COMPUTER AND INFORMATION SCIENCES DEPARTMENT
INSTITUTE OF STATISTICAL STUDIES AND RESEARCH
(ISSR)
CAIRO UNIVERSITY

Developing an Arabic Parser in a Multilingual Machine Translation System

M. Sc. Thesis

Submitted by

Ahmed Farouk Ahmed

Supervised by

Prof. Dr. Ahmed Rafea
Vice Dean, Faculty of Computers and Information, Cairo University

Dr. Khaled Shaalan
Computer Science Dept., Faculty of Computers and Information, Cairo University

A thesis submitted to the Department of Computer Science, Institute of Statistical Studies and Research, Cairo University, in partial fulfillment of the requirements for the degree of Master in Computer Science.

1999

ACKNOWLEDGMENT

I would like to express my deepest gratitude to Prof. Dr. Ahmed Rafea. It was only through his meticulous supervision, support, and constructive advice, that this work has been completed. I would like to extend my gratitude to Dr. Khaled Shaalan. I am greatly indebted to him for his valuable input and his continuos support and direction. Without his generosity with his time and effort, this work would not have attained its present form.

Thanks also to my linguistics friends Ahmed Mostafa, Mohamed Kasem, Mahmoud Saad and Abdel Hameed Sakr whose gave me their linguistics experience in this work.

Special thanks for the Eng. Mohamed Talaat Helal for his great effort and support to finalized this thesis.

Thanks are due to the entire members of Institute of statistical Studies and Research (ISSR), offering me the facilities and support that were needed for my research.

This list of persons could be extended much further, therefore I would like to thank all those not mentioned above, who have helped and supported me.

ABSTRACT

Parsing Arabic sentences is a difficult task. The difficulty comes from several sources. One is that sentences are long and complex. The average length of a sentence is 20 to 30 words, and it often exceeds 100 words. Another difficulty comes from the sentence structure. The Arabic sentence is complex and syntactically ambiguous due to the frequent usage of grammatical relations, order of words and phrases, conjunctions, and other constructions.

The present work reports our attempt in developing an Arabic Parser for modern scientific text. The parser is written in Definite Clause Grammar (DCG) and is targeted to be part of a machine translation system. The developing of the parser was a two-step process. In the first step, we acquired the rules that constitute a grammar for Arabic that gives a precise account of what it is for a sentence to be grammatical. The grammar covers a text from the domain of the agricultural extension documents. The second step was to implement the parser that assigns grammatical structure onto input sentence. Experiment on real extension document was performed. The thesis will also describe our experience with the developed parser and results of its application on a real agricultural extension document.

TABLE OF CONTENTS

.

CHAPTER (1)
INTRODUCTION

This chapter will briefly sketch some background on natural language processing. Then the focus turns to a description of the scope of the current work. Next, *the aim of the current work and the structure of this thesis are summarized*

1.1 Natural Language Processing

Natural language processing (NLP) can be defined, in a very general way, as the discipline having as its ultimate, very ambitious goal that of enabling people to interact with machines using their "natural" faculties and skills. The means, in practice, that machines should be able to understand spoken or written sentences constructed according to the rules of same natural language (NL), and should be capable of generating in reply meaningful sentences in this language. [Zarri, 1998].

In the last few years, the role of computers in natural language processing has increased so much that it will not be to long before linguistic processing by computers is an absolute necessity. This is already true in areas such as word processing and cryptography but there are many other areas that are just emerging that will change completely the way we think about computers and language. Some of the factors that have contributed to the fast growth of this field are the lower cost of computers and their increased capabilities. [Zamora, 1994].

The developer of natural language processing applications faces many problems for which traditional data processing techniques do not provide adequate methodology. In traditional data processing tasks, such as inventory control and accounting, one role of the system is to make sure that the final computer programs preserve the integrity of the data. These applications may be viewed as models of the real world, where money or goods are being represented internally by the computer.

1.1.1 The Standard Paradigm for Natural Language Processing

This section describes the five major aspects of natural language systems [Zarri,1998], [Zamora, 1994].

Lexicon

A lexicon is a dictionary containing the words that are recognized by a natural language system. It is necessary to define what a word is before a lexicon can be constructed. The technique for isolating words affects the content of the dictionary and the applications that use the dictionary. Should the dictionary include contractions like “can’t” or hyphenated words like “mother-in-low”? If capitalized words are allowed? Will the dictionary contain multiple words such as “ hot dog “ or abbreviations like “etc.”?

Having isolated the words in the input sequence, the first step of the analysis concerns the computation of their grammatical category (part of speech “tag”), i.e., the association with information like “noun”, ”verb” adjective” etc this particular information will then be used in all the subsequent phases of the procedure.

Morphology

Recall that some aspects of the lexicon can be expanded by morphological analysis procedures. This consists of examining how words are built up of more basic meaning units called “morphemes” (“root forms”), where a word may consist of a root form plus additional affixes. For example a word like “familiar “ can be considered as a “root form”, From this adjective, we can derive:

- “unfamiliar”—adjective— By adding pripex <un>
- “ familiarity” – uncount— By adding suffix <-ity>
- “ familiarly” – adverb— Be adding suffix <-ly>

The reliability of morphological procedures can be enhanced by introducing specialized dictionaries that contain exceptions to the morphological rules. Assume, for example, that it is necessary to determine whether an English word is the past tense of a verb. This can be accomplished by first looking for the word in a list of exceptions and then checking to see if the word ends in I(-ed).The list of exceptions needs to contain irregular verb forms

such as (took) and (wrote) and other words with (-ed) endings that are not past-tense verb forms.

Syntax

Syntax is the branch of linguistic that deals with the formation of phrases, clauses, and sentences. Even if the words of the input are now endowed with a “tag” (part of speech category), they still represent as more than a simple, flat string of words. To begin understanding the “meaning “ carried on by statement , it is necessary to make clear first its” structure” i.e., to determine , e.g., the subject and the object of each verb , to understand what modifying words modify what other words, what words are of primary importance, etc. Assigning such structure to an input statement is called “syntactic analysis “ or “parsing “.

A natural language parser is a computer program that attempts to identify the grammatical role of the words of a sentence. A parser is then a sort of transducer that takes as input a plat string of words (already linked with their part of speech tags) and produces as output a structure (e.g., a “syntactic tree”) where the mutual relationships of the words in the statement are indicated.

Semantics

Having determined the structural properties of an natural language statement, sentence, or utterance, thanks to the use of the morphological and syntactic tools, we should now utilize the final syntactic structure and the semantic categories associated with the individual words to construct the overall “ meaning “ of the sentence.

We can say that, semantics is the study of the relationship between symbols and their meaning. In reality, both the aims and proper methods of the semantic phase are not very well defined, and this phase is much more fuzzy and problematic than the previous syntactic phase. The central problem concerns, of course, the fact that it is very difficult to find agreement on what the “meaning “ of a statement can be.

Natural language contains many idioms and metaphors that add color to the language but also create ambiguities. An idiom such as “on the other hand “, for example, is used to indicate an alternative and does not have anything to do with “hand”, except in special cases.

Distinguishing these special cases is not easy. Metaphors such as “ time flies like an arrow” would make one think that “time” is a type of winged creature, if one did not know better.

Scientists, mathematicians, and lawyers attempt to develop precise terminology to achieve universal clarity.

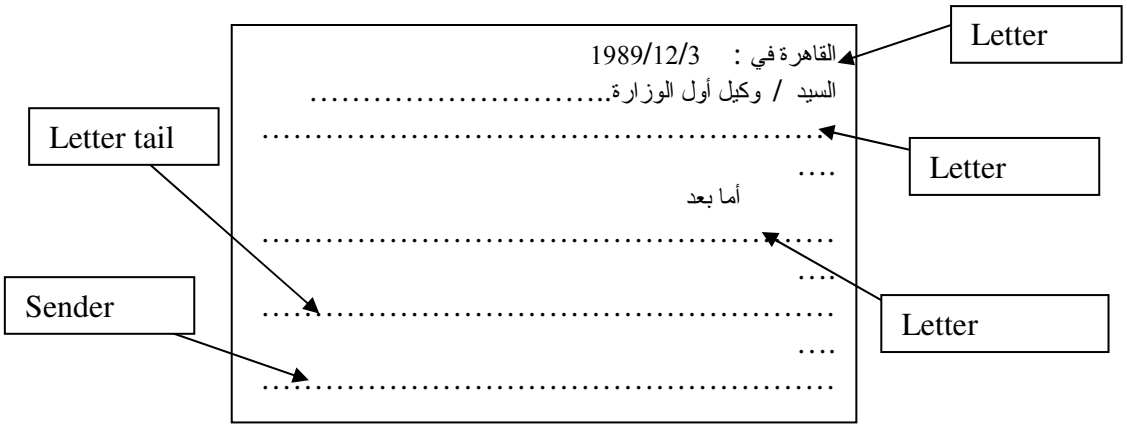
1.2 Scope and Aim of the Work

There are many templates for writing a text in the same language. It is very difficult to develop one parser to exhaustively recognize all such templates. So, it is very important to classify the language text to separate templates and, then, develop separate parser for each of which. The following are the important templates of the natural language:

(1) Letter template:

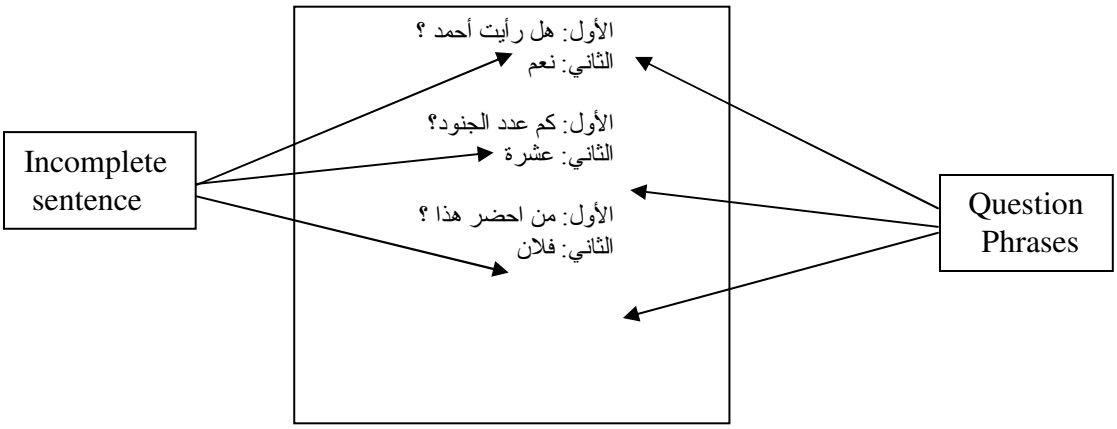
This template has some special structure like the following: letter date, letter header, letter, connectors, letter tail, and sender name

The parser of this template need to be aware from these special phrases and must be set special grammar rules to parse it.



(2) Dialog template:

This template expresses a text that represents the dialog between more than one person. This template is not a predicate sentence (جملة خبرية), but it represents an interactive sentence like: question phrase and Incomplete sentence



(3) Literature template:

This template includes many special forms (أسلوب) which need special grammar, the following table shows some examples:

Name of Special	***Example***
أسلوب الاستغاثة	‘‘ يالله لمنكوبى الحرب’’ ، ‘‘واإسلاماه’’
أسلوب الإغراء	‘‘البر البر بالوالدين’’
النداء	‘‘يا لجمال الورد في الربيع’’
أسلوب القسم	‘‘فورب السماء و الأرض إنه لحق’’
أسلوب المدح و الذم	‘‘حبذا طلب العلم ’’ ، ‘‘نعم الخلق الحلم’’

(4) Traditional template:

This template is versatile enough to satisfy the needs of writing in so many fields like industry, software, sports, agriculture, economics… etc. We will develop our parser based on the traditional template. Training and testing phases will be conducted on the agriculture field. The following are some examples of the styles of the traditional template in the agriculture field:

"القمح محصول جيد"

جملة اسمية (مبتدأ(معرفة) + خبر(تركيب وصفي (نكرة + نكرة)))

"القمح يزرع في الأراضي الجديدة"

جملة اسمية (مبتدأ + خبر (جملة فعلية(فعل + شبه جملة (حرف جر + (تركيب وصفي(معرفة + معرفة))))))

"من المهم اتباع التوصيات الفنية"

جملة اسمية(خبر مقدم(شبه جملة(حرف جر + معرفة)) + مبتدأ (تركيب إضافي(نكرة + معرفة + معرفة)))

"تحرث الأرض باستخدام المحراث الحفار"

جملة فعلية(فعل + مفعول به + مكمل جملة (شبه جملة (حرف جر + تركيب إضافي (نكرة + معرفة + معرفة))))

In short, the goals of this thesis are as follows:

- Design and implement an Arabic Parser. The parser encodes the Arabic grammar rules of "irab" (إعـــراب) and the effects of applying these rules to the constituents of sentences.
- The ability of the system to be integrated into some other Arabic applications such as machine translation systems, systems for teaching Arabic, and system for checking and correcting grammatical errors.
- The ability of the systems to deal with the syntactic ambiguities.

1.3 Structure of the thesis

This thesis is organized as follows. The present chapter aims at presenting some background material and establish the problem area.

Chapter 2 gives a review on the grammar formalisms and parsing techniques that are commonly used in natural language processing systems.

In order to develop an Arabic parser, we should have the two components: a lexicon, and a morphological analyzer. The role of the two components is to prepare the suitable input to the parser. In Chapter 3 we discuss of the structure of the proposed lexicon and present a lexical analyzer for inflected Arabic words. An augmented transition network (ATN) technique was used to represent the context-sensitive knowledge about the relation between a stem and inflectional additions

Chapter 4 presents our attempt in developing an Arabic Parser for modern scientific text. The parser is written in Definite Clause Grammar (DCG). The developing of the parser was a two-step process. In the first step, we will present the architecture of the Arabic parser. In second step, we describe how we acquired the rules that constitute a grammar for Arabic that gives a precise account of what it is for a sentence to be grammatical. Moreover, we discuss the implementation aspects of the parser and show how the parser assigns grammatical structure onto input sentence

Chapter 5 reports on an experiment that tests the Arabic parser. Results are analyzed and discussed. Suggestions for improvements are also proposed.

Chapter 6 concludes the thesis and gives directions for future research.

CHAPTER (2)
DIFFERENT GRAMMAR FORMALISMS AND PARSING TECHNIQUES

This chapter gives a review on the grammar formalisms and parsing techniques that are commonly used in natural language processing systems. A grammar is a scheme for specifying the sentences allowed in the language, indicating the rules for combining words into phrases and clauses. Parsing the computation model for accepting an input sentence according to a formal grammar.

2.1 Rule-Based Grammars

Rule-based grammars describe the syntactic relationships between sentence components as a set of production rules, in the form of "X -> Y". The following grammars are some of most important rule-based grammars.

2.1.1 Formal Grammars

Chomsky defined a series of grammars [Winograd, 1983], based on rules for rewriting sentences into their component parts. He designated these as 0,1,2 and3 based on the restrictions associated with the rewrite rules, with 3 being the most restrictive, see table (2-1).

Table (2-1) the Chomsky Hierarchy

Type	**Language class**	**Rule Restrictions**
0	Recursively enumerable.	None
1	Context sensitive	No length-reducing rules
2	Context free	Left side must be a single non-terminal.
3	Regular	Left side must be a single non-terminal and right side must be a single terminal or a single terminal followed by a single non-terminal.

Phrase structure grammar –type1 and type2 – has been one of the most useful in natural language processing [Gazdar, 1982]. It has the advantage that all sentence structure derivations can be represented as a tree and practical parsing algorithms exist. Although it is a relatively natural grammar, it is unable to capture all the sentence constructions found in most natural languages.

2.1.2 Transformational Grammars

Roughly and intuitively, the transformations are designed to account for the systematic relationship between sentences such as ,active-passive sentence pairs; global sentence relationship, such as the relationship between "*what*" and "*eat*" in "*what will John eat,*" where "*what*" is the questioned object of "*eat";* and ambiguities in sentences such as "*they are flying planes,*" where one and the same string of words is derived from two different base sentences (one where "*flying planes*" is a kind of plane, with "*flying*" an adjective, and one where "*flying*" is the main verb). For instance in one version of transformational grammar developed about 1965 [N. Chomsky, 1965], the sentence "*John will eat the ice cream*" would be generated by a simple set of syntactic rules, and then a transformational rule operating on this basic sentence would invert "*John*" and "*will*" to produce the derived question "*Will John eat the ice cream.*". Another series of transformational operations could act on this sentence yet again to produce the passive sentence "*Will the ice cream be eaten by John*". This last sequence of operations involves adding new elements "*be*" and "*by*"; moving old elements around, and changing the form of existing elements in a sentence. Figure (2.1) gives the overall block diagram for this system.

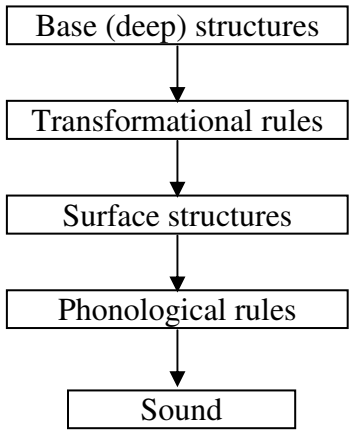


Figure 2.1. A block diagram of the components of a transformational grammar

2.1.2.1 Derivation process

The process of a sentence (surface structure) such as "*Will the ice cream be eaten by John*" has been the source of considerable confusion for computational analysis. A derivation does not give an algorithmic description or flowchart for how derived sentence could be mechanically produced or analyzed by a machine; that is, it does not directly give a parsing procedure for natural languages.

Over the course of many years the theory of transformational grammar has greatly altered the mechanisms used to generate the basic sentences, the definition of transformations, and the way that the final complexity of sentences is accounted for by the various subcomponents of the grammar. The general trend has been to have less and less of the final sentence form be determined by particular transformations or rules in the base grammar. Instead, relatively more of the constraints are written in the form of principles common to all languages or encoded in the dictionary entries associated with each word [B. Radford, 1981], [H. Van, 1986].

This approach is controversial [F. Newmeyer, 1980]. Other researchers in generative grammar have adopted quite different viewpoints about how best to describe the variations within and across natural languages. In general, these alternative accounts adopt means other than transformations to model the basic variation in surface sentences or assume other predicates than phrase structure relations are centrally involved in grammatical descriptions [M. Brame, 1978] . In the recent theory dubbed lexical-functional grammar (LFG) [J. Bresnan, 1983], there are no transformations. Instead, the differences between, for example, an active sentence like "*John kissed Mary*" and a passive sentence like "*Mary was kissed by John*" is encoded in the form of different lexical entries (dictionary entries) for "*kiss*" and "*kissed*" plus a connections between those lexical entries and the grammatical relations of subject and object. In this example, among other things the lexical entry for "*kiss*" says that "*John*" is the subject, and that for "*kissed*" says that "*Mary*" is the object. There is no derivation from deep structure but simply the direct construction of a kind of surface structure plus the assignment of grammatical relations like Subj. In another theory, generalized phrase structure grammar, the active-passive relationship is described by a metagrammar that, given a rule system that can produce the active sentence, derives a rule system to produce the corresponding

passive form. This derived grammar contains no transformations but simply generates all surface forms directly without a derivation from a deep structure.

2.1.2.2 Transforma1tional Component

Referring to figure 2.1, the base structures may now be fed to the transformational component, where zero or more transformations can be applied to generate additional sentences; the output of this process is a surface structure, ready to be "spelled out" and pronounced as a sentence. If no transformations apply, the surface structure is the same as the base phrase structure. This will roughly be the case for ordinary declarative sentences, such as "*John will eat the ice cream*". If transformations do apply, they produce new phrase markers, such as "*Will John eat the ice-cream*".

Each transformation is defined by a structural description that defines phrase markers to which it can apply and a structural change that describes how that phrase marker will be altered. That is, a transformation must match the description of a phrase marker and produces as output a new phrase marker. Further transformations may then apply to this new phrase marker, and so on. In this sense, transformations are much like an if-then production rule system, with the domain of the rules being phrase markers.

2.1.3 Case Grammars

Case grammar is a form of transformational grammar in which the deep structure is based on cases-semantically-relevant syntactic relationships [Fillmore, 1968/1977]. The notion of "case" has been used to refer to several related concepts. Traditionally, it has meant the classification of nouns according to their syntactic role in a sentence, signaled by various inflected forms. In English, only pronouns have these case inflections. For instance, the first person singular pronoun is "I" (nominative case), "me" (accusative/ objective), or "my"(genitive/possessive case) according to its use as subject, object, or possessive article. In languages as Greek all nouns are given affixes that indicate their case.

The idea of a direct relationship between inflections and cases is one kind of case, also called "surface" or "syntactic level" case.

Another sense of "case" (also called "deep case," "semantic case," or "theta role") is a categorization of ***NPs*** according to their conceptual roles in the action described by a sentence. Conceptual roles are independent of the particular verb or predicate being expressed. The agent case, then is a generalization of many ideas: kicker, reader, walker, and dancer; one who performs an action.

2.1.3.1 Surface Cases

The introduction discussed categorization nouns according to their endings or inflections. For the purposes of natural language processing, it is more useful to define surface case as a general syntactic categorization of noun phrases. Another way to think of surface case is as a property that is assigned to an ***NP*** manifested in the sentence as some syntactic marker or signal, called a case marker.

Various linguistic elements can be case markers. The primary one is the affix, that is, an ending attached to a noun form. Many would consider that prepositions (or postposition) serve a similar function. Word order, as in English, can also be viewed as a case marker. In addition, case assignment interacts with such features as gender and definiteness of the ***NP***.

This view of case, then, generalize the notion of surface case from simple noun inflections to a property that all ***NP***s have and that may be expressed with word endings, word order, and so on.

2.1.3.2 Deep Cases and Grammatical Explanation

The notion of deep cases is not new. For instance, Sonnen cheins's demand that cases "denote categories of meaning" [O. Jespersen, 1965] is in effect a statement that there are two levels of cases, the surface level indicating by case affixes and a deeper level that may be common to more than one language [Fillmore , 1968].

Because deep cases focus on (conceptual) events rather than on syntactic construction, they can help explain the relative "acceptability" of certain sentence. For example, the center idea is that the deep structure of a simple sentence consists of a verb and one or more noun phrases associated with the verb in a particular relationship. For instance the verb has the following cases:

agent, experiences, instrument, object, source, goal, location, type, and *path.*

The cases of each verb form an ordered set referred to as a case frame. A cases frame for the verb "*open*" would be:

(object (instrument) (agent))

Which indicates the open always has an object, but the instrument or agent can be omitted, as indicated by their surrounding parentheses. Thus the cases frame associated with the verb provides a template which aids in understanding a sentence.

2.2 Graph-Based Grammars

Another approach to representing natural language structures uses the model known as transition networks. Transition networks are based on the applications of the mathematical notions of graph theory and finite state machines to study of grammar. There are several types of transition networks, which differ in complexity and effectiveness [Wood, 1970].

2.2.1 Simple Transition Networks

This formalism is based of the notion of a transition network consisting of nodes and labeled arcs [Allen, 1995]. Consider the network named *NP* in figure 2.2 each arc is labeled with a word category. Starting at a given node, one can traverse an arc if the current word in the sentence is in the category on the arc. If the arc is followed, the current word is updated to the next word. A phrase is a legal *NP* if there is a path from the node *NP* to a pop arc accounting for every word in the phrase.

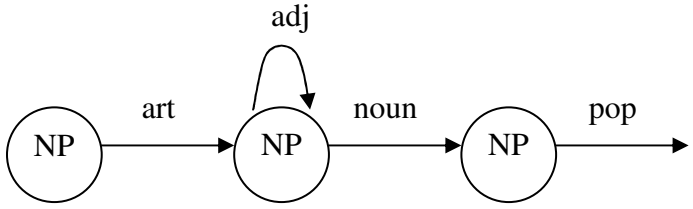


Figure 2.2. A simple NP grammar

2.2.2 Recursive Transition Networks

a recursive transition network is essentially a network of nodes representing partial states of knowledge that arise in the course of parsing a sentence. States are connected by "arcs" indicating kinds of constituents (words or phrases) that can cause transitions from one state to another. The states in the network can be conceptually divided into "levels" corresponding to the different kinds of phrase that can be recognized. Each such level has start-state and one or more final states and can be thought of as a recognition automation for one particular kind of phrase[J. Bresnan, 1983]. A simple pictorial example of a transition network grammar is illustrated in figure 2.3.

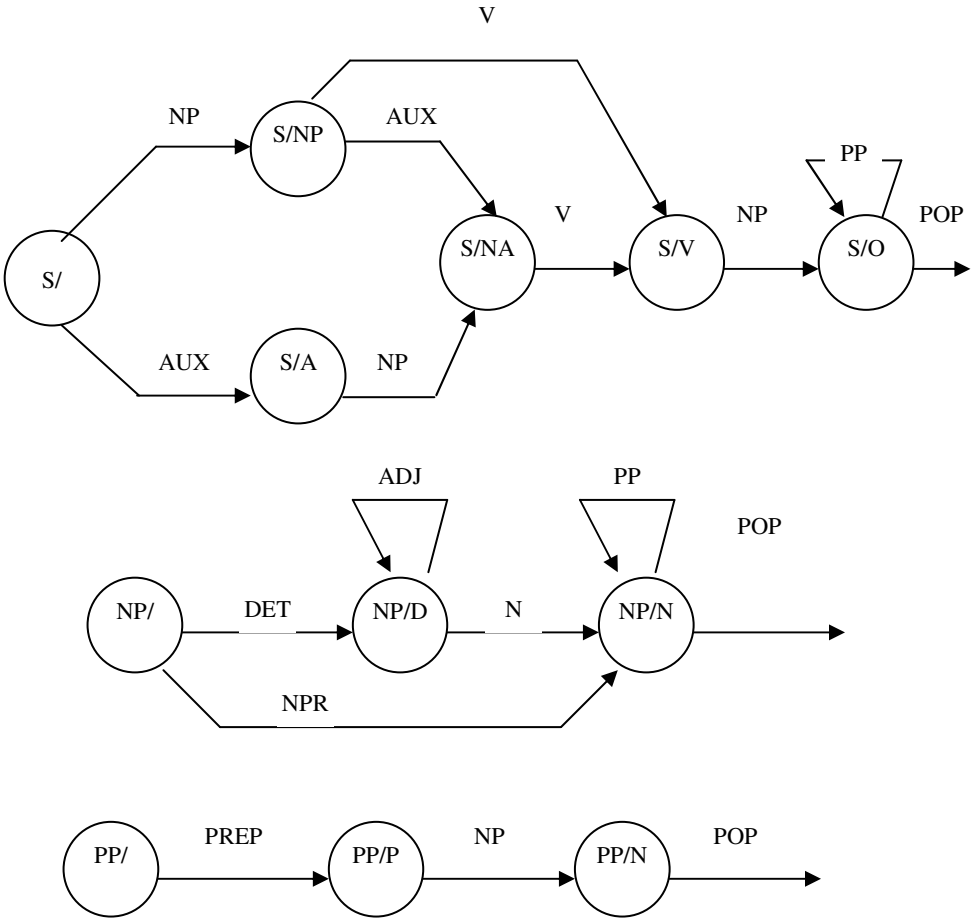


Figure 2.3. Sample transition-network grammar: S = sentence, NP = noun phrase, AUX = auxiliary, V = verb, PP = prepositional phrase, POP = end of phrase, DET = determine, ADJ = adjective, N = noun, PREP = preposition.

In figure 2.3, states are represented by small circles, and arcs are represented by arrows connecting states. Each arc is labeled with the name of kind of constituent that will enable that transition if it is found at that point in the input string. This sample grammar has three levels: S for sentence, NP for noun phrase, and PP for prepositional phrase. Each level begins with a state whose name indicates the kind of constituent being searched for. In the naming convention used here, a state name consists of the name of the constituent being sought, followed slash (/), followed by a brief mnemonic indication of what has been found so far. This naming convention is not an essential part of a transition network grammar but is useful device for making grammars readable. Each level ends with one or more final states (indicated by a short arrow labeled POP), which mark the successful completion of a phrase. A sequence of arcs from a start state to a final state defines a sequence of constituents that can make up a phrase of kind sought by the Start state.

The first state in the sample grammar (S/) is the state in which the parser begins and is the state of knowledge corresponding to the initial assumption that a sentence is to be parsed. The topmost arc sequence in the figure shows that a sentence (S) can consist of a noun phrase(NP), followed by a verb (V), followed by another noun phrase (NP), followed by any number of prepositional phrases (PPs). Alternatively, the first noun phrase can be followed by an auxiliary (AUX) before the verb, or the sentence can be begin with an AUX followed by an NP before picking up in state S/NA with the same predicated verb phrase constituents as in the first two cases.

The grammar model described above is called a recursive transition network or ***RTN*** because the arcs of the grammar can invoke other levels of the network to recognize subordinate constituents that can in turn invoke other levels (recursively). This process may eventually reinvoke some level "inside itself" (genuine recursion).

2.2.3 Augmented Transition Networks

An augmented transition network grammar (***ATN***) is a recursive transition network that is augmented with a set of registers that can hold partial parse structures and with conditions and actions on the arcs that can test and set these registers [Kaplan, 1972]. The

registers can be set to record attributes of the phrases being recognized and tested to determine the acceptability of a particular analysis. For example, register can be used to represent the person and number of the subject of a sentence, and a condition can be used to check that the subject and the subsequent verb agree in person and number.
ATN were developed in order to obtain a grammar formalism with the linguistic adequacy of a transformational grammar and the efficiency of the various context-free parsing algorithms. As a sentence is parsed with an ATN grammar, the conditions and actions associated with the transitions can put pieces of the input string into registers, use the contents of registers to build larger structures, check whether two registers are equal, and so on. It turns out that this model can construct the same kind of structural descriptions as those of a transformational grammar and can do it in a much more economical way. The merging of common parts of alternative structure, which the network grammar provides, permits a very compact representation of quite large grammars.

The ATN suggest “here” a way of viewing a grammar as a map with various landmarks that one encounters in the course of traversing a sentence. Viewed in this way, ATN grammars serve as a conceptual map of possible sentence structures and a framework on which to hang information about constraints that apply between separate constituents of a phrase and the output structure that the grammar should assign to a phrase.

Another advantage of the transition network formalism is the ease with which one can follow the arcs backward and forward in order to predict the types of constituents or words that could occur to the right or left of a given word or phrase.

2.3 Logic-Based Grammars

Research into building grammars for understanding natural language became more popular after the introduction of grammar formalisms based on Horn clauses by Colmerauer in 1975 [Kowalski, 1979]. The so-called metamorphosis grammars (MGs) started a growing interest in expressing linguistic concepts in logic and supported the construction of robust front-ends and interfaces. The primary applications of this research

were the consultation and creation of databases through natural languages, generation of answers and questions, text translation, and text synthesis from formal specifications.

In the framework of logic programming, several formalisms have been proposed for analyzing/generating languages: metamorphosis grammar [Colmeruar, 1978] Extraposition grammars[Pereira, 1981]. These formalisms can be generated under the concept of "Logic Grammars" or "Term Grammars" [Sabatier, 1985]. The following features distinguish these grammars from classical ones:

- Symbols are terms: atoms, predicates (ith variables) and variables; instead of simple atoms.
- The rewriting of symbols involves unification instead of simple replacement.
- Conditions can expressed in a rule, and executed as procedure calls. They generally formulate conditions on the value of variables occurring in the rule.

