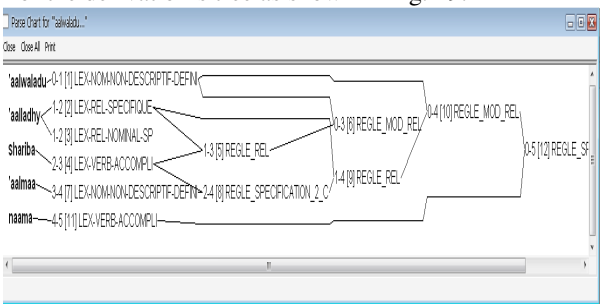


Fig. 19. Result of parsing the sentence: "the boy who dinks the water". This figure presents the nodes number and derivation's tree.

As we can see in Fig. 19, we conclude that 12 nodes are required to the creation graph of derivation's tree. In the following paragraph, we discussed the obtained results.

B. *Evaluation*

The evaluation of the constructed grammar is based on a

corpus of 500 sentences containing essentially relatives. Besides, the test corpus contains other linguistic phenomena such as the elision «الجزم, *al-jazm*», the call «النداء, *al-nidaa*», the description «النعت, *al-na't*». The used lexicon contains approximately 3000 words (~2500 verbs, 450 nouns and 50 particles). It is formed mainly of the corpus words.

Table VI below describes the obtained results. In fact, it recapitulates the distribution of derivation's tree number in the test corpus.

TABLE VI
OBTAINED RESULTS

Number of derivation's tree (n)	Number of sentences having n analyses
0	8
1	420
2	61
>=3	11
Total	500

This Table summarizes the number of sentences having n derivation's trees.

For the tested sentences, we note that the generated parser could correctly build their syntactic structures in a reasonable time. In addition, Table VI shows that 2% of the sentences do not produce derivation trees, 84% of sentences have only one analysis and 14% have at least two derivation trees.

For the remaining sentences, the failure is due to the existence of more than one derivation tree for the same sentence. In fact, this problem was encountered in previous works using LKB system such as [13] and [16]. In our work, we introduced other constraints more specific, to resolve the encountered problem according to the proposed type hierarchy. Nevertheless, ambiguous cases persist. This is caused mainly by ambiguities found during relative sentences analysis. Fig. 20 represents an example of sentence covering ambiguous cases.

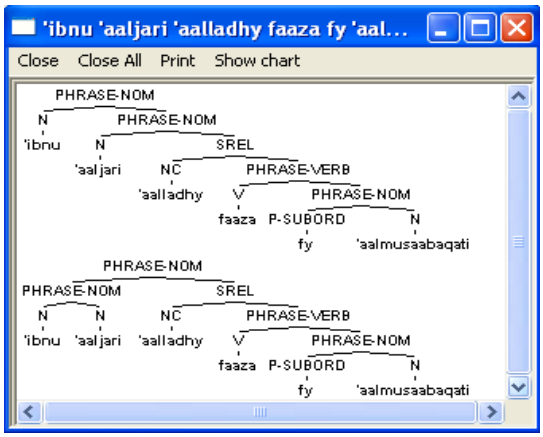


Fig. 20. Result of parsing the sentence: "The son of neighbor who gained in tournament". This figure presents ambiguous cases for this sentence.

Indeed, the relative phrase "الذي فاز في المسابقة, *'alladhy faaza fy al-musaabakati*" can refer to the noun "الجار, *al-jaari*" or to the nominal group "ابن الجار, *ibnu al-jaari*". This nominal group represents an annexed phrase.

Besides, there is another problem at the level of lexicon. This problem was encountered also in previous projects working on Arabic language such as [3], [5] and [11]. In our work, we have added an interface written in JAVA which can enrich the file "*lexique.tdl*" by new words automatically and without knowing TDL syntax. Moreover, this lexicon can easy be extended using tools that we have developed in our laboratory like the translator from LMF toward TDL [12].

VII. CONCLUSION AND PERSPECTIVES

In this paper, we have constructed an HPSG grammar for Arabic language treating particularly relative sentences. For this reason, we have proposed a type hierarchy categorizing Arabic words in different types. According to the proposed type hierarchy, we brought some modifications to HPSG grammar in order to treat Arabic specificities. The constructed grammar was experimented on LKB platform. Therefore, we specified Arabic HPSG with TDL language. This TDL specification is original, in our work since it integrates some operations and verifies certain concepts such as inheritance, adjunction and recursion. The evaluation phase shows that obtained results are satisfactory.

As perspectives of this work, we aim to test our parser on a larger corpus. We plan also to extend the HPSG description to cover other linguistic phenomena. Also, we plan to extend this work to cover semantic analysis. However, more works should be carried out to transform the system written under Windows into a compatible system under UNIX.

APPENDIX

Since LKB Windows version does not support the Arabic letters, we have also implemented a proper transliteration tool based on the used morphological transliteration Qalam system. In fact, "Qalam" is the transliteration developed by A. Heddaya in contribution with W. Hamdy and Mr. H. Sherif, (1985-1992).