2.3.1 Definite Clause Grammars

The notion of definite-clause grammars (DCGs), a special case of MGs, was introduced in 1978 by Pereira and Warren [Pereira and Warren, 1986] as a grammar formalism for which PROLOG provides an efficient parsing mechanism.

It is well known that CFGs are not fully adequate for describing natural language, nor even many programming languages. DCG overcomes this inadequacy by extending CFGs in three important ways [Pereira and Warren, 1986]. Firstly, DCG provide for context-dependency in a grammar, so that the permissible forms for a phrase may depend on the context in which that phrase occurs in the string. Secondly, DCGs allow arbitrary tree structures to be built in the course of the parsing. Thirdly, DCGs allow extra conditions to be included in the grammar rules; these conditions make the course of the parsing depend on auxiliary computations, up to an unlimited extent.

DCG allow any logic term to be a non-terminal, and they are built upon logic symbols (atoms, variables, and terms) instead of only atomic constants. They have only one non-terminal symbol on the left side of each rule.

A DCG rule has the following form:

```
Nonterminal-symbol → body.
```

Where “body” is a sequence of one or more items separated by commas. Each item is either a non-terminal symbol or a sequence of terminal symbols. The meaning of the rules is that “body” is a possible form for a phrase of type non-terminal symbol. A non-terminal symbol is written as a PROLOG term (other than list), and a sequence of terminals is written as a PROLOG list.

In the right side of a rule, in addition to non-terminals and lists of terminals, there may also be sequences procedure calls, written within curly brackets ({and}). These are used to express extra conditions that must be satisfied for the rule to be valid.

2.4 Parsing Techniques

In particular, many different types of formal, generative grammars have been devised to specify the sentences of language and to pair each sentence with a corresponding set of structural descriptions. The nature of these grammars, however, usually does not provide an obvious algorithm for computing structural descriptions from sentence. In this respect they are similar to systems of logic, which implicitly specify a set of provable theorems but don’t explicitly tell how a particular theorem is to be proved. Just as proof procedures must be devised for systems of logic, so must parsing procedures be devised for formal grammars of natural languages. Parsing a given sentence with respect to a given grammar, then, is the process of determining whether the sentence belongs to the language specified by the grammar and, if so, finding all the structures that the grammar pairs with the sentence.

2.4.1 Top-down Parsing

A top-down parser tries to construct a parse tree starting from the root (i.e. the top symbol of the grammar) and expand it using the rules of the grammar [Convington M., 1994]. Figure 2.5 show a sample of a small grammar for experimenting with parser.

S → NP VP
NP → D N
VP → V NP
VP → V
D → *the, all, every*
N → *dog, dogs, cat, cats, elephant, elephants*
V → *chase, chases, see, sees, amuse, amuses*

Figure 2.5 a sample grammar for experimenting with parser.

Now recall how top-down parsing works. To parse *the dog barked*, the parser, in effect, says to itself:

- I'm looking for an **S**.
- To get an **S**, I need an **NP** and a **VP**.
- To get an **NP**, I need a **D** and an **N**.
- To get a **D**, I can use *the*… got it.
- To get an **N**, I can use *dog*… got it.
- That completes the **NP**.
- To get a **VP**, I need a **V**.
- To get a **V**, I can use *barked*… got it.
- That completes the **VP**.
- That completes the **S**.

To get a feel for what's going on, try drawing the tree as you work through these steps. Figure 2.6 shows the order in which a top-down parser discovers the various parts of the tree.

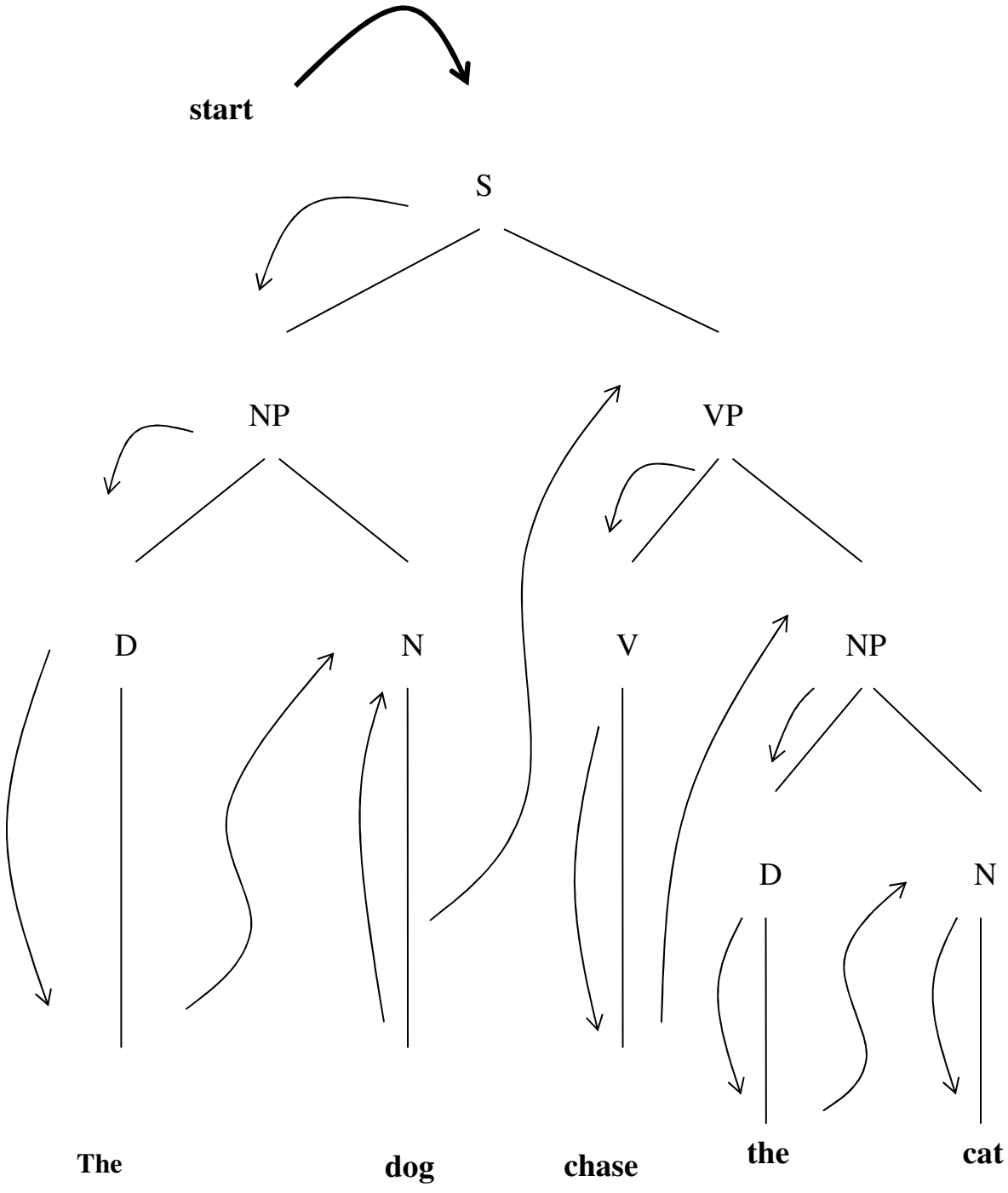


Figure 2.6 top-down parsing—The parser discovers the nodes of the tree in the order shown by the arrows.

Algorithm

The parser will work through the input string, accepting words one by one. Let **C** represents the kind of constituent the parser is looking for at any particular moment (an S, NP, V, or whatever). Then the algorithm to parse a constituent of type C is:

- If C is an individual word look it up in the lexical rules and accept it from input string.
- Otherwise, look in the PS rules to expand C into a list of constituents, and parse those constituents one by one.

2.4.2 Bottom-up Parsing

A familiar limitation of top-down parsers is that they loop on LEFT-RECURSIVE rules of the form **A→ A B** ("to parse an A, parse an A and then …"). Such rules occur in natural language; one example is **NP→ NP** *Conj* **NP** , where *Conj* is a conjunction such as *and* or *or*.

One way to handle left-recursive rules is to parse BOTTOM-UP. A bottom-up parser accept words and tries to combine them into constituents, like this:

- Accept a word… it's *the*.
- *The* is a D.
- Accept another word… it's *dog*.
- *Dog* is an N.
- D and N together make an NP.
- Accept another word… it's *barked*.
- *Barked* is a V
- V by itself makes a VP.
- NP and VP make an S.

Again, try drawing a tree while working through these steps. (During most of the process it will of course be a set of partial trees not yet linked together.) you'll find that you encounter the various parts of the tree in the order shown by the arrows in the Figure 2.7.

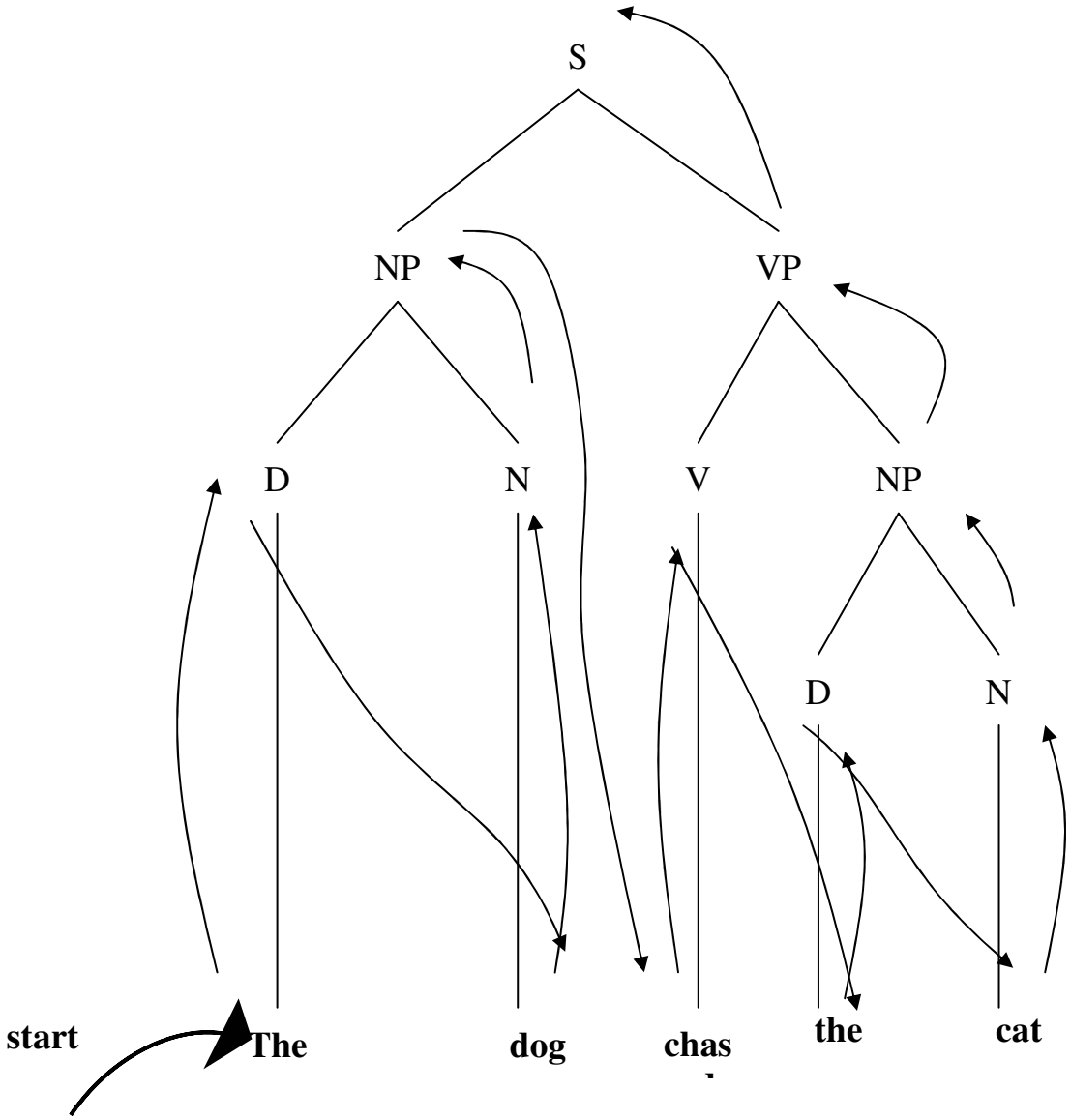


Figure 2.7 Bottom-up parsing—The parser discovers the nodes of the tree in the order shown by the arrows.

Because its actions are triggered only by words actually found, this parser does not loop on left-recursive rules.

Algorithm

The bottom-up that we have just sketched id often called SHIFT-REDUCE parsing. It consists of two basic operations: to SHIFT words onto a stack, and to REDUCE (simplify) the contents of the stack. Table 2.1 shows an example.

So the shift-reduce algorithm is as follows:

1. Shift a word onto the stack.
2. Reduce the stack repeatedly using lexical entries and PS rules, until no further reductions are possible.
3. If there are more words in the input string, go to step1. Otherwise, stop.

A noteworthy feature of shift-reduce parsing is that it has no expectations. That is, you can give it an input string and say, "what kind of constituent is this?" and get an answer. You do not have to say "Parse this as an S" or "Parse this as an NP."

Table 2.1. Example of Shift-reduce algorithm

Step	***Action***	***Stack***	***Input String***
	(Start)		*the dog barked*
1	Shift	*The*	*dog barked*
2	Reduce	**D**	**Dog barked**
3	Shift	**D dog**	**Barked**
4	Reduce	**D N**	**Barked**
5	Reduce	**NP**	**Barked**
6	Shift	**NP barked**	
7	Reduce	**NP V**	
8	Reduce	**NP VP**	
9	Reduce	**S**	

2.4.3 Transition Networks Parsers

a variety of parsing techniques may be used for RTN and ATN formalisms. For instants, a simple top-down backtracking algorithm and mixed-mode algorithm can be used for RTNs.

They are a natural extension of finite-state automata, the automata equivalent of finite-state grammars. A finite-state automata (FSA) is a finite set of states connected by directed arcs, each labeled with a symbol from a terminal vocabulary Vt. One state is designed as the initial state, and some subset of the states are designed as final states. In following a path from the initial state to any of the final states, a sequence of arcs is

traversed, and the corresponding string of labels from Vt is said to constituent a sentence specified by the FSA. the set of sentences so specified is the language generated by the FSA. The automaton is said to be a deterministic finite-state automaton (DFA) if no node has two or more outgoing arcs bearing the same label. Otherwise, the automaton is said to be a non-deterministic finite-state automaton (NFA) [Convington M., 1994].

2.4.4 Chart Parser

A chart parser stores partial results in a table called the chart [Convington M., 1994]. Consider the following two rules

VP → V, NP.

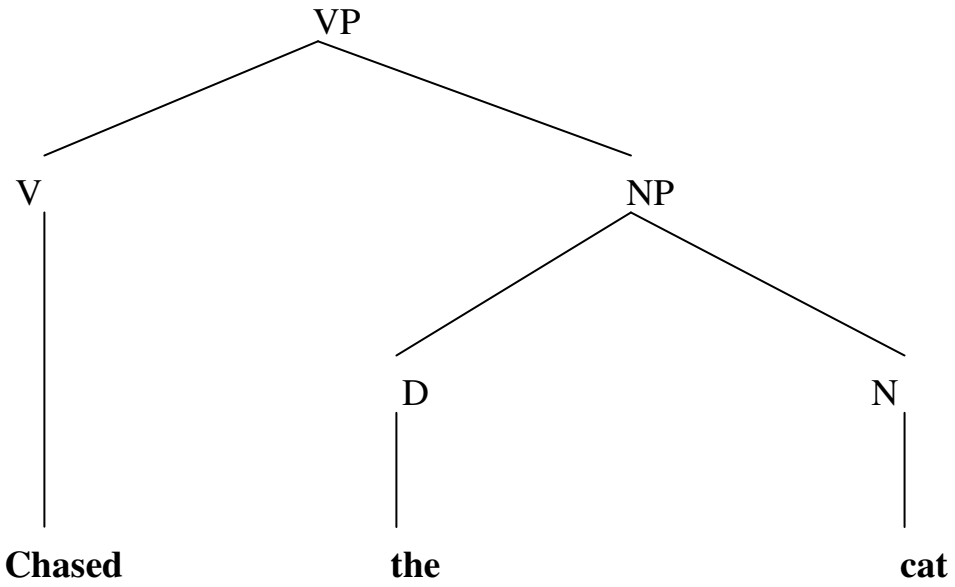
VP → V, NP, PP.

Now consider the query:

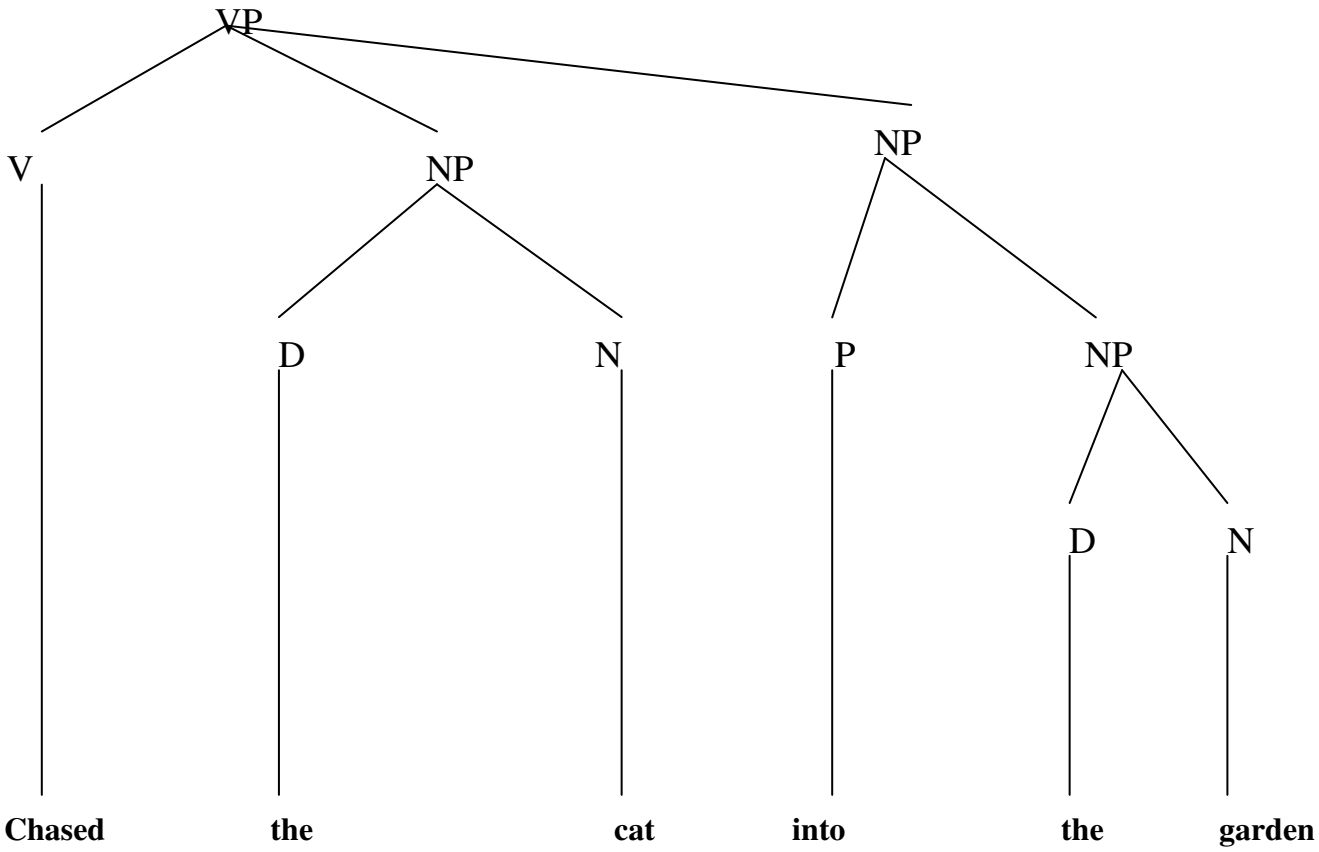
```
?- vp([chased,the,cat,into,the,garden],[]).
```

Which means that, is those words make a VP sentence.

The parser tries the first rule, work through the structure



And then realizes it hasn't used up all words. So it backs up, forgets all the work it has just done, chooses the second VP rule, and parses the same structure again. Then it goes ahead and parses the PP, giving:



Crucially, the parser had to parse the same V and NP twice. And that could have been a lot of work --- recall how complicated an NP can be.

A chart parser or *tabular parser* is a parser that remembers substructures that it has parsed, so that if it has to backtrack, it can avoid repeating its work. For example, the first time through, a chart parser would make a record that *the cat* is an NP. The second time through, when looking for an NP at the beginning of *the cat into the garden,* it would look at its records (the CHART) before using the NP rule. On finding [NP the cat] in the chart it would not have to work through the process of parsing the NP.

To save each constituent, the parser must record:

- What kind of constituent it is;
- Where it begins; and
- Where it ends.

One way to represent this information is to store clauses such as:

```
chart(np,[the,cat,into,the,garden],[into,the,garden]).
```

This means: "if you remove an NP from the beginning of *the cat into the garden* you're left with *into the garden.*" That is, *the cat* is a noun phrase.

The two lists represent positions in the input string. This is not the most concise or efficient way to represent positions, but it is the most readable to human eyes, and we will stick with it through most of this chapter.

The following Prolog code, shows how to use a chart in the parser:

```
parse(C, [Word | S], S): -
     word(C, Word).
parse(C,S1,S): -
     chart(C,S1,S).
parse(C,S1,S): -
     rule(C,Cs),
     parse_list(Cs,S1,S),
     asserta(chart(C,S1,S)).
```

The first clause deals with individual words, as in the original top-down parser. it's faster to look up single words in the lexicon than to try to get them into chart.

The second clause says, “if what you’re looking for is already in the chart, don’t parse it.” The third clause, using *rule* to parse the constituent C, and return its decomposed units Cs, the *parse_list* do the same function of *parse* except that it takes a list of constituents and parses all of them, at the end the parser will asserts a fact into the chart.

Completeness

As showed so far, the chart parser can remember what it found, but it can’t remember what it failed to find. For example, it can remember that the cat is an NP, but it can’t remember that the cat is not a PP. This means that with a complex grammar, the parser can waste more time than it saves.

To parse any constituent, the parser first looks at the chart. If the chart doesn’t contain the constituent that it’s looking for, then the parser proceeds to do all the work that it would have done if there had been no chart in the first place. In such a case the chart doesn’t save it any work.

This situation arises mainly with fairly large grammars. Here is a simple example.

Consider the phrase-structure rules

VP → V NP [PP] [Adv]
NP → D N [PP]
PP → P NP

This is really four VP rules – with and without PP and with and without Adv – as well as two NP rules, with and without PP.

Suppose the parser is parsing [vp saw the boy yesterday]. Two different rules allow a PP to come after *boy*. According, the parser will make two attempts to find a PP at that position, and both attempts will fail. Each attempts will consist of a fair bit of work (invoking PP → P NP, then looking at lexical entries for P, and failing to find *yesterday* among them). If, the first time through, the chart could record that a PP is definitely not present at that position, the second attempt to find one would be unnecessary.

There is also another reason why the chart needs to store negative information: unwanted alternatives. Every parse that can be found in the chart can also be found without using

the chart. This means that, upon backtracking, the parser can often get the same constituent more than one way, and the number of alternatives to try grows exponentially. A much better approach is to use the chart only if it is *complete* for a particular constituent type in a particular position. That is, if you're looking for an NP at the beginning of *the cat near the elephant,* use the chart only if it is known to contain all possible NPs in that position. Otherwise, ignore the chart and parse conventionally. And if, when parsing conventionally, you fail to find what you're looking for, then assert that the chart is complete for that constituent in that position, because there are no more alternatives to be found.

CHAPTER (3)
LEXICON AND THE MORPHOLOGICAL ANALYZER

In order to develop an Arabic parser, we should have the two components: a lexicon, and a morphological analyzer. The role of the two components is to prepare the suitable input to the parser. In this chapter we discuss the structure of the proposed lexicon and present a lexical analyzer for inflected Arabic words. An augmented transition network (ATN) technique was used to represent the context-sensitive knowledge about the relation between a stem and inflectional additions.

In a previous work by [Shaalan, 1989], [Rafea et al., 1993], they developed a morphological analyzer for inflected Arabic words using Pascal language. With respect to the implementation of the Arabic parser, we took the advantage of the already developed morphological analyzer. This morphological analyzer is reimplemented in Prolog and integrated with the Arabic parser.

3.1 The lexicon

Word information in Arabic involves three concepts: root, pattern and form. Word forms (e.g. verbs, verbal nouns, agent nouns, etc.) are obtained from roots by applying derivational rules to obtain corresponding patterns. For example, the meaning of the word form (كاتب) is the combination of the root (كتب) and the pattern 'faal' (ف-ا-ع-ل). Thus, Arabic has the characteristics that from one root the derivational and inflectional systems are able to produce a large number of words (lexical forms) each having specific patterns.

3.1.1. Lexicon entry

Here we will show the constraints, which we find very important, on the entries in the lexicon. These constraints are

1- The word, at any entry, must be meaningful. (The stem form 'صحراء' is meaningful but the stem form 'صحراو' is not).
2- Retrieval of the stem stored in the lexicon from an inflected word using this stem must be governed by rules that can be applied in an efficient way. This

leads to storing the broken plural form, as there are no regular rules to transform a word in plural form into its singular form. Even if artificial rules can be used to obtain the singular form easily from a plural form in some cases. This may be very difficult in other cases. Consequently we take the decision to store all the broken plural forms for homogeneity and consistency.

3- As Arabic is a language that allows many words to be constructed by derivation from others, most researchers in Arabic morphology use a lexicon in which roots are the only entries. This leads to substantial processing effort to obtain the root from a derived word. For example, the words such as 'استعمل' (to use), 'عامل' (worker - factor), 'تعامل' (to deal with), 'معمل' (laboratory), 'استعمال' (the use), 'عميل' (customer), 'عملة' (currency - coin), 'عمالة' (labour force), 'أعمال' (jobs-business), 'عمولة' (commission), 'معاملة' (treatment) and 'عملي' (practical) are all looked up under the root 'عمل'. If the goal of the language-processing is understanding, another lexicon may be created that contains derived nouns because the relation between derivational rules and the semantic change may be very deep and difficult for coding, as shown in the above example. Therefore, we have decided to store the derived nouns as separate entries in the lexicon. This will serve two purposes:

(a) Less processing will be needed to reach the stem, which is a derived noun, as it is stored explicitly in the lexicon.

(b) Only one lexicon will be used for lexical analysis and understanding the meaning of the derived noun. Consequently, there will be no need for another lexicon to be used for understanding after morphological analysis. This is likely to lead to an overall saving in the storage needed for lexical.

3.1.2. The lexicon structure

In our proposed lexicon , for each stem, we store the following features:

- *Stem word***.** The word of the stem , which will be occur in the context.
- *Stem type***.** The type of the stem , (verb , noun , …etc).

Stem Word	Stem Type
بيت	Noun (اسم)
حصد	Verb(فعل)
هؤلاء –هذان ...	demonstrative pronoun (اسم
تحت	accusative place (ظرف مكان)
بين	accusative time (ظرف زمان)
لن –لا ...	Negation article (حرف نفي)
إلي –من ...	Preposition (حرف جر)
هي –هو ...	Separated pronoun (ضمير

- *Transitive verb***.** If the stem type was verb, this feature in the lexicon must be determinate if the stem (verb) was transitive verb (متعدي) or it is intransitive verb (لازم).
- *Special verb***.** If the stem type was verb, we also need to determine if it is special verb or not.

..)...كان واخوتها (كان, اصبح , لا يزال

.) ...إن واخوتها (إن ,أن

- *Stem number* **(العدد).** This feature is applied only on the noun stems, to determine the number of the stem: singular, dual or plural.
- *Defined and undefined* **(التعريف).** Also applied only on the noun stems, if the stem (noun) was defined or undefined or proper name.
- *Morphological pattern* **(الوزن).** If the stem has any of special morphological patterns which needed in the grammar rules.
- *Can be used as an adjective* **(الإمكانية للاستخدام كنعت).** Applied only on the noun stems, to determine its ability to be adjective or not.

Word	ability to adjective
قوي	Yes
خضراء	Yes
قمح	No
رجل	No

- *Ability to Annexation* **(القابلية للإضافة).** Also applied only on the noun stems, to determine if the stem has the ability to be annexed or not .

Word	ability to annexation
رجل	Yes
أرض	Yes
كل	No

Implementation

The lexicon is implemented as Prolog facts. The absence of a stem feature is represented as an empty list, i.e. []. The following are examples of the lexicon entries.

stem(['متعدي'],['فعل'],['د','ا','ع'],[],[],[],[],[],[],[],[]).

stem(['لازم'],['فعل'],['ر','ث','ع'],[],[],[],[],[],[],[],[]).

stem(['حرف_جر'],['ي','ل',' إ'],[],[],[],[],[],[],[],[],[]).

stem(['مفرد'],['مذكر'],[],[],[],['ضمير_منفصل'],['و','ه'],[],[],[],[]).

stem(['حرف_وصل'],['و'],[],[],[],[],[],[],[],[],[]).

stem(['نكرة'],['مفرد'],['مذكر'],[],[],[],['اسم'],['م','ه','م'],[],[],[]).

3.2 The Arabic morphological analyzer

In this section we will introduce the main function of the morphological analyzer, we will discuss the methodology of this function and the implementation of it too. In our definition, the function of the analyzer is to scan the inflected word and break it down into beginning additions, a stem, and last additions. We use the augmented transition network (ATN), in the implementation of the analyzer, as a method for the representation of the relations between the additions for an inflected Arabic word.

3.2.1 Word analysis

An Arabic word can be represented as follows:

```
[Begin_1 | Begin_2] + Stem + [Last_1] + [Last_2] + [Last_3]
```

Where any term between the square brackets may be absent or occur once, and "|" stands for alternatives. The word is scanned for different additions, and what remains is the stem. Unfortunately, the process is not as simple as it may seem. One cannot simply look up the additions in a list and compare them with the beginning and ending characters of an inflected word. The source of difficulties can be summarized as follows:

1. There may be overlapping characters within the additions that occur before a stem, and this may also happen in the additions following the stem. For example, the overlap between the ending additions 'تا' and 'ت' in the stem 'أخذ' (took) can

be found in its inflected form 'أخذت' where the agent is a singular first person (I took), and 'أخذتا' which is the inflected form where the agents are dual second person.

2. There may be overlapping characters between the additions and a stem.
3. The stem itself may be modified due to the addition of the inflectional symbols. Consequently, some modification actions may be required, after the removal of the additions, and before searching the stem in the lexicon. For example, in the stem 'قال' (said), its middle letter can be converted in the inflected word 'يقول' (saying-says).
4. More than one interpretation may arise for a stem and its additions.

3.2.1.1 Word Affixes

English language analysis relies on storing words that are classified according to their type (verb, noun, adjective etc.) directly in the dictionary without the need of word reduction. Similarly, we classified the Arabic words according to its syntactic categories. Depending on the method used for analyzing Arabic words into a stem and affixes, one can tell whether a specific word is a verb, noun, or adjective, etc.

Two types of affix are possible, namely: prefix and suffix.