TABLE VI
SOME ELECTRONIC TRANSLITERATIONS

Letter	Name	Qalam
أ	ALEF	aa
ب	BEH	b
ت	TEH	t
ث	THEH	th
ج	JEEM	j
ح	HAH	h
خ	KHAH	kh
د	DAL	d
ذ	THAL	dh
ر	REH	r
ز	ZAIN	z
س	SEEN	s
ش	SHEEN	Sh
ص	SAD	S
ض	DAD	D
ط	TAH	T
ظ	ZAH	Z
ع	AIN	`
غ	GHAIN	Gh
ة	TEH MARBUTA	t ou h
و	WAW	W
ي	YEH	Y
ى	ALEF MAKSURA	Ae
◌َ	FATHA	A
◌ُ	DAMMA	U
◌ِ	KASRA	I
◌ً	FATHATAN	aN
◌ٌ	DAMMATAN	uN
	KASRATAN	iN
	SHADDA	[a]
◌ْ	SUKUN	-
◌ٔ	HAMZA ON LINE	´
أ	HAMZA ON ALEF	
إ	HAMZA UNDER ALEF	
ؤ	HAMZA ON WAW	
ئ	HAMZA ON YEH	
آ	MADDA ON ALEF	~aa
ٱ	WASLA ON ALEF	E

REFERENCES

[1] A. Abdelkader, K. Haddar and A. Ben Hamadou, « Etude et analyse de la phrase nominale arabe en HPSG », Traitement Automatique des Langue Naturelles, Louvain, UCL Presses de Louvain: 379-388, 2006.

[2] A. Abdelwahed, « ‘alkalima fy ‘attourath ‘allisaany ‘alaraby , الكلمة في التراث اللساني العربي », Librairie Aladin 1ère édition, Sfax – Tunisie : 1-100, 2004.

[3] S. Alnajem and F. Alzhouri, “An HPSG Approach to Arabic Nominal Sentences”, Journal of the American society for information Science and Technology: 422 – 434, 2008.

[4] C. Aloulou, « Analyse syntaxique de l’Arabe: Le système MASPAR », RECITAL, Nantes – France, 2003.

[5] Y. Bahou, L. Hadrich Belguith, C. Aloulou and A. Ben Hamedou. «Adaptation and implementation of HPSG grammars to parse non-voweled Arabic texts », memory of Master, Faculty of Economics and Management of Sfax.

[6] C. Belkacemi, «The relative marker: a definite marker substitute? », ArOr Archiv Orientální, 66/2, 142-148, Based on Arabic dialects, 1998.

[7] P. Blache, «Les Grammaires de Propriétés: des contraintes pour le traitement automatique des langues naturelles». Hermès Sciences, Paris, 2001.

[8] S. Boukedi, K. Haddar and A. Abdelwahed, « Vers une analyse des phrases arabes en HPSG et LKB ». GEI 2008, 8ème Journées Scientifiques des Jeunes Chercheurs en Génie Electrique et Informatique, Sousse, Tunisie : 487- 498, 2008.

[9] A. Copestake, « Implementing Typed Feature Structure Grammars ». CSLI Publications, Stanford University, 2002.

[10] A. Dahdah. « معجم قواعد اللغة العربية في جداول و لوحات », Librairie de Nachirun Lebanon, 5ème edition, 1992.

[11] S. Elleuch, « Analyse syntaxique de la langue arabe basée sur le formalisme d’unification HPSG ». Mémoire de DEA en Système d’information et Nouvelles Technologies, Sfax, Tunisie : 55-88, 2004.

[12] H. Fehri, N. Loukil, K. Haddar and A. Ben Hamadou, “Un système de projection du HPSG arabisé vers la plate-forme LMF ». JETALA, Rabat Maroc, 1-11, 2006.

[13] O. Garcia, « Une introduction à l’implémentation des relatives de l’espagnol en HPSG–LKB », Mémoire de recherche, 2005.

[14] K. Haddar and A. Ben Hamadou, « Un système de recouvrement des ellipses de la langue arabe ». Proceedings of VEXTAL, San Servolo V.I.U. 22(11) : 159-167, 1999.

[15] H. Krieger and U. Schäfer, « TDL: A Type Description Language for HPSG». Part 1 and Part 2, Research Report, RR-94-37, 1994.

[16] F. Laurens, « Implémentation des types de phrases et des types de constructions coordonnées du français avec la platefiorme LKB », Stage réalisé au sein du laboratoire LLF sous la direction de A. Abeille, 2007.

[17] M. Loukam and M. Laskri, « Vers la modélisation de la grammaire de l’arabe standard basée sur le formalisme HPSG », Actes JED’2007, Journées de l’Ecole Doctorale, 27(5), Annaba/Algérie, 2007.

[18] H. Maaloul, K. Haddar and A. Ben Hamadou, «La coordination arabe : étude et analyse en HPSG », MCSEAI 2004, 8ème conférence maghrébine sur le GL et l’IA, Sousse, Tunisie : 487- 498, 2004.

[19] W. D. Meurers, «A Web-based Instructional Platform for Constraint-Based Grammar Formalisms and Parsing». In Dragomir Radev and Chris Brew (eds.), Effective Tools and Methodologies for Teaching NLP and CL, New Brunswick, NJ: The Association for Computational Linguistics: 18 – 25, 2002.

[20] C. Pollard and I. Sag, «Head-drive phrase structure grammars», CSLI series, Chicago University Press, 1994.

[21] I. Shariful and R. Ahmed, “An HPSG Analysis of Arabic Verb”, Proceedings of the 9th International Arab conference on Information Technology (ACIT’08), 2008.

[22] J. Tseng, « Implémentation HPSG avec LKB: La Matrix et la Grenouille », Séminaire HPSG-UFRL, Paris 7, 14(12), 2006.