(1)Prefixes:

A prefix may exist in either verbs or nouns to serve various syntactic goals depending on the prefix value. On one hand, verb prefixes are employed in general to specify the tense of a verb. On the other hand a noun prefix may be a preposition, a determiner, etc. table 3.1 shows the two types of prefixes.

Table 3.1. Prefixes of Arabic Word.

Verb Prefixes	Noun Prefixes
فسأ–سأ	فال
فسن–سن	ف
فست–ست	ك
فسي–سي	لل
ت– ي – ن –أ	ال
فلت– فلي –فلن	بال
لت– لي –لن	ب
	كال
	ل

(2) Suffixes:

The set of suffixes in Arabic may be classified into two disjoint subsets for verb suffixes and noun suffixes. On one hand, verb suffixes may have a string syntactic effect in that a single word can contain both a subject and an object in addition to the verb. On the other hand, noun suffixes are mainly concerned with uniquely determining the "person, number, and gender" of a given noun. Table 3.2 shows the two types of suffixes.

Table 3.2. Suffixes of Arabic Word.

Verb's Suffixes	Noun's Suffixes	Suffixes for both
تما	ة	ا
تم	ات	نا
تن		ان
تا		ين–ون
وا		ه - ها
و		هن – هم –هما
ني		كن– كم –كما

3.2.2 Context-sensitive knowledge representation

We choose to use the ATN as the basic structure for representing syntactic knowledge for building inflected Arabic words. This has enabled us to implement the following:

(a) Placing the context-sensitive knowledge as rules associated with the arcs of the ATN. The rules have both condition and action parts. More than one rule may be associated with one arc. One of these rules is to be selected whenever the arc is traversed.

(b) Getting all possible interpretations of an inflected word by searching the ATN exhaustively along different possible arcs.

3.2.2.1 The algorithm of the lexicon analyzer

The lexical analyzer traverses the ATN in a depth-first manner and backtracks to search exhaustively for all possible solutions. We have found that this is the best way to implement an Arabic lexical analyzer. In effect, the conditions associated with the arcs are placed in such a way that the arc to be traversed first is the one that leads to the most probable solution. This ordering is done using intuitive and heuristic measures. As shown in figure 3.1, the arcs emitted from state *S0* are ordered such that a solution is sought by omitting 'begin' (the first recognizable prefix part of a word) and repeating this process on the rest of the word in case of failure. We use this order of search because the occurrence of nouns is more frequent than the occurrence of the verb. To clarify the method, the word 'فجر' (dawn) will be taken as an example to show how the order of traversing the ATN will affect the output of the lexical analyzer. If the arc labeled 'omit begin' is traversed first, the following is the output:

1. Stem: 'جر' (which is a verb that means 'draws' or 'drag')
2. Begin_2: 'ف' (which is a particle indicating coupling).

However, if the arc labeled 'match' is traversed first, the following is the output of the lexical analyzer:

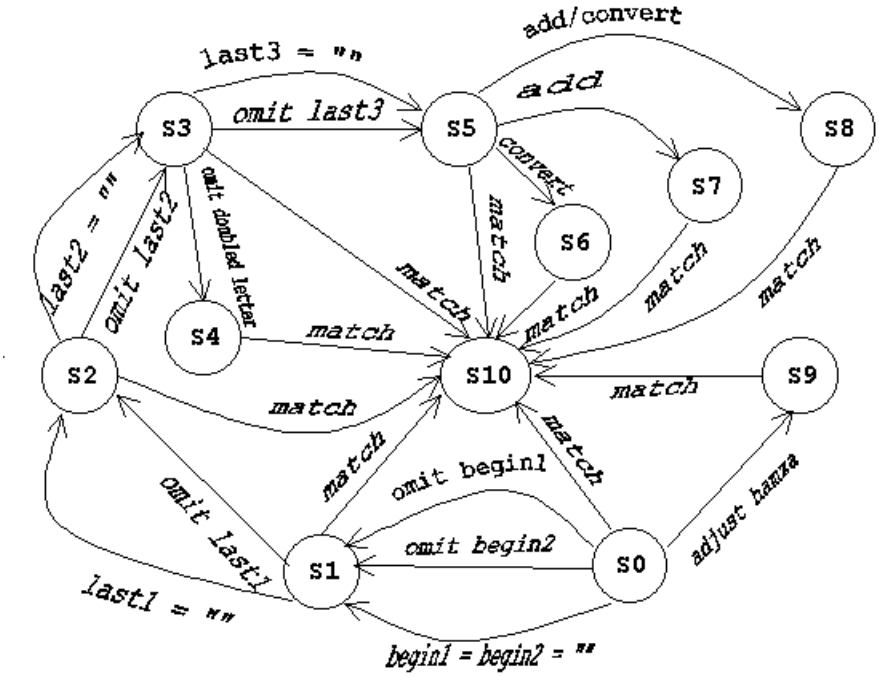
1. Stem 'فجر' (one possible entry for this stem in the lexicon is a noun which means dawn).

This example shows the importance of ordering in a deterministic processing of the ATN. However, non-determinism is a feature of natural language processing. Therefore, we have to ensure that generating all solutions remains possible in our system.

3.2.2.2 The augmented transition network

An augmented transition network (ATN), which describes the relation of an Arabic stem and additions, is built according to figure 3.1. Every state of the ATN is implemented by a Prolog predicate. A transition from a state *si* to an adjacent state *sj* is implemented by calling the procedure corresponding to *sj* from the procedure representing *si.* The condition on an arc is implemented by a Boolean called from a procedure representing a state to decide if a transition to an adjacent state is possible or not.

Fig. 3.1 **ATN representing the relation between the additions and stem of an inflected Arabic word.**



The action associated with this arc, if any, is called from the function representing the condition. If an arc has more than one rule associated with it, the Boolean function representing the condition will scan the conditions of the rules associated with the arc in a fixed order. If one of the conditions is satisfied, the corresponding action is activated. Actions associated with arcs are implemented as procedures.

3.2.2.3 Data structure associated with the ATN

The conditions and actions associated with the arc use registers and a set of status flags. Five registers correspond to the additions. Each of which corresponds to one of the

classes of additions in figure 3.1: `Begin_1,Begin_2,Last_1,Last_2 and Last_3.` The sixth register is reserved for storage of the current assumed stem. The initial value stored in this register is the whole-inflected word. The status flags indicate the modifications made the stem. Any such modification can be described by four parameters:

(a) the action taken (addition, conversion, omission)

(b) the position of the added, converted or omitted letter(s)

(c) the letter to be converted (held currently in the stem register)

(d) the letter to replace the one to be converted in a conversion process or the letter to be added in an addition process.

A possible solution of the inflected word will fill the registers and set the flags according to the interpretation made by the lexical analysis. Alternatives of the morphological analyzer output are kept in a Prolog list, which is passed for the parser.

3.2.2.4 Rules of the morphological analyzer

As we mentioned above, the conditions and actions associated with the arcs can be considered as rules.

The rule associated with the arcs leading to the final state for matching a stem in the lexicon is:

```
IF   the stem is matched with a lexicon entry
THEN retain this situation as a possible solution
```

The rest of the rules can be classified into four groups:

(1) A rule of this group, matches the first or last letters of the inflected word at any stage of parsing, against one of the addition sets. Example of these rules are:

```
IF   first characters of the inflected word are in
     Begin1_set
THEN omit these characters and keep them in
     begin1_register

IF last characters of the inflected word are in Last2_set
THEN omit these characters and keep them in last2_register
```

For example, the application of these two rules to the inflected word 'الناخبون' yield to:

- Begin1_register : ال
- Last2_register : ون
- Stem_register : ناخب

(2) a rule of this group, converts on of the stem letters to its original form (as stated in lexicon). The premises of these rules depend on the letter to be converted and the omitted additions. Examples of this rule is:

```
IF        last letter stored in Stem_register is 'و'
     AND  (    (last2_register = 'ان')
           OR  (last2_register = 'ون')
           OR  (last2_register = 'ات')
          )
THEN convert 'و' to 'ء'
```

For example, the application of this rule to the inflected word 'صحراوان' yield to:

- Last2_register : ان
- Stem_register : صحراء
- Status_flags : 'و' is converted to 'ء'.

(3) A rule of this group adds letters to the unrecognized stem to substitute for the omission of these letters due to the inflection process. In this class, we have three types of rules:

- one type is adding one or more of the following letters depending on the length of the unrecognized stem : 'ي' , 'أ' , 'و' , 'ا' depending on the number of letters in the unrecognized stem. An example of this rule is:

```
IF the length of the unrecognized stem stored in
Stem_register is two letters
```

THEN add 'و' at the beginning of the stem

For example, the application of this rule to the inflected word 'عد' yield to:

- Stem_register : وعد
- Status_falgs : 'و' is added

- the second type is to test the omitted addition before adding any letter. Example of this rule is:

```
IF (      (last1_register = 'و ا')
     OR   (last1_register = 'ي')
     OR   (last1_register = 'و ن')
     OR   (last1_register = 'ين')
     OR   (last1_register = 'و')
   )
THEN add 'ي' at last
```

For example, the application of this rule to the inflected word 'مصطفون' yield to:

- Last2_register : ون
- Stem_register : مصطفى
- Status_falgs : 'ي' is added

- the third type is to add 'ي' in a heuristic manner to recognize the defective noun that its last 'ي' letter is omitted in some situations. This situation arises when all possibility are investigated and stem is not recognized e.g. if the inflected word is 'محام' then adding 'ي' at the ending of the stem will yield to:

- stem_register : محامي
- status_falgs : 'ي' is added

(4) a rule of this group, converts one of the stem letter and adds a letter to the unrecognized stem. This class can be actually splitted into rules belonging to class 2, and class 3 but we include a separate class to accelerate the recognition process

and to accommodate words that may be recognized this way. In other word, if this rule is splitted into two rules one for conversion and the second for addition, each one of these two rules will belong to two different classes. The lexical analyzer will reach the conclusion after two trials. Example of this rule is:

```
IF      middle letter stored in stem_register is 'ي'
   AND  (     (begin2_register = 'أ')
          OR  (begin2_register = 'ن')
          OR  (begin2_register = 'ي')
          OR  (begin2_register = 'ت')
          OR  (begin2_register = 'سأ')
          OR  (begin2_register = 'سن')
          OR  (begin2_register = 'سي')
          OR  (begin2_register = 'ست')
          OR  (begin2_register = 'فسأ')
          OR  (begin2_register = 'فسن')
          OR  (begin2_register = 'فسي')
          OR  (begin2_register = 'فلن')
          OR  (begin2_register = 'فلت')
          OR  (begin2_register = 'فلي')
         )
 THEN     convert 'ي' to 'أ' at middle, and add 'أ' at first
```

For example, the application of this rule to the inflected word 'يستطيع' yield to:

- Begin2_register : ي
- Stem_register : أستطاع
- Status_flags : 'ي' at middle converted to 'ا', and 'أ' added at first.

(Note that, the inflected word 'يستطيع' was broken down into 'ستطيع' and 'ي' before satisfying the condition of the above rule.).

3.2.2.5 Implementation of morphological analyzes in Prolog

It is straightforward to represent ATN in Prolog and to write programs to perform various operations with them before we start writing general network traversal programs, we need to consider how we are going to represent transition networks as Prolog. We need to know the following about networks:

- what are its initial nodes?
- what are its final nodes?
- what are its arc are?

Where each arc is defined by the following: The departure node, the destination node and the label on the arc.

This can be represented in Prolog by using simple predicates of the appropriate kind:

```
initial(NODE)
final(NODE)
arc(DEPARTURE_NODE, DESTINATION_NODE, LABEL)
```

3.2.2.6 Recognition of the stem using exhaustive search

Satisfying the ATN searching for a possible path from the initial state to the final state does the recognition of the stem. The predicate recognize (NODE, STRING, STRINGOUT) will be true if the given STRING can be accepted by our ATN with the traversal starting at the given NODE.

Firstly, we want to recognize the empty string if we are at a final node:

```
recognize(Node, [], []) :-
     finalnode(Node).
```

This clause says that recognize (NODE, IN_STRING,OUT_STRING) will succeed when string is empty and node is final. Secondly, we need to make provision for a traversal of an arc leading from the node we are at, followed by a continuation of the traversal from that point:

```
recognize(Node, String_In, String_Out): -
     arc(Node, Destination_Node, String_In, String_Out),
recognize(Destination_Node, String_Out, String_Out).
```

Finally, in checking recognition of a whole string, we need to make sure that we start out from an initial node, and we can put this requirement in a simple predicate called *morph*, which will call the recognize predicate:

```
morph(String_In, String_Out): -
     initial (node),
   recognize (Node, String_In, String_Out).
```

3. 2. 3 Indications of Inflectional symbols

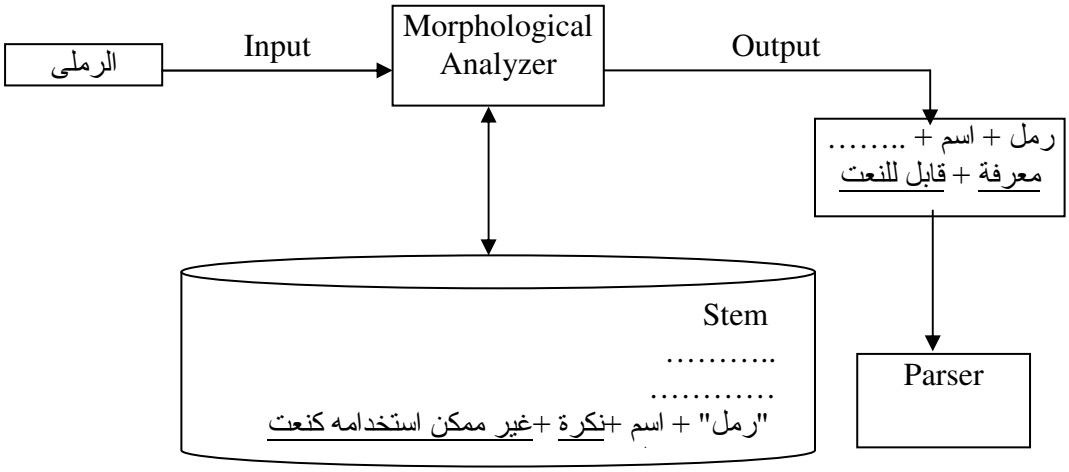
The meaning of inflections joined to the stem is very important in the understanding of the inflected Arabic word. Indications of inflectional symbols can be classified into 2 categories:

(1) *Morphological features*

Some additions have features that override the lexicon features.

Example 1:

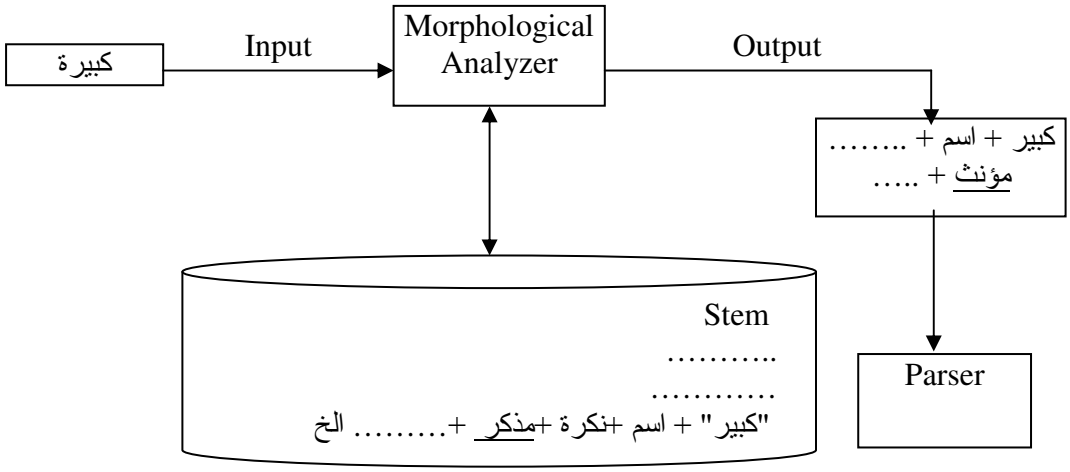
Consider the break downs of the inflected word "الرملى" → (ال + رمل + ى)



In this case we can show that, the morphological analyzer extract the input word (الرملي) into (Begin = ال , Stem = رمل , Last = ي). From the Last (ي), the input word is recognized as an adjective.

Example 2:

For example Consider the break downs of the inflected word كبيرة → (كبير + ة)



In this example, the same action of the previous example taken, but over the Gender feature (التذكير و التأنيث).

(2) Grammatical Features :-

Some additions carry grammatical value to the parsing phase, that affects the building of the grammatical structure of the input sentence.

Example 1:

For example consider the break downs of the inflected word "ضربوا" → (ضرب + وا)

the suffix (وا) means that , the (فاعل) of the sentence is the suffix (وا) and the (فاعل) will not be appear exiplicity in the following words in the sentence .

Example 2:

Consider the break downs of the inflected word "يتخللها" → (يتخلل + ها)

The suffix (ها) means that, the (مفعول) of the sentence is the suffix (ها)

Example 3:

Consider the break downs of the inflected word "يضرب" → (ي + ضرب)

The prefix (ى) means that, the gender of the subject of the sentence must be male

Example 4:

Consider the break downs of the inflected word "تضرب" → (ت + ضرب)

The prefix (ت) means that, the gender of the subject of the sentence must be female.

3. 2. 4 Resolving Morphological Ambiguity

Sometime the morphological analyzer produces erroneous alternatives that can be rejected. For example: -

(ى + قسم (فعل) ← " يقسم "

(ى + قسم (اسم) ← " يقسم "

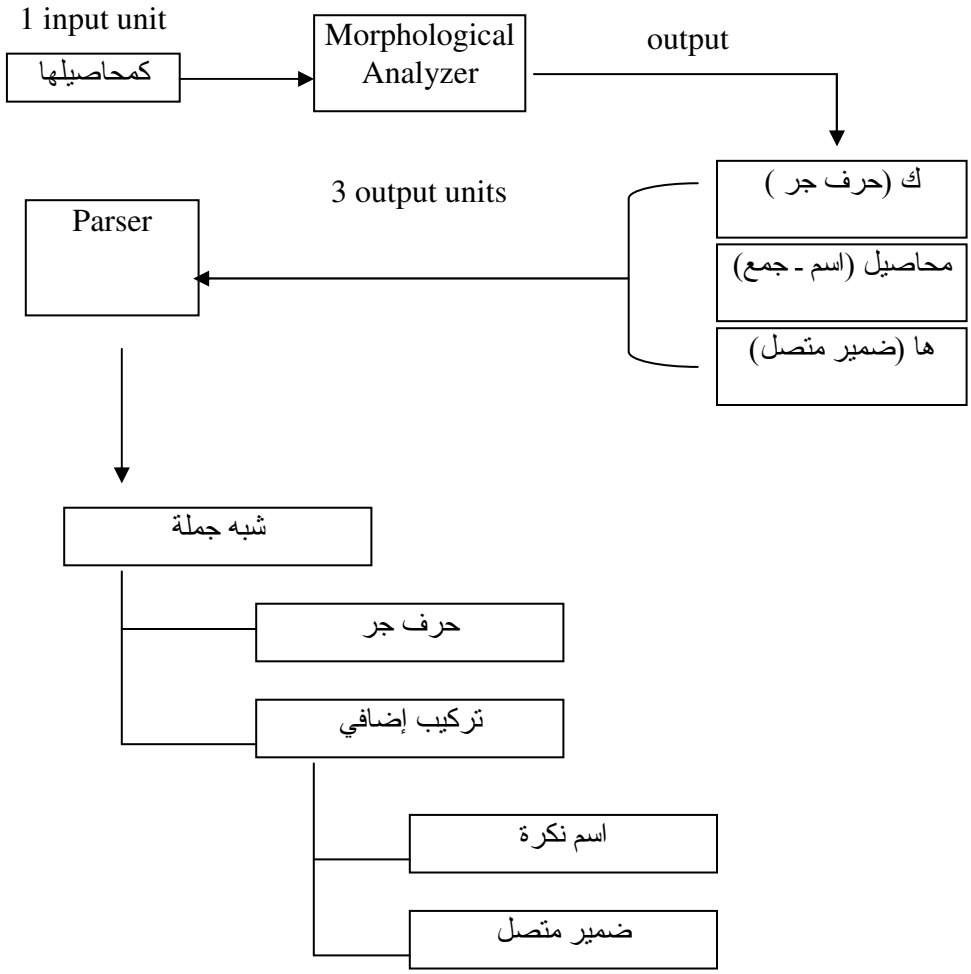
The second alternative is not accepted because cannot be attached to a noun.

3.2.5 The structure passed to the parser

For each morphological analyzer input, we need to pass the output in a structure, i.e. a Prolog list. This structure contains information such as *stem category, definite, verb*

transitive, gender, … etc. For parsing purposes, a post processing is called in order to recognize the parts of the inflected word as individual syntactic constituents of the parser input. For example

ك + محاصيل + ها←كمحاصيلها



CHAPTER (4)
SYNTACTIC PROCESSING OF ARABIC

Parsing Arabic sentences is a difficult task. The difficulty comes from several sources. One is that sentences are long and complex. The average length of a sentence is 20 to 30 words, and it often exceeds 100 words. Another difficulty comes from the sentence structure. The Arabic sentence is complex and syntactically ambiguous due to the frequent usage of grammatical relations, order of words and phrases, conjunctions, and other constructions.

This chapter presents our attempt in developing an Arabic Parser for modern scientific text. The parser is written in Definite Clause Grammar (DCG) and is targeted to be part of a machine translation system. The developing of the parser was a two-step process. In Section 4.1 we will present the architecture of the proposed Arabic parser. In Section 4.2, we describe how we acquired the rules that constitute a grammar for Arabic that gives a precise account of what it is for a sentence to be grammatical. The grammar rules are acquired from an agricultural text. In our approach, we analyzed the text and identified the different patterns of a sentence. With the help of an Arabic linguistics, we derived the Arabic grammar rules from these patterns. Our objective was to formulate the grammar of sentences in the agricultural domain. In Section 4.3, we discuss the implementation aspects of the parser and show how the parser assigns grammatical structure onto input sentence.

4.1 Overall Structure of the Proposed Arabic Parser

Our main goal was to implement a computer system to parse Arabic sentences. The architecture of the system is given in Figure 4-1. In this architecture, the arrows indicate the flow of information. Boxes are modules of the system. As shown, the parser will be fed by the output of the morphological analyzer. The output of the morphological analyzer consists of the words of the input sentence along with their features. These features are either retrieved from the lexicon or recognized during the morphological analysis.

As shown the parser output is a parse tree. A Parse tree is a useful guide for proving the derivation of a sentence for a grammar. Arabic grammarians analyze sentences by identifying the relationships that connect one word to another, called "Irab" (إعراب) , rather than by drawing a hierarchical representation. For example, the sentence " تسبب الإصابة الشديدة تهتك في الأنسجة", is analyzed by the grammarian as

"تسبب" " is the main verb (الفعل)

"الإصابة الشديدة" is the Subject (الفاعل) of the "تسبب"

"تهتك" is the Direct object (مفعول به) of the "تسبب"

"في الأنسجة" is the complement of the "تسبب"

"الأنسجة" is the object of the preposition "في"

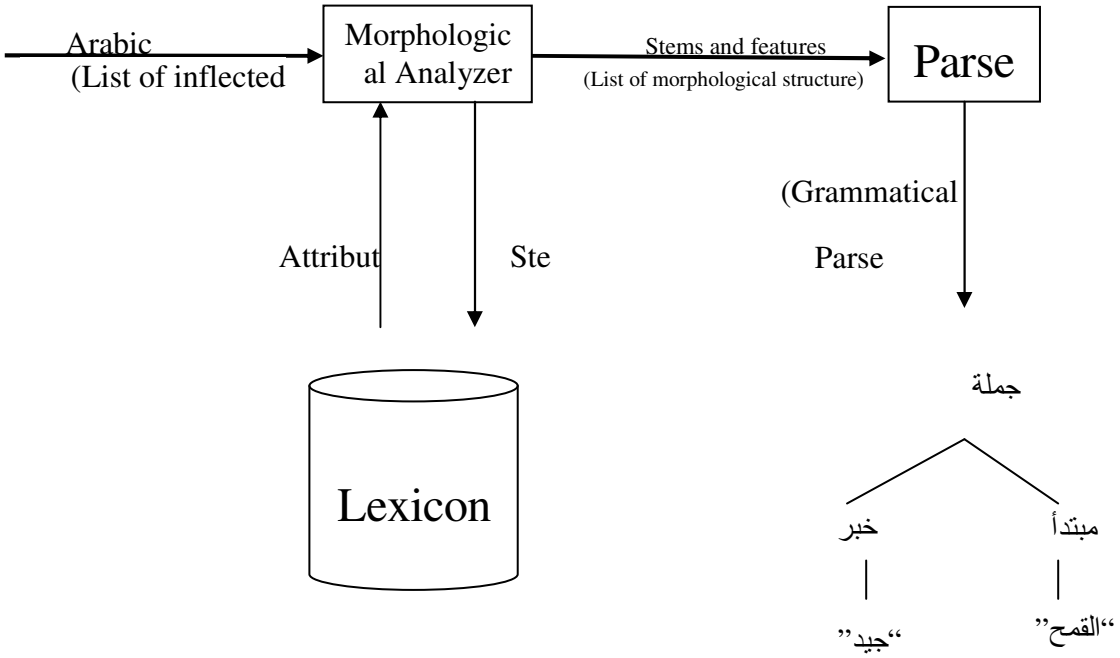


Fig. 4-1 The Parser architecture

4.2 The grammar

The derived grammar for Arabic contains the syntactic knowledge required to analyze a modern scientific sentence. Ideally, in order to construct such a grammar one would take a general-purpose computational grammar of Arabic, and adapt it to the current domain

and application. Unfortunately, however, we do not know of any computational grammar for Arabic that could easily be adapted to the present task.

The grammar is being developed especially for the purposes of understanding scientific Arabic text. On the one hand this has the advantage that the grammar can be tailored to the specific requirements of the scientific domain. On the other hand, we want to adopt general solutions as much as possible, as this increases the chances that the grammar can be used in other domains as well. Thus, in designing the grammar we seek a balance between short-term goals (a grammar which covers sentences typical for the scientific domain and is reasonably robust and efficient) and long-term goals (a grammar which covers the major constructions of Arabic in a general way).

The grammar currently covers sentences that fall into one of two categories: simple sentence and compound sentence. A simple sentence is a sentence, which is not connected by any means with another sentence. However, it may embed another sentence. A compound sentence is more than a simple sentence connected with a conjunction article (أداة عطـــف). Simple sentences are classified into three classes: nominal sentence, verbal sentence, and special sentences. Special sentences are either the special verbs (Kana and its sisters كان و أخواتها), or special particles (‘Inna and its sisters أن و أخواتها).

This section provides focusing on some solutions of practical linguistics problems. At first, we start our discussion with the grammar constituents. Next, we will discuss the grammar of sentences. Finally, we end with the terminal units that are used in the grammar.

4.2.1 Grammar Constituents

Grammar gives two kinds of information about sentence—it divides it into **constituents** and classifies these constituents into **categories** such as *Inchoative* (مبتدأ), *Enunciative* (خبر). We have classified the grammar into 11 main groups, see Fig. 4-2.

In this grammar, constituents can be in one of three styles: *simple*, *contiguous*, and *connected* styles. A simple constituent is a stream of units (terminal or non-terminal) that are **not** separated by any connector. For example consider the grammar rules that defines the simple Annexation constituent:

annexation-constituent → *indeterminate* + *definite*

annexation-constituent → *indeterminate* + adjective-constituent

Which accept the following constituents

"إنتاج الفاكهة الحمضية " ,"زراعة القمح"

A contiguous constituent, sometimes called Multiple *unit* position in [Convigton, 1994], in this case, is built out of some units that can be recursively-repeated n-times. In the following examples

"زيادة كمية مياه الري" , "معظم أراضي الجمهورية" , "استعمال آلة التسطير"

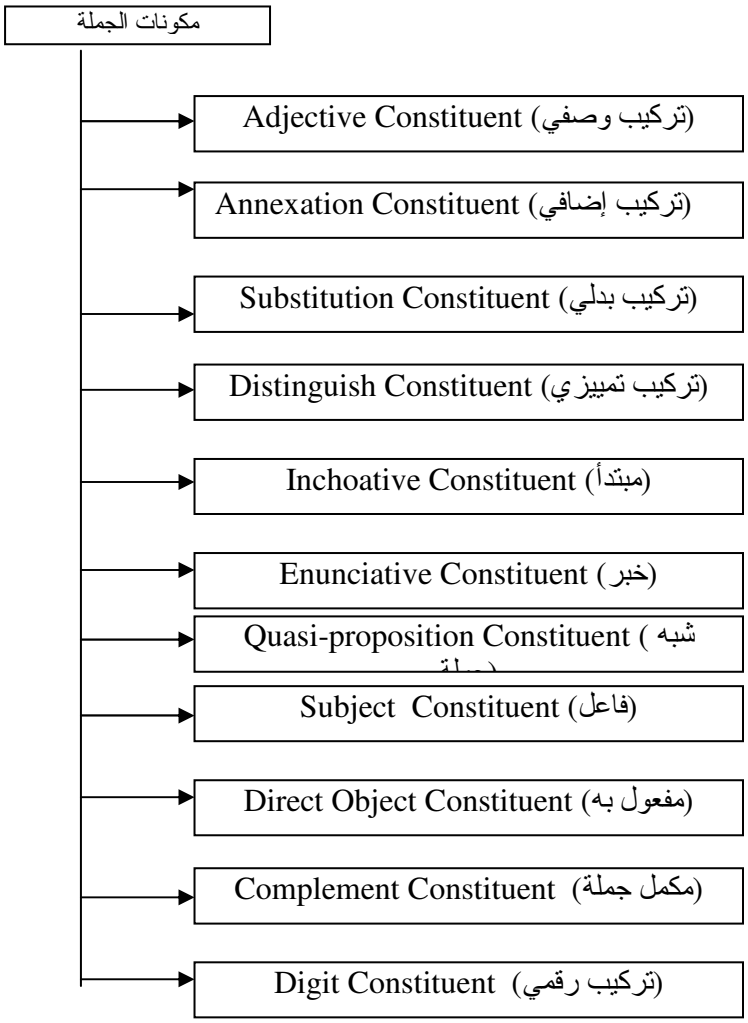


Fig. 4-2 Arabic Grammar Constituents

We can observe the repetition of the *indeterminate*(s) before the occurrence of the *definite* , the grammar need to declare a constituent that accepts those repeated units (indeterminate, in this example), as follows

```
contiguous-indeterminate → indeterminate +
                                    indeterminate

contiguous-indeterminate → indeterminate +
                                    contiguous-indeterminate
```

then, we add the following rule to define the annexation-constituent to be applicable to the above phrases

annexation-constituent → contiguous-indeterminate + *definite*

Notice that we used the recursive rules in the grammar to accept any number of *indeterminate units* as they appear in the input.

The last style of the constituent in our grammar is the connected constituent. A connected constituent is a constituent attached with another unit (terminal or non-terminal) by a connector. The following grammar rule represents a connection between a constituent and a terminal unit.

connected-annexation-constituent → annexation-constituent + *connector* + *definite*

For example, "إنتاج الفاكهة و الموالح" contains two connected units (constituent and terminal unit). As an example of connecting two constituents, consider the following rule

connected-annexation-constituent → annexation-constituent + connector + annexation-constituent

For example, "زراعة الأراضي و شق الترع" contains two connected units (constituent and constituent).

4.2.2 Simple Sentences

The grammar currently covers sentences that fall into one of two categories: simple sentence and compound sentence. A simple sentence is a sentence, which is not connected by any means with another sentence. However, it may embed another sentence. A compound sentence is the connected one.

Simple sentences are classified into three classes: nominal sentence, verbal sentence, and special verbal sentence. The following subsections describe each class of these sentences.

4.2.2.1 Nominal Sentences

In constructing a set of rules for Arabic sentences, we began with the Nominal sentence, which consists of an *inchoative* followed by *enunciative*. The simple rule for nominal sentence in our grammar is the following

```
Nominal-sentence → inchoative + enunciative
```

Table 4.1 shows examples of those sentences.

Table 4.1 Examples of simple nominal sentences

المبتدأ	الخبر
زراعة القمح	جيدة
جيزة167	من الأصناف الجديدة
انتظام توزيع التقاوي	في الحقل

Another rule is when inchoative occurs at the end of the sentence, which entails that the enunciative must be *Quasi-proposition Constituent.* This is defined as follows

```
Nominal-sentence → Quasi-proposition Constituent +
Inchoative
```

This leads us to the general rule of Nominal sentence

```
Nominal-sentence → Quasi-proposition Constituent +
                   Inchoative + complement
```

Table 4.2 shows examples of these sentences.

Table 4.2 Examples of complex Nominal sentences

شبه جملة(خبر مقدم)	المبتدأ	مكمل جملة
من مميزات الزراعة بآلة التسطير	توفير وقت الزراعة و نفقات العمالة اليدوية	
من الضروري	العمل	على مقاومة الحشائش في حقول القمح
فمن المهم	اتباع التوصيات الفنية الخاصة	بإنتاج القمح في تلك المناطق بكل دقة و بقدر الإمكان للتوصل إلي محصول جيد

During the course of building the grammar, we analyzed some nominal sentence, which contain embedded sentences as the enunciative of the whole sentence. We put two conditions to accept this style of nominal sentences. First, when the embedded sentence is a nominal sentence, it must contain a pronoun (ضمير) either connected (متصل) or

separated (منفصل). Second, when the embedded sentence is a verbal sentence, it must its subject (فاعل) absent. Table 4.3 shows examples of these sentences.

Table 4.3 Examples of embedding a sentence into a nominal sentence

المبتدأ	مكمل جملة	الخبر (جملة اسمية أو فعلية)
القمح	في هذه الأراضي	يتعرض لظروف غير ملائمة تبعا لنوعية التربة و قلة خصوبتها و قلة احتفاظها بمياه الري
الصنف جميزة3		يشترط عدم زراعته مبكرا
أفضل ميعاد	للزراعة	هو الفترة من 15 إلي 30 نوفمبر

4.2.2.2 Verbal Sentences

In our grammar, the verbal sentence fall into one of two classes *simple* and *prefixed.* The simple form starts by the verb directly. The simple verbal sentences components are: verb, subject (الفاعل), direct object (المفعول بـه), and complement (مكمل الجملة). All these components, except the verb, can be missed. The grammar rules handle these possibility. The prefixed Verbal sentence may not start directly by the verb. The verb must occur somewhere in the sentence. Table 4.4 and table 4.5 show examples of the verbal sentence.

Tables 4.4 Examples of Simple Verbal Sentences

فعل	مكمل جملة	*فاعل*	مكمل جملة	*مفعول به*	مكمل جملة
تلقى				*قبولا كثيرا*	لدى زراع القمح في الأراضي الجديدة
يجود					في الأراضي الجديدة
ترتفع	بها	*نسبة الملوحة*			
تستجيب	إليه	*حشرات المن*	أثناء الطيران		
ينصح			في الأراضي الموبوءة بالقواقع	*زراعة داير*	من البرسيم حول حقل القمح
يزرع				*هذا الصنف*	في معظم أراضي الجمهورية
تحرث				*الأرض*	باستخدام المحراث الحفار
يعمل		*السماد البلدي*	على تحسين خواص التربة الطبيعية		
يشترط			في السماد البلدي	*أن يكون من مصـــدر موثوق به*	
يميل		*اللون*	قليلا إلي الزرقة		
يروى		*القمح*	في الأراضي الرملية على فترات من 5-10 أيام		
يعتبر		*العلاج*	بعد طرد السنابل	*عديم الجدوى*	

Tables 4.5 Examples Prefixed Verbal Sentences

مكمل جملة	مفعول	مكمل جملة	فاعل	مكمل جملة	فعل	مكمل جملة
للزراعة بمعدل 60 كيلو جرام للفدان في سطور على مسافات 12,5-13 سم و عمق 4-5 سم	السطارة				تعاير	فيها
للفدان على أن تبذر بانتظام على الأرض مع التغطية الجيدة	75 كيلو جرام تقاوي				يستخدم	في حالة الزراعة العفير بدار
	ها (يتخللها)		محصول البرسيم		يتخللها	اتباع دورة زراعية

4.2.2.3 Special Verbal Sentences

Tables 4.6 Examples of Kana and 'Inna Sentences (Special Verbal sentence)

مكمل جملة	خبر	مبتدأ	ضمير متصل	أخوات ***كان*** أو ***أن***
	مبكر التزهير و مبكر النضج		الهاء في (إنه)	إنه
للرطوبة و زيادة كمية مياه الري عند الزراعة	حساس	هذا الصنف		أن
	يكون متجانسا في الحقل (الخبر جملة أخوات كان)			أن
	يجرى الرش بعد تطاير الندى (الخبر جملة فعلية)			أن
	تكون مواعيد الزراعة في الحقول المجاورة متقاربة لتقليل أضرار مهاجمة العصافير مع إزالة الأعشاش (الخبر جملة أخوات كان)			أن
من مسببات الأمراض و يرقات الحشرات و النيماتود	خاليا			يكون
	فعال	تأثير الرش		يكون
على الأشجار المحيطة للحقل	متواجدة			كانت

Recall that the Arabic has special verbal sentence. These sentences start with either the special verbs (Kana and its sisters كان و أخواتها), or with special particles (‘Inna and its sisters أن و أخواتها). Special sentences has special forms, they have no *subject* (فاعل) nor *direct object,* but they have *noun*, and *enunciative* of the sentence. In our grammar we assume that the special verbal sentence starts with one of its instances (Kana or Inna sisters), followed by a nominal sentence. This nominal sentence can be a complete sentence or may miss its *inchoative* (مبتدأ). Also we set another assumption, when the starting special verb is attached with *connected pronoun* (ضمير متصل), then it is followed by the nominal sentence that missed its *inchoative.* Table 4.6 shows examples of Kana and Inna sentences.

4.2.3 Compound Sentences

The compound sentence is one, which connects more than one sentence together Table 4.7 shows examples of compound sentences.

Table 4.7 Compound Sentences

جملة اسمية	حرف وصل	جملة اسمية
انتظام توزيع التقاوي في الحقل	و	انتظام عمق الزراعة و ضمان تغطية الحبوب عقب الزراعة
بمعدل 60 جم/فدان رشا WP % سنكور 70 في طور 2-4 أوراق للقمح	و	هذان المبيدان يستخدمان ضد الحشائش عريضة الأوراق فقط
تكبير المساحات المنزرعة بالقمح	و	تجميعها و توحيد ميعاد الزراعة بقدر الإمكان

4.2.4 Terminal Units

Terminal units are the words that actually occur in the language, such as ‘قطة’and ‘رأى’. They appear only on the right-hand side of the grammar rules. That is, they are not expanded any more. Non-terminal units are defined in terms of terminal units either directly or indirectly. If non-terminal units appeared only on the left-hand side of the grammatical rule, then, they are defined directly by the terminal units on the right hand side. If they appeared on both sides, then, the non-terminal units on the left-hand side are defined indirectly in terms of the terminal units to which the right-hand side non-terminal nodes are expanded in other rules. Every non-terminal has to expand finally to terminal units. Terminal units examples in the proposed grammar are indeterminate, verb, and

definite. Code of the Arabic parser include the rules of all terminal units, See Appendix (E).

4. 3 Parsing

From a linguistic perspective, the current grammar can be characterized as a constraint based grammar, which makes heavy use of lexical information. The design of the grammar was inspired to a certain extent by Head-Driven Phrase Structure Grammar (HPSG) (Pollard and Sag, 1994).

4. 3. 1 The DCG

The grammar formalism is essentially equivalent to Definite Clause Grammar (DCG) (Pereira, Shieber, and David, 1986). The choice for DCG is motivated by the fact that this formalism provides a balance between computational efficiency and linguistic expressiveness, and the fact that it is closely related to constraint-based grammar formalisms (Bouma et al., 1996).

Prolog interpretation algorithm uses exactly the same search strategy as the depth-first top-down parsing algorithm, so all it is needed is a way to reformulate grammar rules as clauses in Prolog. For more than a decade, definite clause grammars (DCGs) notation was developed as a result of research in natural language parsing and understanding understanding (Pereira, Shieber, and David, 1986). DCGs allow one to write grammar rules directly in Prolog. Prolog that conforms to Edinburgh standard have DCGs as a part of their implementations. In the current system, grammar rules of Arabic are written in DCG formalism, which translated into an executable code in Prolog.

A DCG rule is translated into a Prolog clause according to the following basic scheme:
let the DCG rule be:

```
n --> n1, n2, … nn.
```

If all n1, n2, …, nn are non-terminals then the rule is translated into the clause:

```
n(List1, Rest) :-
     n1(List1, List2),
     n2(List2,List3),
```

```
    …
    nn(Listn, Rest).
```

If some of `n1,n2,…,nn` are terminals (in square brackets in the DCG rule) then it is handled differently. It does not appear as a goal in the clause, but is directly inserted ino the corresponding list. As an example, consider the DCG rule:

```
n --> n1, [t2], n3, [t4].
```

Where **`n1`** and **`n3`** are non-terminals, and **`t2`** and **`t4`** are terminals. This is translated into the clause:

```
n(List1, Rest) :-
    n1(List1, [t2|List3]),
    n3(List3,[t4|Rest]).
```

Here is an example grammar for a simple subset of nominal sentence extracted from the grammar:

```
nominal_sentence --> inchoative , enunciative.
inchoative     --> defined.
enunciative    --> indeterminate.
defined --> ['القمح'].
indeterminate --> ['جميل'].
```

The DCG formalism allows us to associate terms and/or logical variables, as many as you want , as arguments to categories. Hence, the following rule is also a legitimate statement in the DCG formalism:

```
nominal_sentence --> inchoative(Number,Gender,Case1),
enunciative(Number, Gender,Case2).
```

This rule claims that a sentence can consist of a inchoative followed by enunciative , where both share the same values for both number and gender

The DCG notation actually makes it possible to incorporate the meaning directly into a grammar. There is another notational extension, provided by DCG, that is useful in this respect. This extension allows us to insert normal Prolog goals into grammar rules. Such

goals have to be enclosed in curly brackets to make them distinguishable from other symbols of the grammar [Bratko, 1990].

The parse tree can be easily constructed using arguments of non-terminals in the DCG notation. We can conveniently represent a parse tree by a Prolog term whose functor is the root of the tree and whose arguments are the subtrees of the tree. For example, the parse tree for the annexation_constituent "زراعة القمح" would be represented as:

```
annexation_constituent(indeterminate('زراعة'),
defined('القمح')).
```

To generate a parse tree, a DCG grammar can be modified by adding to each non-terminal its parse tree as argument [Bratko, 1990]. For example , the parse tree of an annexation_constituent in our grammar has the form:

```
annexation_constituent(IndeterminateTree,
DefinedTree).
```

Here *IndeterminateTree* and *DefinedTree* are the parse trees of an indeterminate and defined respectively. Adding these parse trees as arguments into our annexation_constituent grammar rule results in the modified rule:

```
annexation_constituent(annexation_constituent
(IndeterminateTree, DefinedTree)) -->
     indeterminate(IndeterminateTree),
     defined(DefinedTree).
```

This rule can be read as:

An annexation constituent whose parse tree is *annexation_constituent (IndeterminateTree, DefinedTree)* consists of:

- Indeterminate noun whose parse tree is *IndeterminateTree* , and
- Defined noun whose parse tree is *DefinedTree.*

4.3.2 Feature-Structures

During the construction of the Arabic parser, feature-structures are translated into Prolog terms. Because of this translation step parsing can make use of Prolog's built-in term. The following is a description of these features:

Tree of unit:

String carries the tree of the constituent.

Gender:

Variable stores the gender of the unit (feminine and masculine),

Note that, both verbs and nouns can carry gender indicators.

Number:

Variable stores the number of the unit (singular, dual and plural).

Definiteness:

Variable stores the definiteness of the unit (definite and indefinite),

Note that, both noun units and constituents units (adjective constituent) can carry definiteness indicators.

Noun Information:

List to carry the attributes of the noun units, this list contains the following:

- SpecialNoun attribute.
- MorphPattern attribute.

Verb Information:

List to carry the attributes of the verb units, this list contains the following:

- Case Sign (علامة إعراب ظاهرة) attribute.
- Transitivity of the verb.
- SpecialVerb (كان و أخواتها) attribute.
- VerbPrefix , store prefix of the verb.
- VerbSuffix , store suffix of the verb.

Last Unit Info:

List to carry the some specific attributes of the last non-terminal unit of the constituent, this list contains the following:

- Unit Type (noun, verb , article, annexation constituent … etc.).
- AdjectiveAbility , store boolean value of the ability of the unit to be adjective.

- AnnexationAbility, store boolean value of the ability of the unit to be annexed.

First Unit Info:

List to carry the some specific attributes of the first non-terminal unit of the constituent, This list contains the same attributes in the previous item.

The following examples show the use of these features in DCG rules:

```
inchoative('مبتدأ'(TreeDefined) ,GENDER ,NUM , _Definiteness, NounInfo,
VerbInfo, LastUnitInfo, FirstUnitInfo)-->
      defined(TreeDefined, GENDER , NUM , _Definiteness,
            NounInfo, VerbInfo, LastUnitInfo, FirstUnitInfo).

annexation_constituent('تركيب إضافي' (TreeIndeterminate, TreeDefined)
,Gender ,Number ,Definiteness, NounInfo, VerbInfo,LastUnitInfo,
FirstUnitInfo) -- >
indeterminate(TreeIndeterminate, Gender, Number, _Definiteness1,
NounInfo1, VerbInfo1, LastUnitInfo1, FirstUnitInfo1),
defined(TreeDefined,_Gender2,_Number2, Known2, _Definiteness2,
NounInfo2, VerbInfo2, LastUnitInfo2, FirstUnitInfo2),
      {
      Definiteness = 'معرفة',
      LastUnitInfo = LastUnitInfo2,
      FirstUnitInfo = FirstUnitInfo1,
      NounInfo = NounInfo1,
            VerbInfo = VerbInfo1
            }.
```

4.3.3 Syntactic Disambiguation

Sometimes parsing of the input sentence lead to ambiguous grammatical structures. In our parser, we handle this situation by including either morphological specification constraints or syntactic structure constraints. The former is used in order to specify the agreement constraints, the matching constraints, and the rejection constraints by means of some morphological features. The later is used in order to specify the syntactic constraints among parts of a sentence for disambiguating the parse fragments of the parse

tree. In the following subsections, we discuss the syntactic disambiguation methods used in our parser.

4.3.3.1 Agreement Constraints

If the grammar declare some rule (sentence rule) as a noun phrase followed by verb phrase, then the parser will recognize any occurrence of both as a sentence. This is the case with English and many other languages, the noun phrase and verb phrases in a sentence are not independent: they have to agree in number. This phenomenon is called *context dependence*. Our Arabic parser provides the means to include context-dependency in a grammar.The following is a grammar rule of a verbal sentence:

simple_ verbal_sentence → *verb* + subject + complement + direct-object + complement

This form declares a sequence of the units (terminals or non-terminals) which must occur in order to parse the target sentence, but practically we need some constrains to apply it. This concerns the agreement condition between the *gender of* the *subject* and the *verb* of the sentence. This is handled in our parser by the unification of feature-structure, as follows.

```
verbal_sentence(…)-->
verb(…, GENDER, …, …, …),
subject(…,GENDER, …, …, …),
complement(…),
direct_object(…),
      complement(…).
```

When parsing the following sentences, the first sentence is accepted because the gender feature agrees (unifies) with the subject (الفلاحين) and the verb (يزرع). Whereas, they are not agree in the second sentence.

"يزرع الفلاحون في الغرب القمح بكميات كبيرة"

"تزرع الفلاحون في الغرب القمح بكميات كبيرة"

4.3.3.2 Matching Constraints

There is another type of context dependencies problem, which concerns the matching of some named features in order to determine the correctness of the parse fragments of the parse tree, which meets a specific syntactic structure of parts of a sentence. As an example, consider the case of two contiguous *indeterminate* (نكرة) as *distinguish constituent* (تركيب تمييزي) such that the first indeterminate should its morphological pattern (وزن صرفي) feature unified with (أفعل). This is handled in our parser by the use of the unification of feature-structure, as follows.

```
distinguish_constituent (……)-->
      indeterminate(…, …, …, "أفعل", …, …),
      indeterminate(…, …, …, _MorphPattern2, …, …).
```

So, the following fragments of a sentence are recognized as distinguish constituents.

"أفضل طريقة" "أكثر شمولا"

4.3.3.3 Rejection Constraints

Sometimes we need to reject the parse fragment based. As an example, consider the grammar rule that recognizes an annexation constituent as an subject of a verbal sentence:

```
subject(…, …, …) -->
      annexation_constituent(…).
```

If we would like this rule rejected when the annexation constituent begins with an accusative time or place (ظرف زمان أو مكان), then the rule would be written to include the predicate that does the rejection.

```
subject(…, …, …) -->
      annexation_constituent(…, FirstUnitInfo, …, …, …),
            {
            reject_type(FirstUnitType,'ظرف')
            }.

      reject_type(UnitInfo,RejectedType) :-
                  UnitInfo = [Type | _Tail],
                  Type \== RejectedType.
```

4.3.3.4 Connection Constraints

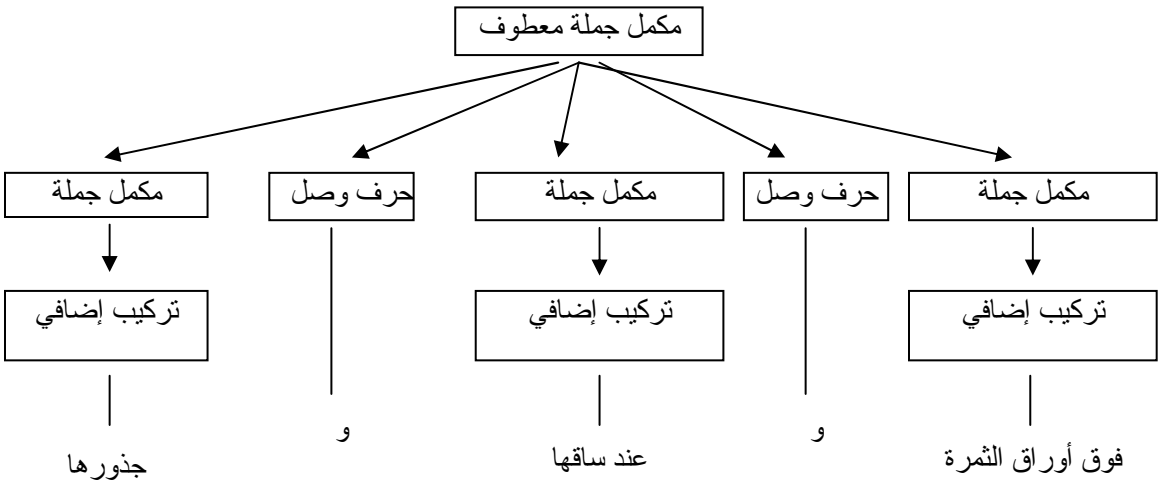
Sometimes we need to enable or disable some features in the grammar in order to disambiguate sentences during parsing as early as possible. For example, if we try to analyze the sentence

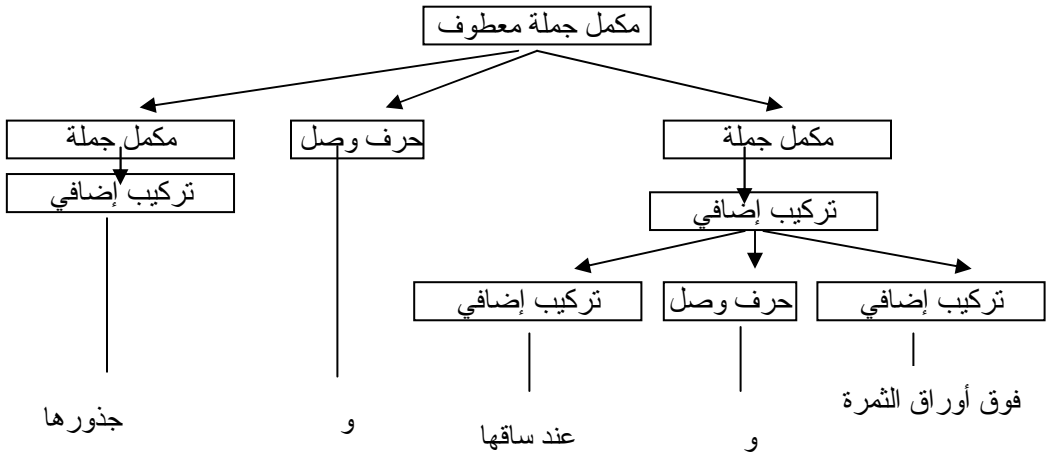
"تظهر البقع **فوق أوراق الثمرة** و **عند ساقها** و **جذورها**"

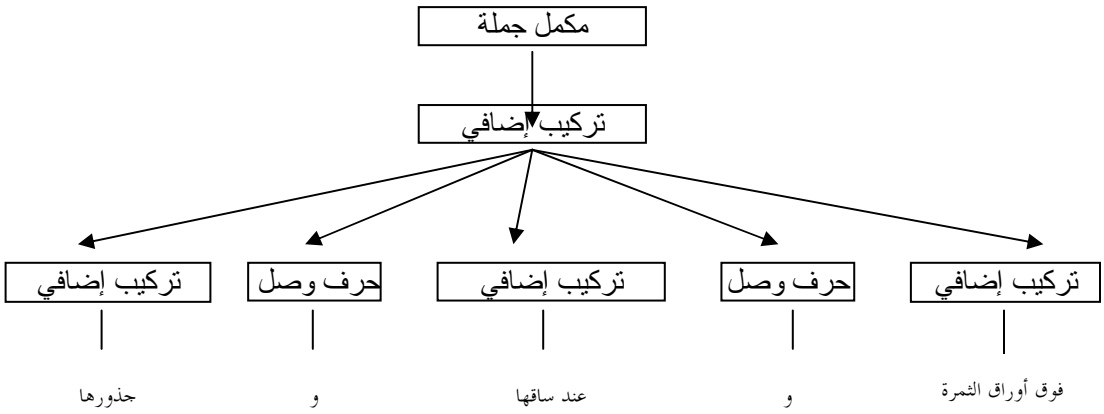
With the following grammar

```
verbal_sentence(…) -->
      verb(…),
      subject(…),
      complement(…).

      complement(…) -->
            annexation_constituent(…),
            connector(…),
            annexation_constituent(…).
      complement(…) -->
            annexation_constituent(…),
            connector(…),
            complement(…).

      annexation_constituent(…) -->
            indeterminate(…),
            defined(…).
      annexation_constituent(…) -->
            indeterminate(…),
            connected_pronoun (…).
      annexation_constituent(…) -->
            accusative_time_or_place(…),
            annexation_constituent(…).
      annexation_constituent(…) -->
            annexation_constituent(…),
            connector(…),
            annexation_constituent(…).
```

We obtain the following parse trees for the complement of the sentence:


مكمل جملة معطوف
مكمل جملة
حرف وصل
مكمل جملة
حرف وصل
مكمل جملة
تركيب إضافي
تركيب إضافي
تركيب إضافي
جذورها
و
عند ساقها
و
فوق أوراق الثمرة


مكمل جملة معطوف
مكمل جملة
حرف وصل
مكمل جملة
تركيب إضافي
تركيب إضافي
تركيب إضافي
حرف وصل
تركيب إضافي
جذورها
و
عند ساقها
و
فوق أوراق الثمرة


مكمل جملة
تركيب إضافي
تركيب إضافي
حرف وصل
تركيب إضافي
حرف وصل
تركيب إضافي
جذورها
و
عند ساقها
و
فوق أوراق الثمرة

The disambiguation arises because the second grammar rule of the annexation_constituent does a redundant work in recognizing the connection cases when it is called from a complement rule. To disambiguate these trees, we have to eliminate the other two trees. This can be done by setting an explicit feature—acts as a constraint—that disambiguates the connection case of the annexation rules. Consequently, the complement rule is defined as

```
complement(...) -->
      annexation_constituent(…, no_connection),
      connector(…),
      annexation_constituent(…, no_connection).
complement(...) -->
      annexation_constituent(…,no_connection),
      connector(…),
      complement(...).
```

The result will be only one tree, the first one.

4.3.3.5 Decomposition Constraints

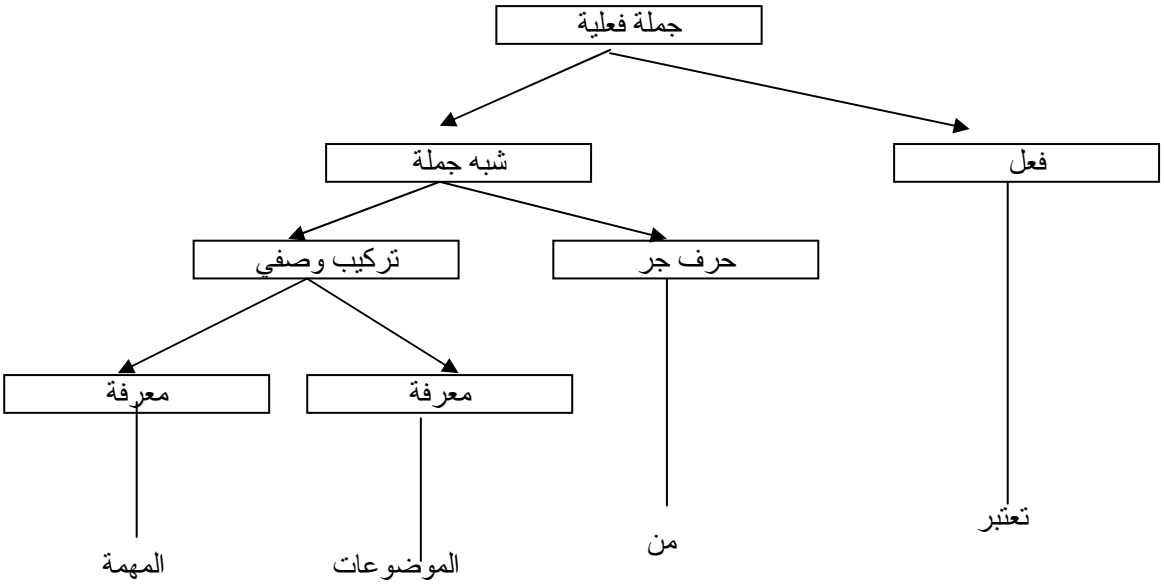
Sometimes we obtain undesired parse trees, because the parser decomposes a single grammatical constituent into some other separated ungrammatical constituents. For example, consider the following sentence, and its corresponding grammar:

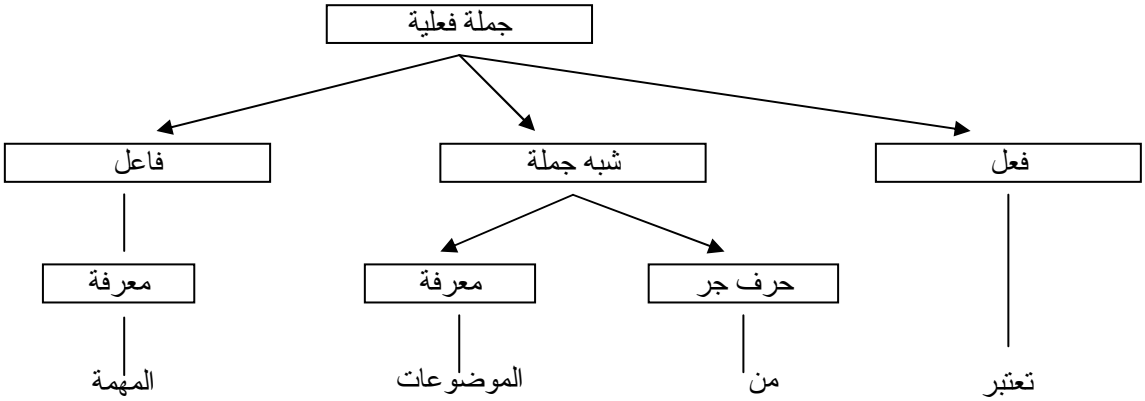
"تعتبر من الموضوعات المهمة"

```
verbal_sentence(…) -->
      verb(…),
      complement(…),
      subject(…).
verbal_sentence(…) -->
      verb(…),
      complement(…).

complement(…) -->
      quasi-proposition(…).

quasi-proposition(…) -->
      preposition(…),
      defined(…).
```

```
quasi-proposition(…) -->
      preposition(…),
      adjective_constituent(…).

adjective_constituent(…) -->
      defined(…),
      defined(…).
```

In analyzing this sentence, we obtain the following two parse trees:





This ambiguation problem comes from the recognition of the defined noun (المهمة) as a separate part from the defined noun (الموضوعات), where they should altogether be drawn directly from the same grammatical entity. This is resolved during parsing by adding a disambiguation rule that handles this undesirable situation. The rule should prevent the generation of the parse tree when it is possible to draw an incorrect parse fragment. Applying this solution to the verb_sentence rule given above, we rewrite it as follows

```
verbal_sentence(…) -->
      verb(…),
      complement(…, LastUnitInfo1, FirstUnitInfo1),
      subject(…, LastUnitInfo2, FirstUnitInfo2),
      {
      decompose_validate(LastUnitInfo1,FirstUnitInfo2).
      }.
```

All we need is to implement the Prolog predicate that validates the decomposition rejection, this predicate matches all the cases of two contiguous parse fragments that must be rejected. The following disambiguation rule handles the case of ("معرفة + معرفة") of decomposition rejection.

```
decompose_validate(LastUnitInfo, FirstUnitInfo) :-
      (     ((is_unit_type(LastUnitInfo, 'معرفة')) ,
            (unite_adverbed(FirstUnitInfo)))->
                  reject_type(FirstUnitInfo,'معرفة')
            ;
                  true
      ).

is_unit_type(UnitInfo, TargetType) :-
      UnitInfo = [TargetType | _Tail].

unite_adverbed(UnitInfo) :-
      UnitInfo = [_Type, 'نعت', _AnnexationAbility].

reject_type(UnitInfo,RejectedType) :-
      UnitInfo = [UnitType | _Tail],
      UnitType \== RejectedType.
```

4.3.4 Improving Parsing Inefficiency

In this section, we discuss our ways in handling inefficiency. Also, we propose suggestions for future improvements.

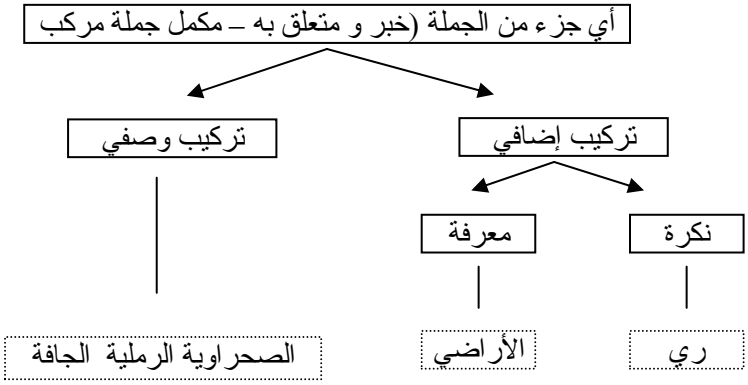
4.3.4.1 Controlling Backtracking

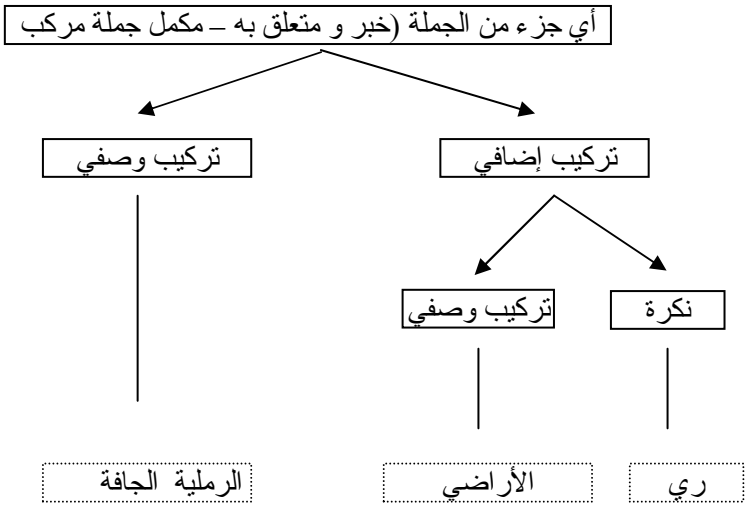
Prolog will automatically backtrack if this is necessary for satisfying a goal. Automatic backtracking is a useful programming concept because it relieves the programmer of the burden of programming backtracking explicitly. On the other hand, uncontrolled backtracking may cause inefficiency in a program. Therefore we sometimes want to control, or to prevent, backtracking. We can do this in Prolog by using the 'cut' facility, which has that syntax "**!**". As an applied example in our parser, assume the following sentence and its corresponding grammar:

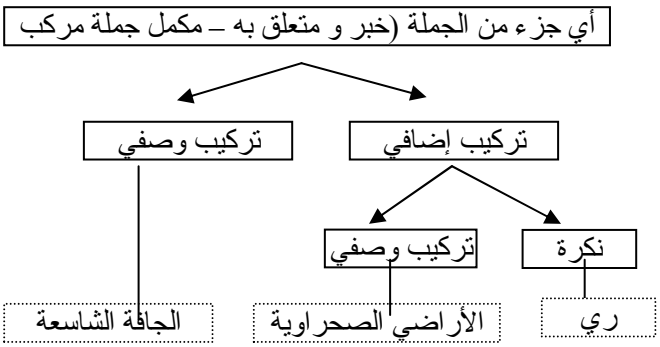
"...ري الأراضي الصحراوية الرملية الجافة الشاسعة"

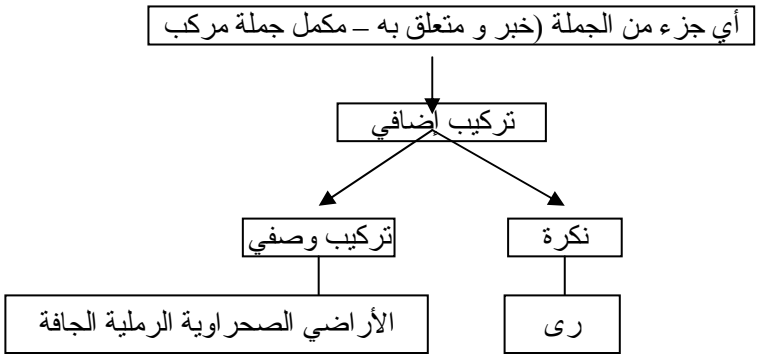
```
annexation_constituent(…) -->
      indeterminate(…) , defined(…).
annexation_constituent(…) -->
      indeterminate(…) , adjective_constituent (…).

adjective_constituent (…) -->
      defined(…), defined(…).
adjective_constituent (…) -->
      defined(…),adjective_constituent (…).
```

In this case, the parser will recognize this constituent and produces the following parse fragments:


أي جزء من الجملة (خبر و متعلق به – مكمل جملة مركب
تركيب وصفي
تركيب إضافي
معرفة
نكرة
الصحراوية الرملية الجافة
الأراضي
ري


أي جزء من الجملة (خبر و متعلق به – مكمل جملة مركب
تركيب وصفي
تركيب إضافي
تركيب وصفي
نكرة
الرملية الجافة
الأراضي
ري


أي جزء من الجملة (خبر و متعلق به – مكمل جملة مركب
تركيب وصفي
تركيب إضافي
تركيب وصفي
نكرة
الجافة الشاسعة
الأراضي الصحراوية
ري



The last parse tree is the desired one. In this case we face two problems. First, the ambiguity of rules that can be solved by adding an action to the rule to reject the decomposition of the unit. Second, the problem of backtracking which wasts some time in solving some parts of sentence which are incorrect grammatically. To solve this wasted backtracking time, we followed two steps: 1) if there are more than one form of the non-terminal unit , then we wrote the *longest* match one (form) at the first , and the *shorter* form(s) in the last., and 2) Stop backtracking (using cut facility) , if we success to get longest match form. Applying this solution to the above rules, we rewrite them as follows

```
annexation_constituent(…) -->
      indeterminate(…) , adjective_constituent (…) , !.
annexation_constituent(…) -->
      indeterminate(…) , defined(…).

adjective_constituent (…) -->
      defined(…),adjective_constituent (…), !.
adjective_constituent (…) -->
      defined(…), defined(…).
```

Notice that this solution is both deterministic and unambiguous. In other words, we consume only the time of the fourth tree, and get only one parse tree, the fourth one.

4.4. Conclusions

This chapter has been concentrated on issues in the design and implementation of an Arabic Parser. The parser encodes the Arabic grammar rules of irab and the effects of

applying these rules to the constituents of sentences. The system can be geared towards any other related system or application since it has been built as a module. The parser tries to avoid some problems associated with ambiguities. The extent to which ambiguity could be resolved is not comprehensive since ambiguity can be semantic which is another research problem.

CHAPTER (5)
THE TESTING OF ARABIC PARSER

In this chapter we report on an experiment that tests the Arabic parser. Results are analyzed and discussed. Suggestions for improvements are also proposed.

In Arabic natural language processing, there are no predefined forms for analyzing the sentence, which makes parsing a difficult task. The syntactic structure of parts of the sentence may be missing, taking different orders, or connected by long distance syntactic relations. Section 5.1 gives some examples of the parser output. In Section 5.2 we discuss the results of running the parser on an agricultural extension document.
At last we will analyze these results in section 5.3.

5. 1. Examples of the parser output

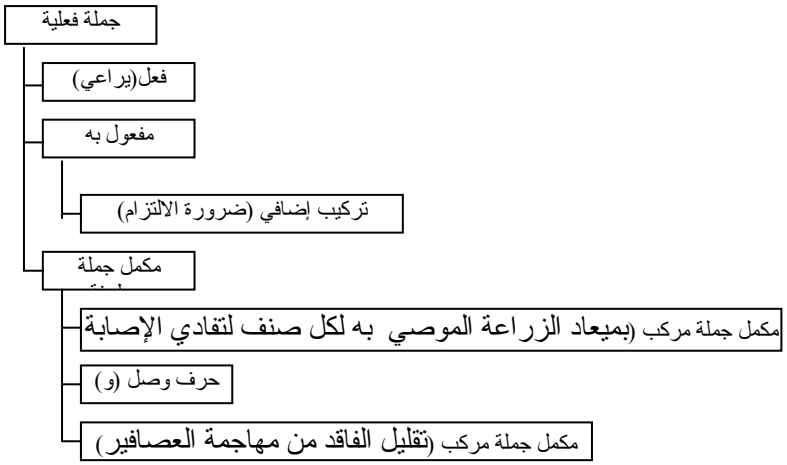
The central task of a parser is to assign a grammatical structure onto input sentences. In practice, it is more common to describe the parser output by means of a parse tree. Parse tree is a useful guide for proving the derivation of a sentence for a grammar. In the following subsection, we give examples that are extracted from the experiment conducted to test our parser. We present each input sentence along with the parse tree that result from the parse. The input sentences are grammatically correct and their parse trees are also successfully produced.

5.1.1. Examples of Verbal Sentences

Example (1): The following sentence shows how the parser is capable to deal with the case when the subject (الفاعل) of the sentence is missing.

يراعى ضرورة الالتزام بميعاد الزراعة الموصي به لكل صنف لتفادي الإصابة بالمن و تقليل الفاقد من مهاجمة العصافير .

The parser produces the following parse tree for the above sentence.

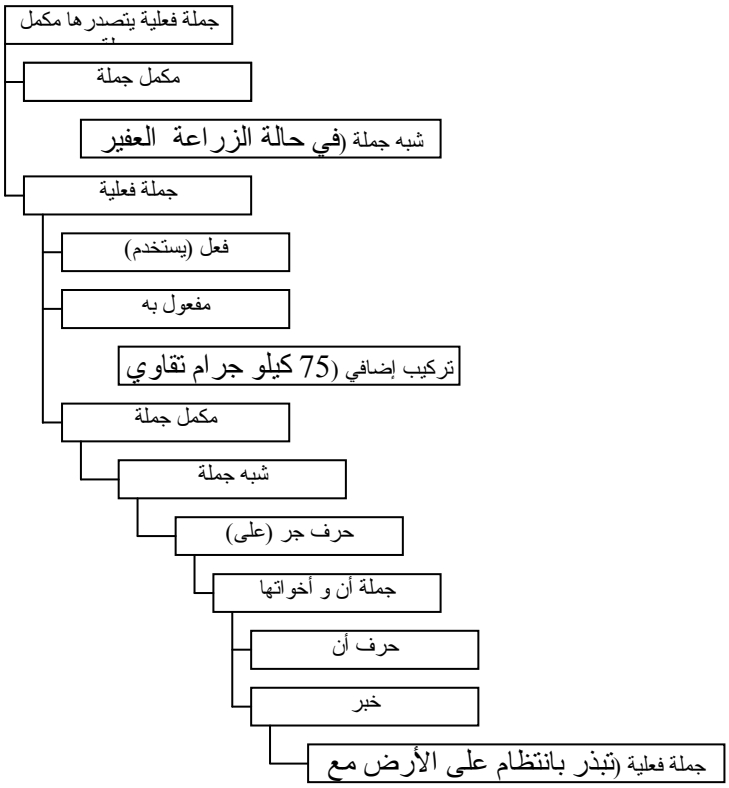


جملة فعلية(فعل(يراعى,متعدي),مفعول به(تركيب إضافي(نكرة(ضرورة,مؤنث),معرفة(الالتزام,مذكر))),مكمل جملة معطوفة(مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(نكرة(ميعاد,مذكر),معرفة(الزراعة,مؤنث)))),مكمل جملة مركب (مكمل جملة(معرفة(الموصي,مذكر)),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),ضمير_متصل(ه))),مكمل جملة مركب (مكمل جملة(شبه جملة(حرف جر(ل),تركيب إضافي(ظرف زمان(كل),نكرة(صنف,مذكر)))),مكمل جملة مركب(مكمل جملة (شبه جملة(حرف جر(ل),تركيب إضافي(نكرة(تفادي,مذكر),معرفة(الإصابة,مؤنث)))),مكمل جملة(شبه جملة(حرف جر (بال) ,معرفة(من,مذكر)))))),حرف وصل(و),مكمل جملة مركب(مكمل جملة(تركيب إضافي(نكرة(تقليل,مذكر) ,معرفة (الفاقد ,مذكر))),مكمل جملة(شبه جملة(حرف جر(من),تركيب إضافي(نكرة(مهاجمة,مؤنث),معرفة(العصافير,مؤنث)))))))

Example (2): The following sentence shows how the parser is capable to deal with the case when the main verb of the verbal sentence may not occur at the beginning of the sentence.

في حالة الزراعة العفير بدار **يستخدم** 75 كيلو جرام تقاوي للفدان على أن تبذر بانتظام على الأرض مع التغطية الجيدة .

The parser produces the following parse tree for the above sentence.



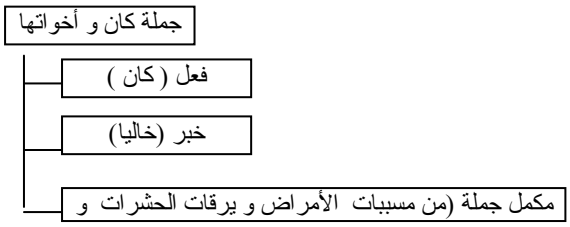
جملة فعلية يتصدرها مكمل جملة مقدم(مكمل جملة(شبه جملة(حرف جر(في),تركيب إضافي(نكرة (حالة,مؤنث) ,معرفة (الزراعة ,معرفة_علم - أعلام متتابعة(العفير,معرفة_علم(بدار,مذكر),مذكر),مؤنث)))),جملة فعلية(فعل(يستخدم,متعدي),مفعول به(تركيب إضافي(وحدة رقمية(75),نكرات متعددة(نكرة(كيلو,مذكر),نكرات متعددة(نكرة(جرام,مذكر),نكرة(تقاوي,مؤنث))))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(لل),معرفة(فدان,مذكر))),مكمل جملة(حرف جر(على),جملة إن و أخواتها(حرف إن (أن),خبر(جملة فعلية(فعل(تبذر,متعدي),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),نكرة (انتظام,مذكر))), مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(على),معرفة(الأرض,مؤنث))),مكمل جملة(شبه جملة(حرف جر(مع),تركيب وصفي(معرفة(التغطية,مؤنث),معرفة(الجيدة,مؤنث)))))))))))

5.1.2. Examples of Nominal Sentences

Example (3): The following sentence shows how the parser is capable to deal with the case when the noun of Kanna sentence was missing.

يكون خاليا من مسببات الأمراض و يرقات الحشرات و النيماتود .

The parser produces the following parse tree for the above sentence.

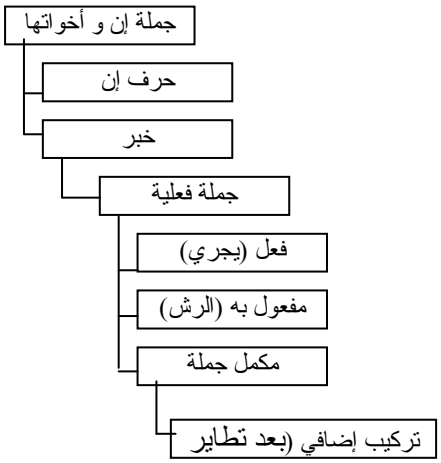


جملة كان و أخواتها(فعل(يكون),خبر(خبر و متعلق به(نكرة(خاليا,مذكر),مكمل جملة(شبه جملة(حرف جر(من),تراكيب إضافية معطوفة(تركيب إضافي(نكرة(مسببات,مؤنث),معرفة(الأمراض,مؤنث)),حرف وصل(و),تراكيب إضافية معطوفة (تركيب إضافي(نكرة(يرقات,مؤنث),معرفة(الحشرات,مؤنث)),حرف وصل(و),معرفة_علم(النيماتود,مذكر)))))))

Example (4): The following sentence shows how the parser is capable to deal with the case when the noun of Inna sentence was missing.

أن يجرى الرش بعد تطاير الندى .

The parser produces the following parse tree for the above sentence.

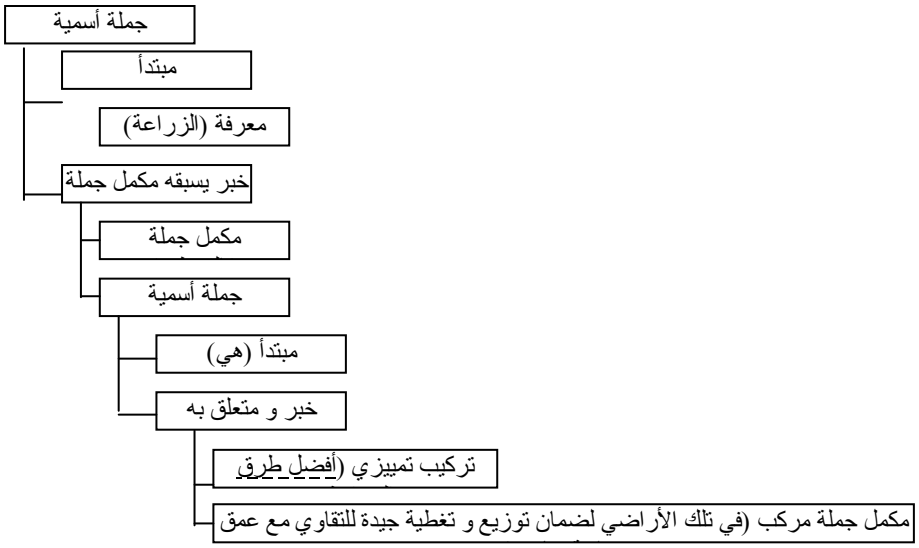


جملة إن و أخواتها(حرف إن(أن),خبر(جملة فعلية(فعل(يجرى,متعدي),مفعول به(معرفة(الرش,مذكر)),مكمل جملة(تركيب إضافي(ظرف زمان(بعد),تركيب إضافي(نكرة(تطاير,مذكر),معرفة(الندى,مذكر)))))))

Example (5): The following sentence shows how the parser is capable to deal with the case when the enunciative (الخبر) of the sentence was connected by a distant relationship.

الزراعة بالتسطير هي أفضل طرق الزراعة في تلك الأراضي لضمان توزيع و تغطية جيدة للتقاوي مع عمق زراعة مناسب .

The parser produces the following parse tree for the above sentence.

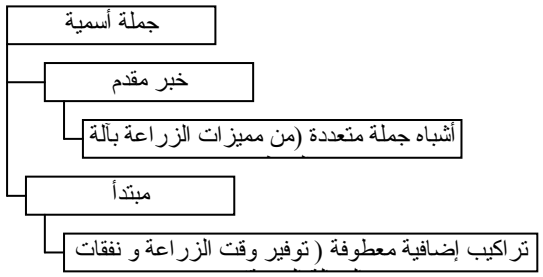


جملة أسمية(مبتدأ(معرفة(الزراعة,مؤنث)),خبر(خبر يسبقه مكمل جملة(مكمل جملة(شبه جملة(حرف جر(بال),معرفة (تسطير,مذكر))) ,جملة أسمية(مبتدأ(ضمير_منفصل(هي)),خبر(خبر و متعلق به(تركيب تمييزي(نكرة(أفضل,مذكر),تركيب إضافي(نكرة(طرق ,مؤنث) ,معرفة(الزراعة,مؤنث))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(في),تركيب بدل(اسم اشارة(تلك), معرفة(الأراضي ,مؤنث))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ل),تراكيب إضافية معطوفة(تركيب إضافي(نكرات متعددة(نكرة (ضمان,مذكر),نكرة(توزيع,مذكر))),حرف وصل(و),تركيب وصفي(نكرة(تغطية,مؤنث),نكرة(جيدة,مؤنث))))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(لل),معرفة(تقاوي,مؤنث))),مكمل جملة(شبه جملة(حرف جر(مع),تركيب وصفي(نكرات متعددة(نكرة(عمق,مذكر),نكرة(زراعة,مؤنث)),نكرة(مناسب,مذكر)))))))))))

Example (6): The following sentence show how the parser is capable to deal with the case when the enunciative (الخبر) of the sentence precedes it's inchoative (المبتدأ)

من مميزات الزراعة بآلة التسطير **توفير** وقت الزراعة و نفقات العمالة اليدوية .

The parser produces the following parse tree for the above sentence.



جملة أسمية(خبر مقدم(أشباه جملة متعددة(شبه جملة(حرف جر(من),تركيب إضافي(نكرة(مميزات,مؤنث),معرفة(الزراعة,مؤنث))),شبه جملة(حرف جر(ب),تركيب إضافي(نكرة(آلة,مؤنث),معرفة(التسطير,مذكر)))))),مبتدأ(تراكيب إضافية معطوفة(تركيب إضافي(نكرات متعددة(نكرة(توفير,مذكر),نكرة(وقت,مذكر)),معرفة(الزراعة,مؤنث)),حرف وصل(و),تركيب إضافي(نكرة(نفقات,مؤنث),تركيب وصفي(معرفة(العمالة,مؤنث),معرفة(اليدوية,مؤنث)))))))

5.2 Running the system

The purpose of our experiment was to investigate whether the parser is sufficiently robust for application to real-word Arabic text. We selected an unrestricted Arabic text, which is an agricultural extension document. The document is entitled " زراعة القمح في الأراضي الجديدة", نشرة زراعية من مركز البحوث الزراعية – نشرة رقم 420 1998, which carries many information and instructions to the Egyptian farmers. The text includes 10 pages, each of which contains about 20 sentences in average.

We have implemented the parser on a Pentium 133 processor, with 96 MG.Byte RAM, and 2.5 GB Hard Disk. The software used was SICStus Prolog ver. 3.7.

5.2.1 Results

In this section we discuss the testing results by first looking at whether the input sentence is parsable and then, if so, whether the complete parse is ambiguous.

The first question to ask is how good the output of the parser is. Table 5.1 shows the results of the parser. These results fall into two categories: the parsable sentence and the unparsable sentence.

The parsable sentence category refers to the case when the parser can parse input sentences leading to one or more parse trees. Sometimes the parser only assigns incorrect grammatical structure to the input sentence. For example, the input sentence

تميل إلي الصفرة ضعيفة النمو .

is assigned the incorrect parse tree

جملة فعلية(فعل(تميل,لازم),مكمل جملة(شبه جملة(حرف جر(إلي),معرفة(الصفرة,مؤنث))),فاعل(تركيب إضافي(نكرة(ضعيفة,مؤنث),معرفة(النمو,مذكر))))

Linguists analyze this sentence as if it were written as

تميل (**هي**) إلي الصفرة ضعيفة النمو .

We get this incorrect parsing because the parser fails to recognize the absence of the subject (الفاعل) of the verbal sentence. In this case it only recognizes the annexation constituent –(ضعيفة النمو)— as the subject (الفاعل) of the verbal sentence.

For this reason, we classified the parsable sentence category into two subcategories:

1. **Grammatical Successful**. Which has led to a complete successful parse of the input sentence.
2. **Grammatical Incorrect.** Which has led to a grammatically incorrect structure of the input sentence.

The unparsable sentence category refers to the case when the parser fails to parse the input sentences. Sometimes failures are due to ill-formedness of the input sentence which, is also not recognizable by linguists according to Arabic grammar rules. This

unexpected syntactic phenomenon in real-world texts is beyond the parser coverage. In other words, if the linguistic parsing fails, our parser fails. For example, the input sentence

يظهر بها نقط سوداء في حجم رأس الدبوس .

is not parsed by our parser because the subject (الفاعل) of the sentence—(نقط سوداء)—takes the feature gender as female, but the prefix (ي) of the verb (يظهـــر) of the sentence indicates that this feature value is expected to be male. The grammatically correct sentence would be as follows

تظهر بها نقط سوداء في حجم رأس الدبوس .

For this reason, we classified the unparsable sentence category into two subcategories:

1. **Ill-formedness.** Which has failed to parse due to an incorrect input sentence.
2. **Failure.** Which has failed to assign a parse tree to a grammatically correct sentence.

Table 5-1 Testing results

		Number of sentences	%
Parsable	**Grammatical Success**	**161**	**77**
	Grammatical incorrect	**27**	**12.9**
Unparsable	**Ill-formed**	**2**	**0.9**
	Failure	**19**	**9**
Total		**209**	

The total number of sentences across the document was 209 sentences. The average sentence length was 10 words. The longest sentence was 44 words long. The result shows that the number of sentences parsed successfully was 161 sentences, about 77%. 27 sentences were assigned incorrect parse trees, about 12.9%. The number of sentences that were not parsed (has not produce a parse tree) was 21 sentences, about 9.9%.

There are some notes about the sentences that produced grammatically incorrect parse trees. First, we do not consider the correctness of the parse fragments of the parse tree, which represent correct syntactic structure of parts of a sentence. In other words, the complete sentence is rejected if any of its parse fragments would be grammatically incorrect. Second, it is possible to decrease the total number of unparsable sentences by acquiring additional grammar rules.

Although the results observed in our experiment are satisfactory, the practical question is how many final parse trees we can get. This structural ambiguity may lead to an enormous amount of possible readings for an input sentence. Techniques will have to be developed to be able to deal with large numbers of analyses, and to be able to choose the most appropriate reading from such a set of candidate analyses. However, this is not covered in the current work.

Table 5-2 shows the ambiguities of sentences parsed successfully across the document. As shown, the maximum number of parse trees was limited to four parse trees. The total disambiguated sentences were 151. The total ambiguous sentences were 37.

Table 5-2 the relation between the number of trees and the number of sentences.

		Number of sentences
Number of Trees	**1**	151
	2	27
	3	9
	4	1

5.3 Analysis of Results

In this section we analyze the output of the parser from different point of views. This includes showing the relationship between the total number of sentences and the corresponding grammatical pattern, the relationship of the parsed sentence with its length, timing of the parser output, and the effect of the connectors on the parser output.

5.3.1 Analysis of Grammatical Patterns

Each grammar rule (form) is acquired from certain grammatical patterns. These patterns are listed in Appendix C. Table 5-3 shows the grammatical pattern along with the number of sentences that are parsed from this pattern. Although the total number of input sentence is 209, the total number of patterns is 236. This is due to the inevitable ambiguous output.

Table 5-3 the relation between the grammatical patterns and the number of sentences.

	Grammatic-ally Correct	%	Grammatic-ally Incorrect	%	Total
Verbal Sentence					
Form 1	1	0.6	5	6.9	6
Form 2	6	4.0	2	2.7	8
Form 3	1	0.6	6	8.3	7
Form 4	1	0.6	1	1.3	2
Form 5	0	0	1	1.3	1
Form 6	20	12.0	1	1.3	21
Form 7	2	1.2	1	1.3	3
Form 8	27	16.0	15	20.8	42
Form 9	4	2.4	0	0	4
Form 10	1	0.6	4	5.5	5
Form 11	30	18.0	1	1.3	31
Form 12	1	0.6	5	6.9	6
Form 13	9	5.4	1	1.3	10
Sub-Total	103	62.8	43	59.7	
Nominal Sentence					
Form 1	21	12.8	10	13.8	31
Form 2	1	0.6	0	0	1
Form 3	3	1.8	0	0	3
Form Inna1	1	0.6	1	1.3	2
Form Inna2	1	0.6	0	0	1
Form Inna3	2	1.2	2	2.7	4
Form Kanna1	0	0	0	0	0
Form Kanna2	1	0.6	0	0	1
Form Kanna3	4	2.4	0	0	4
Sub-Total	34	20.7	13	18.05	
Other Sentence					
Form of Prefixed Sentence	26	15.8	13	18.05	39
Form Verbal Compound Sentence	0	0	1	1.3	1
Form Nominal Compound Sentence	1	0.6	2	2.7	3
Sub-Total	27	16.4	16	22.2	
Total	164	100	72	100	

From Table 5-3 we note the following:
(1) First for grammatically correct patterns:

- Patterns that generate incomplete verbal sentences are the most commonly patterns in the input text. From the analyzed patterns, four patterns represent %51.5. These are *Form11*, *Form8*, *Form6*, and *Form13*. *Form11* pattern corresponds to the case when the subject (الفاعل) and the direct object (المفعول به) were missed. *Form8* pattern corresponds to the case when the subject was missed. *Form6* pattern corresponds to the case when the subject and the direct object were missed. *Form13* pattern corresponds to the case when the subject and the complementary were missed.
- Patterns that generate complete verbal sentences take a minimal percentage in our analyzed text. For example, *Form1* and *Form3* patterns, which contain all components of the sentence (Verb فعل, Subject فاعل, Direct Object مفعول به, Complementary مكمل جملة), represent only 1.2%.
- For the nominal sentence, we can see that, the default nominal pattern *Form1* (inchoative (المبتدأ) + enunciative (الخبر)) has the maximized percentage 12.8%. Where the rare patterns like *Form2* and *Form3*, which represent the enunciative (الخبر) precedes the inchoative (المبتدأ) has only 2.4%.
- Finally, we can observe that, there are some patterns have not correspondent sentences like *Form5* of verbal sentence, *Form Kanna1*, and Form *Verbal Compound Sentence*. We can see that, *Form5* has the form of (verb + subject + direct object), and in usage this form is rarely happened without a complementary. Although this rarity, we put this form in our grammar, because it represents an axiom rule. The same concept was applied over other two patterns.

(2) Second for grammatically incorrect patterns:

For the grammatically incorrect patterns, there is no indicator can be commented from the result. But we can use this entry to know which pattern is often recognized as an incorrect (Form8 of verbal sentence, Form1 of Nominal sentence, the Form of Prefixed Sentence), to focus in studying of them in future work.

5.3.2 Analysis of Sentence Length

In order to show the capability of the parser to analyze long sentence, we set a number of categories, each of which is a multiple of 5 words. Table 5-4 shows the total number of parsed sentences from each category.

Table 5-4 the relation between the sentence length and the parsing results

	0:5	5:10	10:15	15:20	20:25	25:30	30:35	35:40	40:45
Grammatically correct	33	68	32	19	4	4	1	0	0
Grammatically incorrect	2	10	7	4	2	1	0	0	1
UN-Parseable	2	7	6	2	2	0	2	0	0
Total	37	85	45	25	8	5	3	0	1

We can observe that, the sentences which have the length between 5 to 10 words were the most sentences grammatically succeed, 42% (68 sentences from 161 total grammatically correct sentences). In the second stage, the sentences which have the length 0 to 5 words, have the percentage 20.49%, and the sentences with the length 10: 15 have the percentage 19.8%.

5.3.3 Timing the parser output

In this section we time the parser output according to the sentence length, and number of connectors in the sentence.

5.3.3.1 Time versus sentence length

We set a number of categories of the sentence length, each of which is a multiple of 5 words. To make the sampling more homogenous, we set the last category of the sentence length to be the length of sentences which have more than 15 words (this is because, the total number of all sentences more than 15 words was only 28 sentences). Table 5-5 shows the relationship between the processing time and the sentence length.

Table 5-5 the relation between processing time and the sentence length

Sentence Length	0:5	5:10	10 :15	> 15
Average Parsing time for Grammatically correct	0:21	0:40	0:54	1:39

We can simply observe that, the curve of the time proportionally increases with the length of the sentences. This result comes from that, when the sentence has more words, the parser also has many combinations needed to traverse to parse the sentence.

5.3.3.2 Time versus number of connectors

We set a number of categories according to the number of connectors in the sentence, up to 5 connectors. To make the sampling more homogenous, we set the last category of the connectors to be the connectors of sentences which have more than 2 connectors (this because the total number of all grammatically correct sentences, more than 2 connectors, was only 4 sentences).

Table 5-6 shows the relationship between the processing time and the number of connectors.

Table 5-6 the relation between processing time and the number of connectors

Number of connectors in the sentence	0	1	2	3:5
Average Parsing time for Grammatically correct	0:43	0:54	1:12	1:31

Simply we can see that, the connectors in the sentence increase the time of parsing. This is due to the occurrence of the connector which gives a chance to build either two sentences connected to each other, or a single sentence has a connected constituent.

We must observe that, those chances will increase with the increasing of the connectors in the sentence.

5.3.4 Analysis of Connectors

In this section we analyze the effect of connectors versus parsing success. We set a number of categories according to the number of connectors in the sentence, up to 6 connectors. We also set three categories that correspond to the success of the parser in producing a correct parse tree: grammatical, ungrammatical, and un-parasble. Table 5-7 shows the relationship between the connectors and the parsing success.

Table 5-7 the relation between the connectors and the parsing success

	0	1	2	3	4	5	6
Grammatical correct	104	38	15	2	1	1	0
Grammatical incorrect	17	6	1	2	0	0	1
UN-Parsable	10	6	4	1	0	0	0
Total	131	50	20	5	1	1	1

We can observe that, when there is no connector in the sentence, we obtain the most ratio of grammatically correct sentences 64.5% (104 from 161).

5.4 Analysis of Ungrammatical sentence

Recall that the number of ungrammatical sentences was 27 sentence. The parser assigns incorrect parse tree(s) to the input sentence. In other words, a sub-tree of the parse tree was assigned an incorrect syntactic category that occurs at a higher level of the tree. These are classified into four categories:

1. Incorrect Direct object مفعـــول بـــه: The parser has recognized a fragment of the verbal sentence correctly but erroneously assigns it as a direct object of the sentence.

   The sentences that have incorrect direct object can be classified into the following categories:

(a) Some words can't be direct object anyway. This type of problem can be solved be adding an attribute in the lexicon to indicate if the word can be used as a direct object or not. Table 5-8 lists those sentences.

Table 5-8

Sentence #	Incorrect direct object	Notes
1-18	جيدا	Tree #1
1-18	خاصة	Tree #2
2-3	خاصة	
4-7	قبل الري	Tree #1
4-7	بداية	Tree #2
5-5	عند انتشار الحشائش	Tree #2
5-10	ضد الحشائش	
5-11	عند عمر 3–4 ورقات	
5-12	عند عمر 3–4 ورقات	

(b) Some sentences recognize Inna sentence (جملـة أن) as a direct object of the sentence, and this problem needs more study of a huge sample of those sentences, which embed in it another Inna sentences. After this investigation the grammar needs to be provided by additional condition(s) to decide when to accept Inna sentence as a direct object and when to refuse it. Table 5-9 lists those sentences.

Table 5-9

Sentence #	Incorrect direct object
8-14	فإن قسم بحوث القمح
9-5	أن تظهر

(c) Sentence 7-15 recognizes the word (الأمـــر) as the direct object of the sentence.

This is because the parser makes a false segmentation between two words (ســـريعة) and (التـــأثير) as a separated units, but the fact is, the two words are a single constituent which can't be separated.

(d) In sentence 8-2, the parser recognizes the constituent (تساقط الأمطـــار) as the direct object of the sentence. The parser takes this decision depending on the transitivity behavior of the verb (يـــؤدي) which is entered to the lexicon as transitive verb (فعـــل متعـــدي). But in fact this verb can occur sometimes as a transitive and in another time as an intransitive. This dual case of transitivity of the verb wasn't considered in our system.

(e) In sentence 8-16, the parser recognizes the constituent (عالية المحصـــول) as the direct object of the sentence. But this constituent is grammatically parsed just as an adjective constituent to the noun or the constituent which precede it, which in our sentence was (الأصـــناف الجديـــدة). This problem can be solved if the

parser considers of the role of the morphological patterns (الأوزان الصـــرفية) to determine the adjective constituent.

2. Incorrect Enunciative خـــبر: The parser recognized a fragment of the nominal sentence correctly but erroneously assigned it as an enunciative of the sentence.

In those sentences, the parser incorrectly recognized some constituents as enunciative.

The sentences have incorrect enunciative, can be classified into the following categories:

(a) The sentences, which come as a point under some title, are usually just phrases not complete sentences. This type of error comes in the sentences (2-14, 2-17, 6-11, 6-20, and 6-21).

(b) Sometimes there are some incomplete sentences, which occur in the text, but are not a point under a title as in the (a) category. This problem need more investigation to find the cases in which these phrases occur, and then we need to add those cases to the grammar to parse it as a phrase (incomplete sentence.).

This type of error comes in the sentences (4-25, 7-18, and 7-1).

(c) The parser recognizes some constituents as an enunciative wherever the true enunciative follows this constituent in the order. This due to the parser considers the first chance of the enunciative. We can solve this problem by considering the in-determine noun (النكـــرة) as the enunciative of the nominal sentence, whatever the order of appearance of it in the sentence is.

This type of error comes in the sentences (8-8 , and 9-2).

(d) Also there is another problem of the same reason of the parser considering for the first chance of the enunciative in the nominal sentence. But this time we need to take the priority to the Quasi-proposition (شـــبه الجملـــة) over the annexation constituent only if this annexation constituent included a place or time accusative (ظرف مكان أو زمان).

This type of error comes in the sentence 10-3.

3. Incorrect Subject فاعـــل: The parser recognizes a fragment of the verbal sentence correctly but erroneously assigns it as a subject of the sentence.

The sentences which have incorrect subject, can be classified into the following categories:

(a) Segmentation error, in this cases the grammar makes a false segmentation to some constituent. It occurs in the sentence 9-8 when the parser separates the constituent (غرب و وسط و جنوب الدلتا) from the constituent (في منـــاطق) by error, and then this error gives the parser a chance to recognize the constituent (غرب و وسط و جنوب الدلتا) as the subject of the sentence.

(b) The reminder sentences need semantic phase to take the decision to reject a specific constituent from being a subject of the sentence.

This case occurs in the sentence (4-15) when the parser considers the constituent (ضـــعيفة النمـــو) as the subject, and in the sentence (9-18) when it considers the constituent (نتيجة اهتزاز النباتات) also as a subject.

4. Connection عطـــف: The parser has erroneously recognizes the sentence as two connected sentences whereas it is a sentence that contains two connected constituents. There is only one sentence occurred from this type sentence (9-9).

5.5 Unparsable sentence

Recall that the number of unparsable sentences was 21 sentence. The parser fails to assign any parse tree to the input sentence. These are classified into four categories:

1. Agreement: The parser fails to produce a parse tree because the parse fragments that correspond to parts of the sentence do not agree with some features constraints.

(a) The first category in this agreement case was in the adjective constituent. It occurs in the following sentences (3-3) in constituent (الكميات الموصي بها), (6-4) in constituent (نباتات قمح قوية), (6-6) in constituent (طريقة الزراعـــة الحراثـــي), and (8-13) in constituent (أصناف القمح المتزرعة).

(b) The second category of agreement occurs in the sentence (9-28), when the parser reject the non-agreement between the enunciative (خبر) and the determinate (التعريف) of the noun (الجراثيم) in Kanna sentence.

(c) The last one was in the sentence (10-8), when the parser rejects the non-agreement between the gender of the subject (نقط سوداء) which is female and the gender indicator of the verb (تظهر) which indicates to the male subject.

2. Ill-formed: The parser fails to produce a parse tree because input sentence was grammatically incorrect. This case occurs in the two sentences (6-3, and 6-7).

3. Segmentation error: The parser fails to produce a parse tree because it failed to make a correct segmentation between two constituents. This case occurs in the two sentences (7-12) when the parser recognizes the constituent (المهم الاكتشاف المبكر) as a single constituent, but the true it is a two separated constituents (المهم) and then (الاكتشاف المبكر). Also sentence (8-5) recognizes the constituent (نقط ثابتة حول) as a single constituent.

4. Unrecognized grammatical pattern: The parser fails to produce a parse tree because the syntactic form of the sentence is not included in the grammar. Acquiring the grammar that corresponds to the pattern of the input sentence can solve this problem.

The sentences which have this case are (3-17, 5-7, 6-14, 6-18, 7-11, 7-14, 9-15, 9-17, 9-19, 10-12, 10-20).

CHAPTER (6)
CONCLUSION AND FUTURE WORK

6.1 Conclusion

This thesis has been concentrated on issues in the design and implementation of an Arabic Parser. The parser encodes the Arabic grammar rules of irab and the effects of applying these rules to the constituents of sentences. The parser is written in Definite Clause Grammar (DCG). The developing of the parser was a two-step process.

In the first step, we acquired the rules that constitute a grammar for Arabic that gives a precise account of what it is for a sentence to be grammatical. The grammar has been developed especially for the purposes of understanding scientific Arabic text. On the one hand this has the advantage that the grammar can be tailored to the specific requirements of the scientific domain. On the other hand, we want to adopt general solutions as much as possible, as this increases the chances that the grammar can be used in other domains as well. Thus, in designing the grammar we seek a balance between short-term goals (a grammar which covers sentences typical for the scientific domain and is reasonably robust and efficient) and long-term goals (a grammar which covers the major constructions of Arabic in a general way).

The grammar currently covers sentences that fall into one of two categories: simple sentence and compound sentence. A simple sentence is a sentence, which is not connected by any means with another sentence. However, it may embed another sentence. A compound sentence is more than a simple sentence connected with a conjunction article (أداة عطـــف). Simple sentences are classified into three classes: nominal sentence, verbal sentence, and special sentences. Special sentences are either the special verbs (Kana and its sisters كان و أخواتها), or special particles ('Inna and its sisters أن و أخواتها).

The second step was to implement the parser that assigns grammatical structure onto input sentence. A major design goal is that the system can be geared towards any other

related system or application since it has been built as a module. The parser has solved partially the syntactic ambiguities.

Experiment on real extension document was performed. The purpose of our experiment was to investigate whether the parser is sufficiently robust for application to real-word Arabic text. We selected an unrestricted Arabic text, which is an agricultural extension document. The document is entitled "زراعة القمح في الأراضي الجديدة", نشرة زراعية من مركز البحوث الزراعية – 1998 420 نشرة رقم , which carries many information and instructions to the Egyptian farmers. The text includes 10 pages, each of which contains about 20 sentences in average. The total number of sentences across the document was 209 sentences. The average sentence length was 10 words. The longest sentence was 44 words long. The results observed in our experiment are satisfactory. The result shows that the number of sentences parsed successfully was 161 sentences, about 77%. 27 sentences were assigned incorrect parse trees, about 12.9%. The number of sentences that were not parsed (has not produce a parse tree) was 21 sentences, about 10%.

6.2 Efficiency Issues

Here we will introduce proposal deterministic parser, this parser based on two-layers, the first one is reasonable to make segmentation of the sentence into small separated ones to parse each one in the layer 2, the second layer is simply the parser of the simple segmented sentence.

The advantages of this deterministic parser as follows:

1. The reductions of wasted time, which consume in solve some sentences more than one time. As an example if we has a sentence which contain 3 embedded sentences, and each one need the following time 2,3,2 minuets respectively, then the total time we spent to parse this large sentence will be (2*3*2 = 12 minuets). But if we separate each one and parse it alone, we will need only (2+3+2 = 7 minuets).
2. The ability to parallelism, if we can get parallel processor, we can send each subsentence to a processor in the same time.

We used some heuristics rules to make the segmentation operations on layer1 of the parser, the following are the rules which recognize the separations between embedded sentences:

1. Word (حتى) + verb.
2. Word (لما) + verb.
3. Word (حتى) + negation article (حرف نفي) + verb.
4. Word (إذا) + verb.
5. Connector articles (“أو” , “و”) + word (قد).
6. Connector articles (“أو” , “و”) + word (هناك).
7. Connector articles (“أو” , “و”) + verb.
8. Connector articles (“أو” , “و”) + negation article (حرف نفي) + verb.
9. Preposition “lam” (حرف الجر لام) + verb.
10. Connector articles (“أو” , “و”) + article “maa” (حرف ما).
11. Preposition “lam” (حرف الجر لام) + demonstrative pronoun (اسم اشارة) + non-noun word.
12. Preposition “kaf” (حرف الجر كاف) + demonstrative pronoun (اسم اشارة) + non-noun word.
13. Conjunctive noun (اسم صلة) + verb.
14. in this case we will end this segment at either a new separation or an occurrence of another verb.
15. Connector articles (“أو” , “و”) + Conjunctive noun (اسم صلة) + verb.
16. in this case we will end this segment at either a new separation or an occurrence of another verb.
17. Separation word (لذا - حيث - لو - بينما - مما).
18. Separation sign (comma “,”).

6.3 Future Work

We have several on going activities, all concerned with extending our thesis work to be more powerful and applicable. In what follows, we present some of these activities.

Improve the parsing efficiency, this can be done in several ways. One way is to segment the sentence into parts and build the parse fragments for each of which, then in a final

phase these partial parse fragments is formed into a complete parse tree. Another way is to use a programming language that supports a threading mechanism in order to parallelize the parsing of the input Arabic sentence.

Implement the semantic analysis phase, which concerns representing the meaning of an individual Arabic sentence. This task is difficult because there is no universally accepted notation has been devised for semantics. This phase is essential for implementing natural language understanding system for Arabic and solving semantics ambiguities.

Conduct a comparative study by reimplementing the acquired grammar using other parsing techniques such as chart parsing.

Investigate the possibilities of describing the Arabic grammar using new formalisms such as Lexical Function Grammars.

Integrate this system into some other Arabic applications such as machine translation systems, systems for teaching Arabic, and system for checking and correcting grammatical errors.

APPENDIX (A)
ANALYSIS OF RESULTS

Page #1

Has grammat i-cal tree	# of conn- ectors	# of words	#Of Trees	Time	Text	No.
Yes	4	24	2	3:30	تعتبر الأراضي الصحراوية **أو** الرملية **أو** الجيرية المستصلحة حديثا **و** المروية جزءا هاما في خطة التوسع الأفقي في الأراضي الزراعية **و** في الإنتاج الزراعي .	1-1
Yes	1	28	1	0:27	نظرا لأهمية محصول القمح باعتباره المحصول الغذائي الأول في مصر فإنه يعتمد على تلك الأراضي في إنتاج القمح لتقليل الفجوة بين المنتج المحلي **و** المستهلك من حبوب القمح .	1-2
Yes	2	19	2	3:50	القمح في هذه الأراضي يتعرض لظروف غير ملائمة تبعا لنوعية التربة **و** قلة خصوبتها **و** قلة احتفاظها بمياه الري .	1-3
Yes	1	20	1	0:50	فمن المهم اتباع التوصيات الفنية الخاصة بإنتاج القمح في تلك المناطق بكل دقة **و** بقدر الإمكان للتوصل إلي محصول جيد .	1-4
Yes	0	3	1	0:20	سخا8 من الأصناف .	1-5
Yes	0	9	2	0:42	تلقى قبولا كثيرا لدى زراع القمح في الأراضي الجديدة .	1-6
Yes	0	4	1	0:09	يجود في الأراضي الجديدة .	1-7
Yes	0	4	1	0:21	ترتفع بها نسبة الملوحة .	1-8
Yes	0	7	1	0:35	يزرع هذا الصنف في معظم أراضي الجمهورية .	1-9
Yes	0	4	1	0:15	جيزة167 من الأصناف الجديدة .	1-10
Yes	0	5	1	0:10	تجود زراعته في الأراضي الجديدة .	1-11
Yes	0	13	2	0:41	سدس1 من الأصناف الجديدة عالية المحصول المقاومة للأمراض تجود زراعتها في الأراضي الجديدة .	1-12
Yes	0	6	3	0:55	الصنف جميزة3 يشترط عدم زراعته مبكرا .	1-13
Yes	1	6	1	0:26	إنه مبكر التزهير **و** مبكر النضج .	1-14
Yes	1	12	1	0:25	أن هذا الصنف حساس للرطوبة **و** زيادة كمية مياه الري عند الزراعة .	1-15
Yes	0	14	1	1:13	ينصح بأن تكون رية الزراعة على الحامي في حالة الري السطحي في الأراضي الجيرية .	1-16
Yes	0	5	1	0:30	تحرث الأرض باستخدام المحراث الحفار .	1-17
No	0	14	2	0:36	تنعم جيدا باستخدام المحراث القرصي مع نقاوة الحشائش خاصة في حالة استعمال آلات التسطير .	1-18

Page #2

Has grammati -cal tree	# of conne ct-ors	# of words	#Of Trees	Time	Text	No.
Yes	0	7	1	0:30	تقسم إلي أحواض بمساحة حوالي 3*4 متر .	2-1

2-2	تحرث الأرض باستخدام المحراث القرصي مع نقاوة الحشائش .	1:05	1	8	0	Yes
2-3	تسوى بقدر الإمكان خاصة عند استعمال آلات التسطير .	0:50	1	8	0	No
2-4	تعتبر الزراعة العفير بدار هي أكثر طرق الزراعة استخداما **و** فيها تبذر التقاوي عقب حرث **و** خدمة الأرض .	2:02	4	18	2	Yes
2-5	تغطى جيدا بأي وسيلة ممكنة .	0:20	2	5	0	Yes
2-6	تقسم الأرض إلي أحواض صغيرة نسبيا لإحكام الري في حالة الأراضي المروية بالغمر **أو** ريا سطحيا .	1:55	2	16	1	Yes
2-7	تترك بدون تقسيم في حالة الري بالرش المحوري **أو** المتنقل **أو** الثابت .	0:55	2	12	2	Yes
2-8	الزراعة بالتسطير هي أفضل طرق الزراعة في تلك الأراضي لضمان توزيع **و** تغطية جيدة للتقاوي مع عمق زراعة مناسب .	2:15	1	19	1	Yes
2-9	فيها تعاير السطارة للزراعة بمعدل 60 كيلو جرام للفدان في سطور على مسافات 12,5-13 سم **و** عمق 4-5 سم .	0:25	1	19	1	Yes
2-10	في حالة الزراعة العفير بدار يستخدم 75 كيلو جرام تقاوي للفدان على أن تبذر بانتظام على الأرض مع التغطية الجيدة .	10:10	1	20	0	Yes
2-11	باستعمال آلة التسطير يستخدم 60 كيلو جرام تقاوي للفدان .	0:18	1	9	0	Yes
2-12	في حالة الزراعة العفير يستخدم 85 كيلو جرام تقاوي للفدان .	0:20	1	10	0	Yes
2-13	توفير كمية التقاوي المستخدمة في الزراعة .	0:25	1	6	0	Yes
2-14	انتظام توزيع التقاوي في الحقل **و** انتظام عمق الزراعة **و** ضمان تغطية الحبوب عقب الزراعة .	2:00	1	15	2	No

2-15	يؤدي إلي زيادة سرعة**و** نسبة الإنبات**و** انتظام نمو النباتات**و** جودة التفريع**و** تقليل منافسة النباتات لبعضها**و** بالتالي زيادة المحصول من الحبوب بحوالي 20% عن الزراعة اليدوية .	1:20	2	30	5	Yes
2-16	من مميزات الزراعة بآلة التسطير توفير وقت الزراعة**و** نفقات العمالة اليدوية .	0:25	1	12	1	Yes
2-17	إمكانية استعمال الماكينات المجهزة للتسميد بالجرعة التنشيطية مع الزراعة .	0:36	1	9	0	No

Page #3

No.	Text	Time	#Of Trees	# of words	# of connect-ors	Has grammati-cal tree
3-1	أفضل ميعاد للزراعة هو الفترة من 15 إلي 30 نوفمبر .	0:52	1	10	0	Yes
3-2	يراعى ضرورة الالتزام بميعاد الزراعة الموصي به لكل صنف لتفادي الإصابة بالمن**و** تقليل الفاقد من مهاجمة العصافير .	1:07	1	18	1	Yes
3-3	نظرا لطبيعة الأراضي الجديدة**و** قلة خصوبتها**و** عدم احتفاظها بالعناصر الغذائية فإنها تحتاج إلي معدلات مرتفعة نسبيا من الأسمدة مع ضرورة مراعاة الكميات الموصي بها**و** إضافتها في المواعيد المقررة .	0:48	0	31	3	
3-4	يعمل السماد البلدي على تحسين خواص التربة الطبيعية .	0:18	1	8	0	Yes
3-5	يزيد من قدرة الأراضي الجديدة على الحفاظ على مياه الري**و** العناصر الغذائية .	0:55	1	13	1	Yes
3-6	يشترط في السماد البلدي أن يكون من مصدر موثوق به .	1:15	1	10	1	Yes
3-7	يكون خاليا من مسببات الأمراض**و** يرقات الحشرات **و** النيماتود .	1:13	1	10	2	Yes
3-8	يضاف السماد البلدي بمعدل 20 متر مكعب للفدان .	0:30	1	8	0	Yes
3-9	تنثر على الحقل .	0:11	1	3	0	Yes

3-10	تخلط جيدا بالتربة مع الخدمة .	0:20	1	5	0	Yes
3-11	لا ينصح أبدا بنقل الأتربة **في** نواتج التطهير إلي الحقل .	0:40	1	10	1	Yes
3-12	تسببه من انتشار الحشائش **في** الأمراض **في** زيادة ملوحة التربة .	0:35	1	10	2	Yes
3-13	يستخدم سماد سوبر فوسفات الجير الأحادي .	0:25	2	6	0	Yes
3-14	يضاف بمعدل 150 كيلو جرام في الأراضي الجيرية **في** بمعدل 200 كيلو جرام في الأراضي الرملية .	0:40	1	16	1	Yes
3-15	يضاف 50 كيلو جرام من سماد سلفات بوتاسوم للفدان قبل الزراعة **في** أثناء الخدمة مع التقليب جيدا في التربة .	1:05	1	19	1	Yes
3-16	يمكن استعمال المخصبات الحيوية .	0:20	2	4	0	Yes
3-17	تحتوي على بكتيريا تحلل الفوسفور المثبت في التربة لجعله في صورة ميسرة **في** صالحة لامتصاص النبات .	1:35	0	16	1	
3-18	يضاف مخصب الفوسفور بواقع 2-3 أكياس للفدان عند الزراعة .	0:45	1	9	0	Yes
3-19	يضاف السماد النتروجيني بمعدل 100 إلي 120 كيلو جرام نتروجين للفدان .	0:30	1	11	0	Yes
3-20	هذا المعدل يعادل 300-360 كيلو جرام سماد نترات نشادر 33,5% آزوت .	0:16	1	11	0	Yes

Page # 4

No.	Text	Time	#Of Trees	# of words	# of connect-ors	Has grammati-cal tree
4-1	هما نوعا السماد الموصي بهما في الأراضي الجديدة .	0:40	1	8	0	Yes

4-2	لا يوصى باستعمال سماد اليوريا في الأراضي الجديدة .	0:35	1	8	0	Yes
4-3	تتم إضافة السماد الآزوتي في الأراضي الجيرية .	0:20	1	7	0	Yes
4-4	تروى بنظام الغمر على ثلاث دفعات عند الزراعةو عند رية المحاياةو قبل مرحلة طرد السنابل .	1:02	1	17	2	Yes
4-5	في الأراضي الرملية يقسم السماد النتروجيني على 5-6 دفعات متساوية .	0:20	1	10	0	Yes
4-6	تروى بنظام الري بالرش .	0:35	1	4	0	Yes
4-7	تضاف قبل الري بداية من الزراعة .	0:40	2	6	0	No
4-8	يفضل إضافة السماد مع مياه الري باستخدام السمادة للحصول على نتائج أفضل .	0:39	1	12	0	Yes
4-9	يمكن للمزارع أن يتعرف على مدى احتياج نباتات القمح إلي السماد النتروجيني .	1:05	1	12	0	Yes
4-10	ذلك عن طريق ملاحظة لون النباتات .	0:48	1	6	0	Yes
4-11	من المفروض أن تكون النباتات خضراء اللون .	0:35	1	7	0	Yes
4-12	يميل اللون قليلا إلي الزرقة .	0:20	1	5	0	Yes
4-13	أن يكون متجانسا في الحقل .	1:10	1	5	0	Yes
4-14	النباتات ذات اللون الأخضر الشاحب فتدل على احتياجها للسماد النتروجيني .	1:25	2	10	0	Yes

No	0	5	1	0:25	تميل إلي الصفرة ضعيفة النمو .	4-15
Yes	1	20	1	0:25	نظرا لقلة احتفاظ الأراضي الجديدة عامة **في** الأراضي الرملية خاصة بمياه الري فإن تلك الأراضي تحتاج إلي فترات ري متقاربة .	4-16
Yes	0	8	1	0:55	يمكن توفير الرطوبة الأرضية اللازمة لنمو نباتات القمح .	4-17
Yes	0	18	1	0:55	في الأراضي الرملية يراعى إعطاء رية بعد حوالي يوم واحد من رية الزراعة لضمان توفير الرطوبة الكافية للإنبات .	4-18
Yes	1	18	2	1:42	تتكرر الريات اليومية حتى ظهور البادرات مع ضرورة الاهتمام بعدم تعطيش النباتات طول فترات النمو **في** حتى الحصاد .	4-19
Yes	0	10	1	0:31	يروى القمح في الأراضي الرملية على فترات من 5-10 أيام .	4-20
Yes	0	19	1	1:15	في الأراضي الجيرية يحتاج القمح إلي حوالي 7 ريات خلال الموسم مع عدم تأخير رية المحاياة عن 20-225 يوما .	4-21
Yes	0	3	1	0:30	تروى بنظام الغمر .	4-22
Yes	0	7	3	0:48	يوالي الري بعد ذلك كل 15-20 يوما .	4-23
Yes	1	27	1	2:23	يمكن للمزارع أن يدرك مدى احتياج الحقل إلي الري عن طريق جفاف التربة **في** ظهور الشقوق على سطح التربة في الأراضي الجيرية **في** جفاف الطبقة السطحية بها .	4-24
No	1	10	1	1:23	التفاف أوراق النباتات **في** ضعف نموها كمظهر من مظاهر العطش .	4-25

Page # 5

Has grammati-cal tree	# of connect-ors	# of words	#Of Trees	Time	Text	No.
Yes	0	9	1	0:40	من الضروري العمل على مقاومة الحشائش في حقول القمح .	5-1

5-2	لا تنافس المحصول في احتياجاته البيئية .	0:26	1	6	0	Yes
5-3	تتم المقاومة سواء يدوية أو مكانيكية أو كيماوية .	0:55	1	8	2	Yes
5-4	لزم الأمر .	0:10	2	2	0	Yes
5-5	يمكن استعمال المكافحة الكيماوية للحشائش عند انتشار الحشائش بشدة مع ضبط معدل الرش لضمان الانتظام .	1:53	2	15	0	No
5-6	أن يجرى الرش بعد تطاير الندى .	0:27	1	6	0	Yes
5-7	يفضل أن يتم رش الحشائش كيماويا بعد ري الحقل بأيام قليلة .	1:25	0	11	0	
5-8	يكون تأثير الرش فعال .	0:19	1	4	0	Yes
5-9	سينال 10% SC بمعدل 40 سم3/ف رشا قبل رية المحاياة بيوم واحد مع 200 لتر ماء للفدان .	0:50	1	17	0	Yes
5-10	سنكور 70% WP بمعدل 60 جم/فدان رشا في طور 2-4 أوراق للقمح و هذان المبيدان يستخدمان ضد الحشائش عريضة الأوراق فقط .	2:01	1	21	1	No
5-11	أريلون 50% FL بمعدل 1,25 لتر/فدان مع 200 لتر ماء يرش عند عمر 3-4 ورقات لنبات القمح مع رج العبوة جيدا قبل الاستعمال .	2:47	3	23	0	No
5-12	IPFLOW 50% FL بمعدل 1,25 لتر في 200 لتر ماء للفدان يرش عند عمر 3-4 ورقات لنبات القمح .	1:55	3	18	0	No
5-13	لا يستخدم أي من هذين المبيدين في الأراضي الرملية أو الكلسية .	0:30	1	11	1	Yes
5-14	يتم استخدامها في الأراضي الطينية فقط .	0:34	1	6	0	Yes

5-15	يستخدم مبيد سافيكس 20% EC بمعدل 1,25 لتر للفدان أو مبيد جراسب 10% EC بمعدل واحد لتر للفدان في حوالي 200 لتر ماء .	1:03	1	23	1	Yes
5-16	يرش عند عمر 4-5 ورقات لنبات القمح .	0:45	2	7	0	Yes
5-17	توبيك 24% EC بمعدل 100 سم3 للفدان رشا خلال شهر بعد المحاياة .	1:30	1	12	0	Yes
5-18	اسبرت 25% EC بمعدل 850 سم3 للفدان رشا بعد 30-35 يوما من الزراعة أو بوناسوبر 75% EW بمعدل 500 سم3 للفدان رشا بعد 30-35 يوما من الزراعة .	1:51	1	27	1	Yes

Page # 6

No.	Text	Time	#Of Trees	# of words	# of connect-ors	Has grammati-cal tree
6-1	اتباع دورة زراعية يتخللها محصول البرسيم .	0:35	1	6	0	Yes
6-2	يساهم في تقليل أعداد نباتات الزمير بتكرار حش البرسيم .	1:05	1	9	0	Yes
6-3	إجراء الخدمة الجيدة و زراعة القمح بالسطارة و اتباع كافة التوصيات الفنية .	1:55	0	12	2	
6-4	يساعد على زيادة نسبة إنبات القمح و إعطاء نباتات قمح قوية النمو .	1:20	0	12	1	
6-5	يزيد القدرة على منافسة الحشائش و التفوق إليها في النمو .	0:20	1	10	1	Yes
6-6	اتباع طريقة الزراعة الحراثي في الأرض الموبوءة بحشيشة الزمير .	0:55	0	9	0	
6-7	خاصة في الأراضي الجيرية تحت نظام الري بالغمر .	0:13	0	8	0	
6-8	تعتبر حشرات من القمح من أهم مجاميع الحشرات الثاقبة الماصة .	1:03	1	10	0	Yes

6-9	تصيب المحاصيل النجيلية عموما .	0:28	2	4	0	Yes
6-10	تتأثر النباتات المصابة نتيجة .	0:22	1	4	0	Yes
6-11	السحب المباشر للعصير الخلوي نتيجة التغذية بأعداد كبيرة من الحشرات على الأنسجة النباتية .	1:44	1	13	0	No
6-12	تعجيز الأوراق المصابة نتيجة لتجمع الإفرازات العسلية و نمو الأعفان و قفل الثغور التنفسية للأوراق .	1:35	1	15	2	Yes
6-13	تاثر الأنسجة النباتية نتيجة للإفرازات اللعابية لمن القمح في صورة بقع حمراء ذات مراكز سوداء .	1:43	1	15	0	Yes
6-14	لا تلبث أن تنتشر .	0:06	0	4	0	
6-15	يعقبها تدمير للمادة الخضراء بالأوراق .	0:10	1	5	0	Yes
6-16	تموت الأنسجة المصابة .	0:09	1	3	0	Yes
6-17	تلاحظ هذه الظاهرة على الأوراق السفلية بالقرب من سطح الأرض و المنطقة الوسطى من النبات .	0:38	1	15	1	Yes
6-18	هناك بعض العوامل نوجز منها ما يلي .	0:55	0	7	0	
6-19	أثر بالسلب على وفرة الأعداء الحيوية .	0:40	1	6	0	Yes
6-20	الاكتشاف المتأخر للإصابة .	0:21	1	3	0	No
6-21	تداخل مواعيد الزراعة و صغر حجم الحيزات و عدم الاعتناء بإزالة بقايا المحصول و تنقية الحشائش .	1:05	1	16	3	No

6-22	التسميد النتروجيني الغزير و في صورة مجزئة يأتي في مصلحة انتشار و تزايد حشرات المن .	0:30	1	15	2	Yes

Page #7

No.	Text	Time	#Of Trees	# of words	# of connect-ors	Has grammati-cal tree
7-1	زراعة الأصناف ذات الاحتياجات السمادية العالية .	0:49	1	6	0	No
7-2	من الجدير بالذكر أن هناك بعض التوصيات الهامة يجب اتباعها .	0:23	1	10	0	Yes
7-3	تكبير المساحات المنزرعة بالقمح و تجميعها و توحيد ميعاد الزراعة بقدر الإمكان .	2:17	1	12	2	Yes
7-4	تعمل المساحات الصغيرة و المتبادلة مع محاصيل أخرى على عكس طول موجي لوني معين .	0:18	1	14	1	Yes
7-5	تستجيب إليه حشرات المن أثناء الطيران .	0:32	1	6	0	Yes
7-6	فتهبط على القمح .	0:12	1	3	0	Yes
7-7	من ناحية أخرى يعمل تداخل مواعيد الزراعة على تحرك حشرات المن بحثا عن الأوراق الغضة و العمر الأصغر .	0:24	1	18	1	Yes
7-8	نتيجة لتغير الظروف البيئية و الميل التدريجي لارتفاع درجات الحرارة خلال السنوات السابقة لوحظ تواجد حشرات المن على البادرات عقب الإنبات بحوالي 10 أيام .	0:48	2	24	1	Yes
7-9	تظل أعداد الحشرات منخفضة حسب الضغوط البيئية .	1:30	3	7	0	Yes
7-10	تتعرض إليها .	0:12	1	2	0	Yes
7-11	سمحت درجات الحرارة فإن الأعداد تزداد بصورة وبائية .	0:15	0	8	0	

7-12	فمن المهم الاكتشاف المبكر للإصابة بمعنى مرور السادة مشرفي الأحواض على المساحات المسئولة منهم و اكتشاف بؤر الإصابة .	0:20	0	18	1	
7-13	تبدأ عند الحواف و التعامل معها بسرعة قبل انتشارها .	0:45	2	9	1	Yes
7-14	تعظيم أو المحاولة لإظهار فاعلية الأعداء الحيوية بعدم اللجوء السريع لاستخدام المبيدات التقليدية خاصة لأن الإصابات الفيروسية الأساسية بالقمح حتى الآن ذات نسبة قليلة .	3:50	0	24	0	
7-15	لا يستدعي الأمر استخدام مبيدات سريعة التأثير .	0:58	1	7	0	No
7-16	عند استخدام المكافحة الكيماوية ينصح بأحد المواد الآتية .	0:40	1	8	0	Yes
7-17	سوبر مصرونا 94% مستحلب بمعدل 1 لتر لكل 100 لتر ماء .	0:40	1	11	0	Yes
7-18	هذا مع مراعاة استخدام رشاشة ظهرية ذات بشبوري منحني لأسفل .	2:07	1	10	0	No
7-19	يصل محلول الرش للسطوح السفلي للأوراق .	0:23	1	6	0	Yes

Page #8

No.	Text	Time	#Of Trees	# of words	# of connect-ors	Has grammati-cal tree
8-1	يراعى استخدام المواد البديلة للمبيدات .	0:33	1	5	0	Yes
8-2	يؤدي تساقط الأمطار إلي الإقلال من فاعليتها .	0:43	1	7	0	No
8-3	يعتبر العلاج بعد طرد السنابل عديم الجدوى .	1:42	3	7	0	Yes
8-4	أفضل طريقة لمكافحة الفئران هي استخدام المقاومة الجماعية بأحد الطعوم السامة مثل فوسفيد الزنك مع جريش الذرة أو المبيدات المسيلة للدم بالنسب المقررة .	2:35	1	23	1	Yes

8-5	توضع في نقط ثابتة حول و داخل الحقل و على مسافات .	0:15	0	11	2	
8-6	تتناسب مع الكثافة العددية للفئران مع متابعتها المستمرة .	0:25	1	8	0	Yes
8-7	بالنسبة للعصافير فإنه يفضل أن تتم زراعة القمح في تجمعات كبيرة في المواعيد الموصي بها .	1:07	1	15	0	Yes
8-8	أن تكون مواعيد الزراعة في الحقول المجاورة متقاربة لتقليل أضرار مهاجمة العصافير مع إزالة الأعشاش .	2:05	1	15	0	No
8-9	كانت متواجدة على الأشجار المحيطة للحقل .	0:25	1	6	0	Yes
8-10	ينصح في الأراضي الموبوءة بالقواقع زراعة داير من البرسيم حول حقل القمح .	0:43	1	12	0	Yes
8-11	يكون بمثابة مصيدة للقواقع .	0:40	1	4	0	Yes
8-12	من أهمها أصداء القمح الثلاثة الأسود و البرتقالي و الأصفر و مرض التفحم السائب .	0:40	1	14	3	Yes
8-13	فإن أغلب أصناف القمح المنزرعة و المستنبطة بمعرفة قسم بحوث القمح مقاومة للأصداء الثلاثة .	2:10	0	14	1	
8-14	في حالة ظهور طرز جديدة من الأمراض الفطرية تصيب بعض الأصناف المنزرعة فإن قسم بحوث القمح يعمل على إحلالها بأصناف جديدة عالية المحصول و مقاومة للمرض .	2:05	2	26	1	No
8-15	بالنسبة لمرض التفحم السائب فإن التقاوي الموصي بها من قبل قسم بحوث القمح و الموزعة عن طريق وزارة الزراعة يتم معاملتها ضد المرض بالمطهرات الفطرية الجهازية لمنع ظهور و انتشار المرض .	0:40	1	31	2	Yes
8-16	فإنه يوصى دائما باستعمال التقاوي المعتمدة للأصناف الجديدة عالية المحصول و الموزعة عن طريق وزارة الزراعة و عدم اللجوء إلي زراعة الأصناف القديمة و غير الموصي بها لشدة إصابتها و انخفاض محصولها و الزراعة في المواعيد المقررة و اتباع كافة التوصيات الفنية لإنتاج القمح .	2:10	1	44	6	No
8-17	تظهر العلامات على شكل بقع صفراء منفصلة إليها مظهر مسحوقي مرتبة في صفوف طولية مع محور الورقة و متوازية .	0:48	1	19	1	Yes

8-18	تظهر الإصابة على الأوراق و الأغماد و القنابع .	0:14	1	8	2	Yes
8-19	في نهاية الموسم أو عند اشتداد الحرارة يتحول اللون الأصفر إلي اللون المسود اللامع .	0:16	1	14	1	Yes

Page # 9

No.	Text	Time	#Of Trees	# of words	# of connect-ors	Has grammati-cal tree
9-1	يناسب المرض درجات الحرارة المنخفضة نهارا .	1:12	3	6	0	Yes
9-2	أن يكون الفرق بين درجات الحرارة بين الليل و النهار كبيرا .	2:13	1	11	1	No
9-3	ينتشر في مناطق شمال و وسط و جنوب الدلتا عن المناطق الأخرى في وسط و جنوب الوادي .	0:35	1	17	3	Yes
9-4	تظهر العلامات على هيئة بقع مسحوقية لونها بني فاتح مستديرة مبعثرة بدون نظام .	0:35	1	13	0	Yes
9-5	يمكن أن تظهر على سطحي الورقة .	0:52	1	6	0	No
9-6	لا تظهر الإصابة إلا على الأوراق فقط .	0:18	1	7	0	Yes
9-7	يناسب المرض درجات الحرارة المتوسطة نوعا .	1:15	3	6	0	Yes
9-8	يسود في مناطق غرب و وسط و جنوب الدلتا و مصر الوسطى .	0:55	2	12	3	No
9-9	تظهر الإصابة على هيئة بقع مسحوقية لونها بني داكن أو مسود غير منتظمة تلتحم مع بعضها .	1:04	1	16	1	No
9-10	تظهر الإصابة على الساق و الأوراق و السنابل .	0:19	1	8	2	Yes

9-11	تسبب الإصابة الشديدة تهتك في الأنسجة الدعامية و الناقلة .	0:43	2	9	1	Yes
9-12	تسبب رقاد النباتات و ضعف المحصول .	0:30	1	6	1	Yes
9-13	يناسب المرض درجات الحرارة العالية نوعا .	1:22	3	6	0	Yes
9-14	المرض لا يشكل خطورة نظرا لدرجات المقاومة العالية في معظم الأصناف التجارية المنزرعة .	2:17	2	13	0	Yes
9-15	عند ظهور الإصابة بأمراض الأصداء يجب سرعة إجراء الرش الوقائي بالمبيد الفطري سومي ايت لمنع انتشار المرض و خاصة في الإصابات المبكرة و ذلك بمعدل 35 سم3 لكل 100 لتر ماء للفدان .	1:23	0	32	2	
9-16	تظهر علامات الإصابة على النباتات عند طرد السنابل .	0:38	1	8	0	Yes
9-17	يظهر محور السنبلة مغطى تماما بمسحوق أسود من جراثيم الفطر .	0:31	0	10	0	
9-18	تتطاير نتيجة اهتزاز النباتات بفعل الرياح أو غيرها .	0:33	1	8	1	No
9-19	بعد فترة يظهر محور السنبلة فقط و هو عاري تماما نتيجة تطاير جراثيم الفطر و سقوطها على مياسم الأزهار القابلة للإخصاب .	0:50	0	21	2	
9-20	تنبت الجرثومة .	0:11	2	2	0	Yes
9-21	تسلك نفس سلوك حبة اللقاح .	0:27	1	5	0	Yes
9-22	تصل إلي المبيض .	0:15	1	3	0	Yes
9-23	يسكن الفطر بجوار الجنين .	0:20	1	4	0	Yes

9-24	بعد الحصاد و الدراس لا يظهر على الحبوب أي أعراض مرضية .	0:43	1	11	1	Yes
9-25	عند زراعة الحبوب المصابة في الموسم التالي ينشط الفطر .	0:20	1	9	0	Yes
9-26	يستطيل مع استطالة النبات .	0:38	1	4	0	Yes
9-27	عند تكوين السنبلة يقضي على الحبوب .	0:25	1	6	0	Yes
9-28	يكون الجراثيم .	0:15	0	2	0	
9-29	تظهر على هيئة مسحوق أسود عند تكشف السنبلة أو طرد السنابل .	1:15	1	11	1	Yes
9-30	تعيد دورة الحياة .	0:25	2	3	0	Yes

Page # 10

No.	Text	Time	#Of Trees	# of words	# of connect-ors	Has grammati-cal tree
10-1	يلاحظ أن السنابل المتفحمة تظهر مبكرا قبل بقية السنابل بيومين أو ثلاثة .	1:17	1	12	1	Yes
10-2	لمقاومة هذا المرض يجب استعمال التقاوي المعتمدة و المعاملة بالمطهرات الفطرية .	0:25	1	11	1	Yes
10-3	معاملة التقاوي قبل الزراعة بالمبيد الفطري سومي ايت بمعدل 3 جم لكل كيلو جرام واحد من التقاوي مع الخلط الجيد .	1:12	1	20	0	No
10-4	تظهر علامات الإصابة على الأوراق و السيقان و السنابل على هيئة بقع بيضاء غير منتظمة .	0:35	1	15	2	Yes
10-5	تتحد مع بعضها .	0:10	1	3	0	Yes

10-6	يكون لها ملمس قطني .	0:37	1	4	0	Yes
10-7	يتحول اللون إلي الرمادي مع تقدم الإصابة و اصفرار الأوراق .	0:20	1	10	1	Yes
10-8	يظهر بها نقط سوداء في حجم رأس الدبوس .	0:22	0	8	0	
10-9	لمقاومة المرض يجب استعمال التقاوي المعتمدة للأصناف الموصي بها .	0:30	1	9	0	Yes
10-10	فيروس تقزم الشعير الأصفر على القمح .	0:26	1	6	0	Yes
10-11	فيه تتلون قمة أوراق نبات القمح باللون الأصفر أو القرمزي .	0:15	1	10	1	Yes
10-12	يؤدي المرض إلي نمو غير طبيعي لنبات القمح و انعدام المحصول تقريبا .	1:50	0	12	1	
10-13	ينتقل المرض عن طريق الإصابة بحشرة المن .	0:25	1	7	0	Yes
10-14	تعتبر الناقل الرئيسي للمرض .	0:10	1	4	0	Yes
10-15	لمقاومة هذا المرض تتبع نفس خطوات مقاومة حشرة المن .	0:25	1	9	0	Yes
10-16	لا ينتقل المرض إلي حقول القمح .	0:16	1	6	0	Yes
10-17	يبدأ الحصاد في أواخر شهر أبريل و أوائل مايو .	1:20	2	9	1	Yes
10-18	يجب أن يتم الحصاد بعد النضج التام مباشرة .	1:38	1	8	0	Yes

10-19	يتميز باصفرار السلامية العليا الحاملة للسنبلة في حوالي 50% من نباتات الحقل .	0:55	1	12	0	Yes
10-20	يكون الحصاد إما في الصباح الباكر أو قبل الغروب .	0:48	0	9	1	
10-21	لا يحدث فرط للحبوب أو تكسير للسنابل مع تقليل الفاقد بالعناية بعمليات النقل و الدراس .	1:30	1	15	2	Yes

APPENDIX (B)
OUTPUT FROM THE ARABIC PARSER

Parse Tree structure	No.
[جملة فعلية(فعل(تعتبر,متعدي),مفعول به(تراكيب وصفية معطوفة(تركيب وصفي(معرفة(الأراضي,مؤنث),معرفة(الصحراوية,مؤنث)),حرف وصل(أو),تراكيب وصفية معطوفة(معرفة(الرملية,مؤنث),حرف وصل(أو),تركيب وصفي(معرفة(الجيرية,مؤنث),معرفة(المستصلحة,مؤنث))))),مكمل جملة معطوفة(مكمل جملة(نكرة(حديثا,مذكر)),حرف وصل(و),مكمل جملة معطوفة(مكمل جملة مركب(مكمل جملة(معرفة(المروية,مؤنث)),مكمل جملة مركب(مكمل جملة(نكرة(جزءا,مذكر)),مكمل جملة مركب(مكمل جملة(نكرة(هاما,مذكر)),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(في),تركيب إضافي(نكرة(خطة,مؤنث),تركيب وصفي(معرفة(التوسع,مذكر),معرفة(الأفقي,مذكر))))),مكمل جملة(شبه جملة(حرف جر(في),تركيب وصفي(معرفة(الأراضي,مؤنث),معرفة(الزراعية,مؤنث)))))))),حرف وصل(و),مكمل جملة(شبه جملة(حرف جر(في),تركيب وصفي(معرفة(الإنتاج,مذكر),معرفة(الزراعي,مذكر)))))),جملة فعلية(فعل(تعتبر,متعدي),فاعل(تراكيب وصفية معطوفة(تركيب وصفي(معرفة(الأراضي,مؤنث),معرفة(الصحراوية,مؤنث)),حرف وصل(أو),تراكيب وصفية معطوفة(معرفة(الرملية,مؤنث),حرف وصل(أو),تركيب وصفي(معرفة(الجيرية,مؤنث),معرفة(المستصلحة,مؤنث))))),مكمل جملة معطوفة(مكمل جملة(نكرة(حديثا,مذكر)),حرف وصل(و),مكمل جملة(معرفة(المروية,مؤنث))),مفعول به(تركيب وصفي(نكرة(جزءا,مذكر),نكرة(هاما,مذكر))),مكمل جملة معطوفة(مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(في),تركيب إضافي(نكرة(خطة,مؤنث),تركيب وصفي(معرفة(التوسع,مذكر),معرفة(الأفقي,مذكر))))),مكمل جملة(شبه جملة(حرف جر(في),تركيب وصفي(معرفة(الأراضي,مؤنث),معرفة(الزراعية,مؤنث)))))),حرف وصل(و),مكمل جملة(شبه جملة(حرف جر(في),تركيب وصفي(معرفة(الإنتاج,مذكر),معرفة(الزراعي,مذكر)))))))]	1-1
]جملة فعلية يتصدرها مكمل جملة مقدم(مكمل جملة مركب(مكمل جملة(نكرة(نظرا,مذكر)),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ل),تركيب إضافي(نكرات متعددة(نكرة(أهم,مؤنث),نكرة(محصول,مذكر)),معرفة(القمح,مذكر)))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي)تركيب إضافي(نكرة(اعتبار,مذكر),ضمير_متصل(ه)),تركيب وصفي(معرفة(المحصول,مذكر),تركيب وصفي(معرفة(الغذائي,مذكر),معرفة(الأول,مذكر))))),مكمل جملة(شبه جملة(حرف جر(في),معرفة_علم(مصر,مؤنث))))), جملة إن و أخواتها(حرف إن (إن),ضمير_متصل(ه),خبر(جملة فعلية(فعل(يعتمد,لازم),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(على),تركيب بدل(اسم اشارة(تلك),معرفة(الأراضي,مؤنث)))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(في),تركيب إضافي(نكرة(إنتاج,مذكر),معرفة(القمح,مذكر)))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ل),تركيب إضافي(نكرة(تقليل,مذكر),معرفة(الفجوة,مؤنث)))),مكمل جملة مركب(مكمل جملة(تركيب إضافي(ظرف مكان(بين),تراكيب وصفية معطوفة(تركيب وصفي(معرفة(المنتج,مذكر),معرفة(المحلي,مذكر)),حرف وصل(و),معرفة(المستهلك,مذكر))),مكمل جملة(شبه جملة(حرف جر(من),تركيب إضافي(نكرة(حبوب,مؤنث),معرفة(القمح,مذكر)))))))))))]	1-2
[جملة أسمية(مبتدأ(معرفة(القمح,مذكر)),خبر(خبر يسبقه مكمل جملة(مكمل جملة(شبه جملة(حرف جر(في),تركيب بدل(اسم اشارة(هذه),معرفة(الأراضي,مؤنث)))),جملة فعلية(فعل(يتعرض,لازم),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف	1-3

	جر(ل,نكرة(ظروف,مؤنث))),مكمل جملة مركب(مكمل جملة(تركيب إضافي(ظرف زمان(غير),نكرة(ملائمة,مؤنث))),مكمل جملة مركب(مكمل جملة(نكرة(تبعا,مذكر)),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ل),تراكيب إضافية معطوفة(تركيب إضافي(نكرة(نوعية,مؤنث),معرفة(التربة,مؤنث)),حرف وصل(و),تراكيب إضافية معطوفة(تركيب إضافي(نكرات متعددة(نكرة(قلة,مؤنث),نكرة(خصوبة,مؤنث)),ضمير_متصل(ها)),حرف وصل(و),تركيب إضافي(نكرات متعددة(نكرة(قلة,مؤنث),نكرة(احتفاظ,مذكر)),ضمير_متصل(ها))))),مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(نكرة(مياه,مؤنث),معرفة(الري,مذكر))))))))))),جملة فعلية يتصدرها مكمل جملة مقدم(مكمل جملة مركب(مكمل جملة(معرفة(القمح,مذكر)),مكمل جملة(شبه جملة(حرف جر(في),تركيب بدل(اسم اشارة(هذه),معرفة(الأراضي,مؤنث))))),جملة فعلية(فعل(يتعرض,لازم),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ل),نكرة(ظروف,مؤنث))),مكمل جملة مركب(مكمل جملة(تركيب إضافي(ظرف زمان(غير),نكرة(ملائمة,مؤنث))),مكمل جملة مركب(مكمل جملة(نكرة(تبعا,مذكر)),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ل),تراكيب إضافية معطوفة(تركيب إضافي(نكرة(نوعية,مؤنث),معرفة(التربة,مؤنث)),حرف وصل(و),تراكيب إضافية معطوفة(تركيب إضافي(نكرات متعددة(نكرة(قلة,مؤنث),نكرة(خصوبة,مؤنث)),ضمير_متصل(ها)),حرف وصل(و),تركيب إضافي(نكرات متعددة(نكرة(قلة,مؤنث),نكرة(احتفاظ,مذكر)),ضمير_متصل(ها))))),مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(نكرة(مياه,مؤنث),معرفة(الري,مذكر))))))))))]
1-4	[جملة أسمية(خبر مقدم(شبه جملة(حرف جر(فمن),معرفة(المهم,مذكر))),مبتدأ(تركيب إضافي(نكرة(اتباع,مذكر),تركيب وصفي(معرفة(التوصيات,مؤنث),تركيب وصفي(معرفة(الفنية,مؤنث),معرفة(الخاصة,مؤنث))))),مكمل جملة معطوفة(مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(نكرة(إنتاج,مذكر),معرفة(القمح,مذكر)))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(في),تركيب بدل(اسم اشارة(تلك),معرفة(المناطق,مؤنث)))),مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(ظرف زمان(كل),نكرة(دقة,مؤنث))))),حرف وصل(و),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(نكرة(قدر,مذكر),معرفة(الإمكان,مذكر)))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(لل),معرفة(توصل,مذكر))),مكمل جملة(شبه جملة(حرف جر(إلي),تركيب وصفي(نكرة(محصول,مذكر),نكرة(جيد,مذكر)))))))]
1-5	[جملة أسمية(مبتدأ(معرفة_علم(سخا8,مذكر)),خبر(شبه جملة(حرف جر(من),معرفة(الأصناف,مؤنث))))]
1-6	[جملة فعلية(فعل(تلقى,متعدي),مكمل جملة مركب(مكمل جملة(نكرة(قبولا,مذكر)),مكمل جملة مركب(مكمل جملة(نكرة(كثيرا,مذكر)),مكمل جملة مركب(مكمل جملة(تركيب إضافي(ظرف مكان(لدى),تركيب إضافي(نكرة(زراع,مذكر),معرفة(القمح,مذكر)))),مكمل جملة(شبه جملة(حرف جر(في),تركيب وصفي(معرفة(الأراضي,مؤنث),معرفة(الجديدة,مؤنث)))))))),جملة فعلية(فعل(تلقى,متعدي),مفعول به(تركيب وصفي(نكرة(قبولا,مذكر),نكرة(كثيرا,مذكر))),مكمل جملة مركب(مكمل جملة(تركيب إضافي(ظرف مكان(لدى),تركيب إضافي(نكرة(زراع,مذكر),معرفة(القمح,مذكر)))),مكمل جملة(شبه جملة(حرف جر(في),تركيب وصفي(معرفة(الأراضي,مؤنث),معرفة(الجديدة,مؤنث)))))))]
1-7	[جملة فعلية(فعل(يجود,لازم),مكمل جملة(شبه جملة(حرف جر(في),تركيب وصفي(معرفة(الأراضي,مؤنث),معرفة(الجديدة,مؤنث)))))]
1-8	[جملة فعلية(فعل(ترتفع,لازم),مكمل جملة(شبه جملة(حرف جر(ب),ضمير_متصل(ها))),فاعل(تركيب إضافي(نكرة(نسبة,مؤنث),معرفة(الملوحة,مؤنث))))]
1-9	[جملة فعلية(فعل(يزرع,متعدي),مفعول به(تركيب بدل(اسم اشارة(هذا),معرفة(الصنف,مذكر))),مكمل جملة(شبه جملة(حرف جر(في),تركيب إضافي(نكرات

	متعددة(نكرة(معظم,مذكر),نكرة(أراضي,مؤنث)),معرفة(الجمهورية,مؤنث))))))]
1-10	[جملة أسمية(مبتدأ(معرفة_علم(جيزة167,مذكر)),خبر(شبه جملة(حرف جر(من),تركيب وصفي(معرفة(الأصناف,مؤنث),معرفة(الجديدة,مؤنث))))))]
1-11	[جملة فعلية(فعل(تجود,لازم),فاعل(تركيب إضافي(نكرة(زراعة,مؤنث),ضمير_متصل(ه))),مكمل جملة(شبه جملة(حرف جر(في),تركيب وصفي(معرفة(الأراضي,مؤنث),معرفة(الجديدة,مؤنث))))))]
1-12	[جملة أسمية(مبتدأ(معرفة_علم(سدس1,مذكر)),خبر(خبر يسبقه مكمل جملة(مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(من),تركيب وصفي(معرفة(الأصناف,مؤنث),معرفة(الجديدة,مؤنث))))),مكمل جملة مركب(مكمل جملة(تركيب إضافي(نكرة(عالية,مؤنث),معرفة(المحصول,مذكر))),مكمل جملة مركب(مكمل جملة(معرفة(المقاومة,مؤنث)),مكمل جملة(شبه جملة(حرف جر(ل),معرفة(أمراض,مؤنث))))))),جملة فعلية(فعل(تجود,لازم),فاعل(تركيب إضافي(نكرة(زراعة,مؤنث),ضمير_متصل(ها))),مكمل جملة(شبه جملة(حرف جر(في),تركيب وصفي(معرفة(الأراضي,مؤنث),معرفة(الجديدة,مؤنث)))))))),جملة فعلية يتصدرها مكمل جملة مقدم(مكمل جملة مركب(مكمل جملة(معرفة_علم(سدس1,مذكر)),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(من),تركيب وصفي(معرفة(الأصناف,مؤنث),معرفة(الجديدة,مؤنث))))),مكمل جملة مركب(مكمل جملة(تركيب إضافي(نكرة(عالية,مؤنث),معرفة(المحصول,مذكر))),مكمل جملة مركب(مكمل جملة(معرفة(المقاومة,مؤنث)),مكمل جملة(شبه جملة(حرف جر(ل),معرفة(أمراض,مؤنث))))))),جملة فعلية(فعل(تجود,لازم),فاعل(تركيب إضافي(نكرة(زراعة,مؤنث),ضمير_متصل(ها))),مكمل جملة(شبه جملة(حرف جر(في),تركيب وصفي(معرفة(الأراضي,مؤنث),معرفة(الجديدة,مؤنث)))))))]
1-13	[جملة أسمية(مبتدأ(معرفة(الصنف,معرفة_علم(جميزة3,مذكر),مذكر)),خبر(جملة فعلية(فعل(يشترط,متعدي),مفعول به(تركيب إضافي(نكرات متعددة(نكرة(عدم,مذكر),نكرة(زراعة,مؤنث)),ضمير_متصل(ه))),مكمل جملة(نكرة(مبكرا,مذكر)))),جملة فعلية يتصدرها مكمل جملة مقدم(مكمل جملة(معرفة(الصنف,معرفة_علم(جميزة3,مذكر),مذكر)),جملة فعلية(فعل(يشترط,متعدي),فاعل(تركيب إضافي(نكرات متعددة(نكرة(عدم,مذكر),نكرة(زراعة,مؤنث)),ضمير_متصل(ه))),مفعول به(نكرة(مبكرا,مذكر)))),جملة فعلية يتصدرها مكمل جملة مقدم(مكمل جملة(معرفة(الصنف,معرفة_علم(جميزة3,مذكر),مذكر)),جملة فعلية(فعل(يشترط,متعدي),مفعول به(تركيب إضافي(نكرات متعددة(نكرة(عدم,مذكر),نكرة(زراعة,مؤنث)),ضمير_متصل(ه))),مكمل جملة(نكرة(مبكرا,مذكر))))]
1-14	[جملة إن و أخواتها(حرف إن (إن),ضمير_متصل(ه),خبر(تراكيب إضافية معطوفة(تركيب إضافي(نكرة(مبكر,مذكر),معرفة(التزهير,مذكر)),حرف وصل(و),تركيب إضافي(نكرة(مبكر,مذكر),معرفة(النضج,مذكر)))))]
1-15	[جملة إن و أخواتها(حرف إن (إن),جملة أسمية(مبتدأ(تركيب بدل(اسم اشارة(هذا),معرفة(الصنف,مذكر))),خبر(خبر و متعلق به(نكرة(حساس,مذكر),مكمل جملة معطوفة(مكمل جملة(شبه جملة(حرف جر(ل),معرفة(رطوبة,مؤنث))),حرف وصل(و),مكمل جملة مركب(مكمل جملة(تركيب إضافي(نكرات متعددة(نكرة(زيادة,مؤنث),نكرات متعددة(نكرة(كمية,مؤنث),نكرة(مياه,مؤنث))),معرفة(الري,مذكر))),مكمل جملة(تركيب إضافي(ظرف مكان(عند),معرفة(الزراعة,مؤنث)))))))))]
1-16	[جملة فعلية(فعل(ينصح,متعدي),مكمل جملة(حرف جر(ب), جملة إن و أخواتها(حرف إن (إن),خبر(جملة كان و أخواتها(فعل(تكون,متعدي),جملة أسمية(مبتدأ(تركيب إضافي(نكرة(رية,مؤنث),معرفة(الزراعة,مؤنث))),خبر(خبر و متعلق به(شبه جملة(حرف جر(على),معرفة(الحامي,مذكر)),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(في),تركيب إضافي(نكرة(حالة,مؤنث),تركيب وصفي(معرفة(الري,مذكر),معرفة(السطحي,مذكر))))),مكمل جملة(شبه جملة(حرف جر(في),تركيب

	وصفي(معرفة(الأراضي,مؤنث),معرفة(الجيرية,مؤنث)))))))))))]
1-17	[جملة فعلية(فعل(تحرث,متعدي),مفعول به(معرفة(الأرض,مؤنث)),مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(نكرة(استخدام,مذكر),تركيب وصفي(معرفة(المحراث,مذكر),معرفة(الحفار,مذكر)))))]
1-18	[جملة فعلية(فعل(تنعم,متعدي),مفعول به(نكرة(جيدا,مذكر)),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(نكرة(استخدام,مذكر),تركيب وصفي(معرفة(المحراث,مذكر),معرفة(القرصي,مذكر))))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(مع),تركيب إضافي(نكرة(نقاوة,مؤنث),معرفة(الحشائش,مؤنث)))),مكمل جملة مركب(مكمل جملة(نكرة(خاصة,مؤنث)),مكمل جملة(شبه جملة(حرف جر(في),تركيب إضافي(نكرات متعددة(نكرة(حالة,مؤنث),نكرات متعددة(نكرة(استعمال,مذكر),نكرة(آلات,مؤنث))),معرفة(التسطير,مذكر)))))))),جملة فعلية(فعل(تنعم,متعدي),مكمل جملة مركب(مكمل جملة(نكرة(جيدا,مذكر)),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(نكرة(استخدام,مذكر),تركيب وصفي(معرفة(المحراث,مذكر),معرفة(القرصي,مذكر))))),مكمل جملة(شبه جملة(حرف جر(مع),تركيب إضافي(نكرة(نقاوة,مؤنث),معرفة(الحشائش,مؤنث)))),مفعول به(نكرة(خاصة,مؤنث)),مكمل جملة(شبه جملة(حرف جر(في),تركيب إضافي(نكرات متعددة(نكرة(حالة,مؤنث),نكرات متعددة(نكرة(استعمال,مذكر),نكرة(آلات,مؤنث))),معرفة(التسطير,مذكر)))))]

Parse Tree structure	No.
[جملة فعلية(فعل(تقسم,متعدي),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(إلي),نكرة(أحواض,مؤنث))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),نكرة(مساحة,مؤنث))),مكمل جملة(تركيب إضافي(ظرف زمان(حوالي),تركيب إضافي(وحدة رقمية(3*4),نكرة(متر,مذكر)))))))]	2-1
[جملة فعلية(فعل(تحرث,متعدي),مفعول به(معرفة(الأرض,مؤنث)),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(نكرة(استخدام,مذكر),تركيب وصفي(معرفة(المحراث,مذكر),معرفة(القرصي,مذكر))))),مكمل جملة(شبه جملة(حرف جر(مع),تركيب إضافي(نكرة(نقاوة,مؤنث),معرفة(الحشائش,مؤنث)))))]	2-2
[جملة فعلية(فعل(تسوى,متعدي),مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(نكرة(قدر,مذكر),معرفة(الإمكان,مذكر)))),مفعول به(نكرة(خاصة,مؤنث)),مكمل جملة(تركيب إضافي(ظرف مكان(عند),تركيب إضافي(نكرات متعددة(نكرة(استعمال,مذكر),نكرة(آلات,مؤنث)),معرفة(التسطير,مذكر)))))]	2-3
[جملة فعلية مركبة(جملة فعلية(فعل(تعتبر,متعدي),مفعول به(معرفة(الزراعة,معرفة_علم - أعلام متتابعة(العفير,معرفة_علم(بدار,مذكر),مذكر),مؤنث)),مكمل جملة مركب(مكمل جملة - ضمير شأن للتوكيد(ضمير_منفصل(هي)),مكمل جملة مركب(مكمل جملة(تركيب تمييزي(نكرة(أكثر,مذكر),تركيب إضافي(نكرة(طرق,مؤنث),معرفة(الزراعة,مؤنث)))),مكمل جملة(نكرة(استخداما,مذكر))),حرف وصل(و),جملة فعلية يتصدرها مكمل جملة مقدم(مكمل جملة(شبه جملة(حرف جر(في),ضمير_متصل(ها))),جملة فعلية(فعل(تبذر,متعدي),فاعل(معرفة(التقاوي,مؤنث)),مفعول به(تركيب إضافي(ظرف زمان(عقب),تركيب إضافي(نكرات(نكرات معطوفة(نكرة(حرث,مذكر),حرف وصل(و),نكرة(خدمة,مؤنث))),معرفة(الأرض,مؤنث))))))),جملة فعلية مركبة(جملة فعلية(فعل(تعتبر,متعدي),مفعول به(معرفة(الزراعة,معرفة_علم - أعلام متتابعة(العفير,معرفة_علم(بدار,مذكر),مذكر),مؤنث)),مكمل جملة مركب(مكمل جملة - ضمير شأن للتوكيد(ضمير_منفصل(هي)),مكمل جملة مركب(مكمل جملة(تركيب تمييزي(نكرة(أكثر,مذكر),تركيب إضافي(نكرة(طرق,مؤنث),معرفة(الزراعة,مؤنث)))),مكمل	2-4

	جملة(نكرة(استخداما,مذكر)))),حرف وصل(و),جملة فعلية يتصدرها مكمل جملة مقدم(مكمل جملة(شبه جملة(حرف جر(في),ضمير_متصل(ها))),جملة فعلية(فعل(تبذر,متعدي),مفعول به(معرفة(التقاوي,مؤنث)),مكمل جملة(تركيب إضافي(ظرف زمان(عقب),تركيب إضافي(نكرات(نكرات معطوفة(نكرة(حرث,مذكر),حرف وصل(و),نكرة(خدمة,مؤنث))),معرفة(الأرض,مؤنث)))))),جملة فعلية مركبة(جملة فعلية(فعل(تعتبر,متعدي),فاعل(معرفة(الزراعة,معرفة_علم - أعلام متتابعة(العفير,معرفة_علم(بدار,مذكر),مذكر),مؤنث)),مكمل جملة - ضمير شأن للتوكيد(ضمير_منفصل(هي)),مفعول به(تركيب تمييزي(نكرة(أكثر,مذكر),تركيب إضافي(نكرة(طرق,مؤنث),معرفة(الزراعة,مؤنث)))),مكمل جملة(نكرة(استخداما,مذكر))),حرف وصل(و),جملة فعلية يتصدرها مكمل جملة مقدم(مكمل جملة(شبه جملة(حرف جر(في),ضمير_متصل(ها))),جملة فعلية(فعل(تبذر,متعدي),فاعل(معرفة(التقاوي,مؤنث)),مفعول به(تركيب إضافي(ظرف زمان(عقب),تركيب إضافي(نكرات(نكرات معطوفة(نكرة(حرث,مذكر),حرف وصل(و),نكرة(خدمة,مؤنث))),معرفة(الأرض,مؤنث)))))),جملة فعلية مركبة(جملة فعلية(فعل(تعتبر,متعدي),فاعل(معرفة(الزراعة,معرفة_علم - أعلام متتابعة(العفير,معرفة_علم(بدار,مذكر),مذكر),مؤنث)),مكمل جملة - ضمير شأن للتوكيد(ضمير_منفصل(هي)),مفعول به(تركيب تمييزي(نكرة(أكثر,مذكر),تركيب إضافي(نكرة(طرق,مؤنث),معرفة (الزراعة,مؤنث)))),مكمل جملة(نكرة(استخداما,مذكر))),حرف وصل(و),جملة فعلية يتصدرها مكمل جملة مقدم(مكمل جملة(شبه جملة(حرف جر(في),ضمير_متصل(ها))),جملة فعلية(فعل(تبذر,متعدي),مفعول به(معرفة(التقاوي,مؤنث)),مكمل جملة(تركيب إضافي(ظرف زمان(عقب),تركيب إضافي(نكرات(نكرات معطوفة(نكرة(حرث,مذكر),حرف وصل(و),نكرة(خدمة,مؤنث))),معرفة(الأرض,مؤنث)))))))]
2-5	[جملة فعلية(فعل(تغطى,متعدي),مكمل جملة مركب(مكمل جملة(نكرة(جيدا,مذكر)),مكمل جملة(شبه جملة(حرف جر(ب),تركيب وصفي(نكرات متعددة(نكرة(أي,حياد),نكرة(وسيلة,مؤنث)),نكرة(ممكنة,مؤنث)))))),جملة فعلية(فعل(تغطى,متعدي),مفعول به(نكرة(جيدا,مذكر)),مكمل جملة(شبه جملة(حرف جر(ب),تركيب وصفي(نكرات متعددة(نكرة(أي,حياد),نكرة(وسيلة,مؤنث)),نكرة(ممكنة,مؤنث)))))]
2-6	[جملة فعلية(فعل(تقسم,متعدي),مفعول به(معرفة(الأرض,مؤنث)),مكمل جملة معطوفة(مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(إلي),تركيب وصفي(نكرة(أحواض,مؤنث),نكرة(صغيرة,مؤنث)))),مكمل جملة مركب(مكمل جملة(نكرة(نسبيا,مذكر)),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ل),تركيب إضافي(نكرة(إحكام,مذكر),معرفة(الري,مذكر)))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(في),تركيب إضافي(نكرة(حالة,مؤنث),تركيب وصفي(معرفة(الأراضي,مؤنث),معرفة(المروية,مؤنث)))),مكمل جملة(شبه جملة(حرف جر(بال),معرفة(غمر,مذكر))))))),حرف وصل(أو),مكمل جملة(تركيب وصفي(نكرة(ريا,مذكر),نكرة(سطحيا,مذكر)))),جملة فعلية(فعل(تقسم,متعدي),مفعول به(معرفة(الأرض,مؤنث)),مكمل جملة معطوفة(مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(إلي),تركيب وصفي(نكرة(أحواض,مؤنث),نكرة(صغيرة,مؤنث)))),مكمل جملة مركب(مكمل جملة(نكرة(نسبيا,مذكر)),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ل),تركيب إضافي(نكرة(إحكام,مذكر),معرفة(الري,مذكر)))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(في),تركيب إضافي(نكرة(حالة,مؤنث),تركيب وصفي(معرفة(الأراضي,مؤنث),معرفة(المروية,مؤنث)))),مكمل جملة(شبه جملة(حرف جر(بال),معرفة(غمر,مذكر))))))),حرف وصل(أو),مكمل جملة مركب(مكمل جملة(نكرة(ريا,مذكر)),مكمل جملة(نكرة(سطحيا,مذكر)))))]
2-7	[جملة فعلية(فعل(تترك,متعدي),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(نكرات متعددة(نكرة(دون,حياد),نكرة(تقسيم,مذكر)))))),مكمل جملة مركب(مكمل جملة(شبه

	جملة(حرف جر(في),تركيب إضافي(نكرة(حالة,مؤنث),معرفة(الري,مذكر))),مكمل جملة(شبه جملة(حرف جر(بال),تراكيب وصفية معطوفة(تركيب وصفي(معرفة(رش,مذكر),معرفة(المحوري,مذكر)),حرف وصل(أو),معارف معطوفة(معرفة(المتنقل,مذكر),حرف وصل(أو),معرفة(الثابت,مذكر))))))),جملة فعلية(فعل(تترك,متعدي),مكمل جملة معطوفة(مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(نكرات متعددة(نكرة(دون,حياد),نكرة(تقسيم,مذكر))))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(في),تركيب إضافي(نكرة(حالة,مؤنث),معرفة(الري,مذكر))),مكمل جملة(شبه جملة(حرف جر(بال),تراكيب وصفية معطوفة(تركيب وصفي(معرفة(رش,مذكر),معرفة(المحوري,مذكر)),حرف وصل(أو),معرفة(المتنقل,مذكر))))),حرف وصل(أو),مكمل جملة(معرفة(الثابت,مذكر))))]
2-8	[جملة أسمية(مبتدأ(معرفة(الزراعة,مؤنث)),خبر(خبر يسبقه مكمل جملة(مكمل جملة(شبه جملة(حرف جر(بال),معرفة(تسطير,مذكر))),جملة أسمية(مبتدأ(ضمير_منفصل(هي)),خبر(خبر و متعلق به(تركيب تمييزي(نكرة(أفضل,مذكر),تركيب إضافي(نكرة(طرق,مؤنث),معرفة(الزراعة,مؤنث))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(في),تركيب بدل(اسم اشارة(تلك),معرفة(الأراضي,مؤنث))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ل),تراكيب إضافية معطوفة(تركيب إضافي(نكرات متعددة(نكرة(ضمان,مذكر),نكرة(توزيع,مذكر))),حرف وصل(و),تركيب وصفي(نكرة(تغطية,مؤنث),نكرة(جيدة,مؤنث))))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ل),معرفة(تقاوي,مؤنث))),مكمل جملة(شبه جملة(حرف جر(مع),تركيب وصفي(نكرات متعددة(نكرة(عمق,مذكر),نكرة(زراعة,مؤنث)),نكرة(مناسب,مذكر)))))))))))]
2-9	[جملة فعلية يتصدرها مكمل جملة مقدم(مكمل جملة(شبه جملة(حرف جر(في),ضمير_متصل(ها))),جملة فعلية(فعل(تعاير,متعدي),مفعول به(معرفة(السطارة,مؤنث))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ل),معرفة(زراعة,مؤنث))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(نكرة(معدل,مذكر),تركيب إضافي(وحدة رقمية(60),نكرات متعددة(نكرة(كيلو,مذكر),نكرة(جرام,مذكر))))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ل),معرفة(فدان,مذكر))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(في),نكرة(سطور,مؤنث))),مكمل جملة(شبه جملة(حرف جر(على),تراكيب إضافية معطوفة(تركيب إضافي(نكرة(مسافات,مؤنث),تركيب إضافي(وحدة رقمية(125-13),نكرة(سم,مذكر))),حرف وصل(و),تركيب إضافي(نكرة(عمق,مذكر),تركيب إضافي(وحدة رقمية(4-5),نكرة(سم,مذكر)))))))))))]
2-10	[جملة فعلية يتصدرها مكمل جملة مقدم(مكمل جملة(شبه جملة(حرف جر(في),تركيب إضافي(نكرة(حالة,مؤنث),معرفة(الزراعة,معرفة_علم - أعلام متتابعة(العفير,معرفة_علم(بدار,مذكر),مذكر),مؤنث)))),جملة فعلية(فعل(يستخدم,متعدي),مفعول به(تركيب إضافي(وحدة رقمية(75),نكرات متعددة(نكرة(كيلو,مذكر),نكرات متعددة(نكرة(جرام,مذكر),نكرة(تقاوي,مؤنث))))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ل),معرفة(فدان,مذكر))),مكمل جملة(حرف جر(على), جملة إن و أخواتها(حرف إن (إن),خبر(جملة فعلية(فعل(تبذر,متعدي),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),نكرة(انتظام,مذكر))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(على),معرفة(الأرض,مؤنث))),مكمل جملة(شبه جملة(حرف جر(مع),تركيب وصفي(معرفة(التغطية,مؤنث),معرفة(الجيدة,مؤنث)))))))))))]
2-11	[جملة فعلية يتصدرها مكمل جملة مقدم(مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(نكرات متعددة(نكرة(استعمال,مذكر),نكرة(آلة,مؤنث)),معرفة(التسطير,مذكر)))),جملة فعلية(فعل(يستخدم,متعدي),مفعول به(تركيب إضافي(وحدة رقمية(60),نكرات متعددة(نكرة(كيلو,مذكر),نكرات متعددة(نكرة(جرام,مذكر),نكرة(تقاوي,مؤنث))))),مكمل جملة(شبه جملة(حرف جر(ل),معرفة(فدان,مذكر)))))]

2-12	[جملة فعلية يتصدرها مكمل جملة مقدم(مكمل جملة(شبه جملة(حرف جر(في),تركيب إضافي(نكرة(حالة,مؤنث),معرفة(الزراعة,معرفة_علم(العفير,مذكر),مؤنث))))),جملة فعلية(فعل(يستخدم,متعدي),مفعول به(تركيب إضافي(وحدة رقمية(85),نكرات متعددة(نكرة(كيلو,مذكر),نكرات متعددة(نكرة(جرام,مذكر),نكرة(تقاوي,مؤنث))))),مكمل جملة(شبه جملة(حرف جر(لل),معرفة(فدان,مذكر)))))]
2-13	[جملة أسمية(مبتدأ(تركيب إضافي(نكرات متعددة(نكرة(توفير,مذكر),نكرة(كمية,مؤنث)),تركيب وصفي(معرفة(التقاوي,مؤنث),معرفة(المستخدمة,مؤنث)))),خبر(شبه جملة(حرف جر(في),معرفة(الزراعة,مؤنث))))]
2-14	[جملة مركبة(جملة أسمية(مبتدأ(تركيب إضافي(نكرات متعددة(نكرة(انتظام,مذكر),نكرة(توزيع,مذكر)),معرفة(التقاوي,مؤنث))),خبر(شبه جملة(حرف جر(في),معرفة(الحقل,مذكر)))),رابط جمل(حرف وصل(و)),جملة أسمية(مبتدأ(تراكيب إضافية معطوفة(تركيب إضافي(نكرات متعددة(نكرة(انتظام,مذكر),نكرة(عمق,مذكر)),معرفة(الزراعة,مؤنث)),حرف وصل(و),تركيب إضافي(نكرات متعددة(نكرة(ضمان,مذكر),نكرة(تغطية,مؤنث)),معرفة(الحبوب,مؤنث)))),خبر(تركيب إضافي(ظرف زمان(عقب),معرفة(الزراعة,مؤنث)))))]
2-15	[جملة فعلية(فعل(يؤدي,متعدي),مكمل جملة معطوفة(مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(إلي),تراكيب إضافية معطوفة(تركيب إضافي(نكرات(نكرة(زيادة,مؤنث),نكرات معطوفة(نكرة(سرعة,مؤنث),حرف وصل(و),نكرة(نسبة,مؤنث))),معرفة(الإنبات,مذكر)),حرف وصل(و),تراكيب إضافية معطوفة(تركيب إضافي(نكرات متعددة(نكرة(انتظام,مذكر),نكرة(نمو,مذكر)),معرفة(النباتات,مؤنث)),حرف وصل(و),تراكيب إضافية معطوفة(تركيب إضافي(نكرة(جودة,مؤنث),معرفة(التفريع,مذكر)),حرف وصل(و),تركيب إضافي(نكرات متعددة(نكرة(تقليل,مذكر),نكرة(منافسة,مؤنث)),معرفة(النباتات,مؤنث)))))),مكمل جملة(شبه جملة(حرف جر(ل),تركيب إضافي(ظرف مكان(بعض),ضمير_متصل(ها))))),حرف وصل(و),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(بال),معرفة(تالي,مذكر))),مكمل جملة مركب(مكمل جملة(تركيب إضافي(نكرة(زيادة,مؤنث),معرفة(المحصول,مذكر))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(من),معرفة(الحبوب,مؤنث))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(ظرف زمان(حوالي),تركيب إضافي(نسبة مئوية(20%))))),مكمل جملة(شبه جملة(حرف جر(عن),تركيب وصفي(معرفة(الزراعة,مؤنث),معرفة(اليدوية,مؤنث)))))))))),جملة فعلية(فعل(يؤدي,متعدي),مكمل جملة معطوفة(مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(إلي),تراكيب إضافية معطوفة(تركيب إضافي(نكرات(نكرة(زيادة,مؤنث),نكرات معطوفة(نكرة(سرعة,مؤنث),حرف وصل(و),نكرة(نسبة,مؤنث))),معرفة(الإنبات,مذكر)),حرف وصل(و),تراكيب إضافية معطوفة(تركيب إضافي(نكرات متعددة(نكرة(انتظام,مذكر),نكرة(نمو,مذكر)),معرفة(النباتات,مؤنث)),حرف وصل(و),تراكيب إضافية معطوفة(تركيب إضافي(نكرة(جودة,مؤنث),معرفة(التفريع,مذكر)),حرف وصل(و),تركيب إضافي(نكرات متعددة(نكرة(تقليل,مذكر),نكرة(منافسة,مؤنث)),معرفة(النباتات,مؤنث)))))),مكمل جملة(شبه جملة(حرف جر(ل),تركيب إضافي(ظرف مكان(بعض),ضمير_متصل(ها))))),حرف وصل(و),مكمل جملة(شبه جملة(حرف جر(بال),معرفة(تالي,مذكر)))),مفعول به(تركيب إضافي(نكرة(زيادة,مؤنث),معرفة(المحصول,مذكر))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(من),معرفة(الحبوب,مؤنث))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(ظرف زمان(حوالي),تركيب إضافي(نسبة مئوية(20%))))),مكمل جملة(شبه جملة(حرف جر(عن),تركيب وصفي(معرفة(الزراعة,مؤنث),معرفة(اليدوية,مؤنث)))))))]
2-16	[جملة أسمية(خبر مقدم(أشباه جملة متعددة(شبه جملة(حرف جر(من),تركيب إضافي(نكرة(مميزات,مؤنث),معرفة(الزراعة,مؤنث))),شبه جملة(حرف جر(ب),تركيب إضافي(نكرة(آلة,مؤنث),معرفة(التسطير,مذكر))))),مبتدأ(تراكيب إضافية معطوفة(تركيب

	إضافي(نكرات متعددة(نكرة(توفير,مذكر),نكرة(وقت,مذكر)),معرفة(الزراعة,مؤنث)),حرف وصل(و),تركيب إضافي(نكرة(نفقات,مؤنث),تركيب وصفي(معرفة(العمالة,مؤنث),معرفة(اليدوية,مؤنث))))))]
2-17	[جملة أسمية(مبتدأ(تركيب إضافي(نكرات متعددة(نكرة(إمكانية,مؤنث),نكرة(استعمال,مذكر)),تركيب وصفي(معرفة(الماكينات,مؤنث),معرفة(المجهزة,مؤنث)))),خبر(خبر و متعلق به(شبه جملة(حرف جر(لل),معرفة(تسميد,مذكر)),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(بال),تركيب وصفي(معرفة(جرعة,مؤنث),معرفة(التنشيطية,مؤنث)))),مكمل جملة(شبه جملة(حرف جر(مع),معرفة(الزراعة,مؤنث)))))))]

No.	Parse Tree structure
3-1	[جملة أسمية(مبتدأ(تركيب تمييزي(نكرة(أفضل,مذكر),نكرة(ميعاد,مذكر))),خبر(خبر يسبقه مكمل جملة(مكمل جملة(شبه جملة(حرف جر(لل),معرفة(زراعة,مؤنث))),جملة أسمية(مبتدأ(ضمير_منفصل(هو)),خبر(خبر و متعلق به(خبر(معرفة(الفترة,مؤنث)),مكمل جملة(شبه جملة(حرف جر(من),تركيب إضافي(وحدة رقمية(15,حرف جر(إلي),30),معرفة_علم(نوفمبر,مذكر)))))))))]
3-2	[جملة فعلية(فعل(يراعى,متعدي),مفعول به(تركيب إضافي(نكرة(ضرورة,مؤنث),معرفة(الالتزام,مذكر))),مكمل جملة معطوفة(مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(نكرة(ميعاد,مذكر),معرفة(الزراعة,مؤنث)))),مكمل جملة مركب(مكمل جملة(معرفة(الموصي,مذكر)),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),ضمير_متصل(ه))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ل),تركيب إضافي(ظرف زمان(كل),نكرة(صنف,مذكر))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ل),تركيب إضافي(نكرة(تفادي,مذكر),معرفة(الإصابة,مؤنث)))),مكمل جملة(شبه جملة(حرف جر(بال),معرفة(من,مذكر)))))),حرف وصل(و),مكمل جملة مركب(مكمل جملة(تركيب إضافي(نكرة(تقليل,مذكر),معرفة(الفاقد,مذكر))),مكمل جملة(شبه جملة(حرف جر(من),تركيب إضافي(نكرة(مهاجمة,مؤنث),معرفة(العصافير,مؤنث)))))))]
3-3	UnParseable
3-4	[جملة فعلية(فعل(يعمل,لازم),فاعل(تركيب وصفي(معرفة(السماد,مذكر),معرفة(البلدي,مذكر))),مكمل جملة(شبه جملة(حرف جر(على),تركيب إضافي(نكرات متعددة(نكرة(تحسين,مذكر),نكرة(خواص,مؤنث)),تركيب وصفي(معرفة(التربة,مؤنث),معرفة(الطبيعية,مؤنث)))))))]
3-5	[جملة فعلية(فعل(يزيد,متعدي),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(من),تركيب إضافي(نكرة(قدرة,مؤنث),تركيب وصفي(معرفة(الأراضي,مؤنث),معرفة(الجديدة,مؤنث))))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(على),معرفة(الحفاظ,مذكر))),مكمل جملة(شبه جملة(حرف جر(على),تراكيب إضافية معطوفة(تركيب إضافي(نكرة(مياه,مؤنث),معرفة(الري,مذكر)),حرف وصل(و),تركيب وصفي(معرفة(العناصر,مؤنث),معرفة(الغذائية,مؤنث)))))))]
3-6	[جملة فعلية(فعل(يشترط,متعدي),مكمل جملة(شبه جملة(حرف جر(في),تركيب وصفي(معرفة(السماد,مذكر),معرفة(البلدي,مذكر)))),مفعول به(جملة إن و أخواتها(حرف إن(إن),خبر(جملة كان و أخواتها(فعل(يكون,متعدي),خبر(خبر و متعلق به(شبه جملة(حرف جر(من),تركيب وصفي(نكرة(مصدر,مذكر),نكرة(موثوق,مذكر))),مكمل جملة(شبه جملة(حرف جر(ب),ضمير_متصل(ه)))))))))]
3-7	[جملة كان و أخواتها(فعل(يكون,متعدي),خبر(خبر و متعلق به(نكرة(خاليا,مذكر),مكمل جملة(شبه جملة(حرف جر(من),تراكيب إضافية معطوفة(تركيب إضافي(نكرة(مسببات,مؤنث),معرفة(الأمراض,مؤنث)),حرف وصل(و),تراكيب إضافية معطوفة(تركيب إضافي(نكرة(يرقات,مؤنث),معرفة(الحشرات,مؤنث)),حرف

	وصل(و),معرفة_علم(النيماتود,مذكر)))))))]
3-8	[جملة فعلية(فعل(يضاف,متعدي),مفعول به(تركيب وصفي(معرفة(السماد,مذكر),معرفة(البلدي,مذكر))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(نكرة(معدل,مذكر),تركيب إضافي(وحدة رقمية(20),نكرات متعددة(نكرة(متر,مذكر),نكرة(مكعب,مذكر))))),مكمل جملة(شبه جملة(حرف جر(لل),معرفة(فدان,مذكر))))]
3-9	[جملة فعلية(فعل(تنثر,متعدي),مكمل جملة(شبه جملة(حرف جر(على),معرفة(الحقل,مذكر))))]
3-10	[جملة فعلية(فعل(تخلط,لازم),مكمل جملة مركب(مكمل جملة(نكرة(جيدا,مذكر)),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(بال),معرفة(تربة,مؤنث))),مكمل جملة(شبه جملة(حرف جر(مع),معرفة(الخدمة,مؤنث))))))]
3-11	[جملة فعلية(فعل يسبقه حرف(لا,ينصح,متعدي),مكمل جملة(جملة اعتراضية(كلمة عارضة(أبدا)),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),تراكيب إضافية معطوفة(تركيب إضافي(نكرة(نقل,مذكر),معرفة(الأتربة,مذكر)),حرف وصل(و),تركيب إضافي(نكرة(نواتج,مؤنث),معرفة(التطهير,مذكر))))),مكمل جملة(شبه جملة(حرف جر(إلي),معرفة(الحقل,مذكر)))))]
3-12	[جملة فعلية(فعل و ضمير متصل(سبب,متعدي,ضمير_متصل(ه)),مكمل جملة(شبه جملة(حرف جر(من),تراكيب إضافية معطوفة(تركيب إضافي(نكرة(انتشار,مذكر),معرفة(الحشائش,مؤنث)),حرف وصل(و),معرفة(الأمراض,مؤنث),حرف وصل(و),تركيب إضافي(نكرات متعددة(نكرة(زيادة,مؤنث),نكرة(ملوحة,مؤنث)),معرفة(التربة,مؤنث)))))]
3-13	[جملة فعلية(فعل(يستخدم,متعدي),فاعل(تركيب إضافي(نكرات متعددة(نكرة(سماد,مذكر),نكرات متعددة(نكرة(سوبر,مذكر),نكرة(فوسفات,مذكر))),تركيب وصفي(معرفة(الجير,مذكر),معرفة(الأحادي,مذكر))))),جملة فعلية(فعل(يستخدم,متعدي),مفعول به(تركيب إضافي(نكرات متعددة(نكرة(سماد,مذكر),نكرات متعددة(نكرة(سوبر,مذكر),نكرة(فوسفات,مذكر))),تركيب وصفي(معرفة(الجير,مذكر),معرفة(الأحادي,مذكر)))))]
3-14	[جملة فعلية(فعل(يضاف,متعدي),مكمل جملة معطوفة(مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(نكرة(معدل,مذكر),تركيب إضافي(وحدة رقمية(150),نكرات متعددة(نكرة(كيلو,مذكر),نكرة(جرام,مذكر))))),مكمل جملة(شبه جملة(حرف جر(في),تركيب وصفي(معرفة(الأراضي,مؤنث),معرفة(الجيرية,مؤنث))))),حرف وصل(و),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(نكرة(معدل,مذكر),تركيب إضافي(وحدة رقمية(200),نكرات متعددة(نكرة(كيلو,مذكر),نكرة(جرام,مذكر))))),مكمل جملة(شبه جملة(حرف جر(في),تركيب وصفي(معرفة(الأراضي,مؤنث),معرفة(الرملية,مؤنث)))))))]
3-15	[جملة فعلية(فعل(يضاف,متعدي),مفعول به(تركيب إضافي(وحدة رقمية(50),نكرات متعددة(نكرة(كيلو,مذكر),نكرة(جرام,مذكر)))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(من),تركيب إضافي(نكرات متعددة(نكرة(سماد,مذكر),نكرات متعددة(نكرة(سلفات,مؤنث),نكرة(بوتاسوم,مذكر))))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(لل),معرفة(فدان,مذكر))),مكمل جملة مركب(مكمل جملة(تراكيب إضافية معطوفة(تركيب إضافي(ظرف زمان(قبل),معرفة(الزراعة,مؤنث)),حرف وصل(و),تركيب إضافي(ظرف زمان(أثناء),معرفة(الخدمة,مؤنث)))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(مع),معرفة(التقليب,مذكر))),مكمل جملة مركب(مكمل جملة(نكرة(جيدا,مذكر)),مكمل جملة(شبه جملة(حرف جر(في),معرفة(التربة,مؤنث)))))))]
3-16	[جملة فعلية(فعل(يمكن,متعدي),فاعل(تركيب إضافي(نكرة(استعمال,مذكر),تركيب وصفي(معرفة(المخصبات,مؤنث),معرفة(الحيوية,مؤنث))))),جملة فعلية(فعل(يمكن,متعدي),مفعول به(تركيب إضافي(نكرة(استعمال,مذكر),تركيب وصفي(معرفة(المخصبات,مؤنث),معرفة(الحيوية,مؤنث)))))]
3-17	[جملة فعلية(فعل(يضاف,متعدي),مفعول به(تركيب

	إضافي(نكرة(مخصب,مذكر),معرفة(الفوسفور,مذكر))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(نكرة(واقع,مذكر),تركيب إضافي(وحدة رقمية(2-3),نكرة(أكياس,مؤنث))))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ال),معرفة(فدان,مذكر))),مكمل جملة(تركيب إضافي(ظرف مكان(عند),معرفة(الزراعة,مؤنث))))))]
3-18	[جملة فعلية(فعل(يضاف,متعدي),مفعول به(تركيب إضافي(نكرة(مخصب,مذكر),معرفة(الفوسفور,مذكر))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(نكرة(واقع,مذكر),تركيب إضافي(وحدة رقمية(2-3),نكرة(أكياس,مؤنث))))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ال),معرفة(فدان,مذكر))),مكمل جملة(تركيب إضافي(ظرف مكان(عند),معرفة(الزراعة,مؤنث))))))]
3-19	[جملة فعلية(فعل(يضاف,متعدي),مفعول به(تركيب وصفي(معرفة(السماد,مذكر),معرفة(النتروجيني,مذكر))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(نكرة(معدل,مذكر),تركيب إضافي(وحدة رقمية(100,حرف جر(إلي),120),نكرات متعددة(نكرة(كيلو,مذكر),نكرات متعددة(نكرة(جرام,مذكر),نكرة(نتروجين,مذكر)))))),مكمل جملة(شبه جملة(حرف جر(ال),معرفة(فدان,مذكر)))))]
3-20	[جملة أسمية(مبتدأ(تركيب بدل(اسم اشارة(هذا),معرفة(المعدل,مذكر))),خبر(جملة فعلية(فعل(يعادل,متعدي),مفعول به(تركيب إضافي(وحدة رقمية(300-360),نكرات متعددة(نكرة(كيلو,مذكر),نكرات متعددة(نكرة(جرام,مذكر),نكرات متعددة(نكرة(سماد,مذكر),نكرات متعددة(نكرة(نترات,مؤنث),نكرة(نشادر,مؤنث))))))),مكمل جملة(تركيب إضافي(نسبة مئوية(335%),نكرة(آزوت,مذكر)))))]

Parse Tree structure	No.
[جملة أسمية(مبتدأ(ضمير_منفصل(هما)),خبر(خبر و متعلق به(تركيب إضافي(نكرة(نوعا,مذكر),تركيب وصفي(معرفة(السماد,مذكر),معرفة(الموصي,مذكر))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),ضمير_متصل(هما))),مكمل جملة(شبه جملة(حرف جر(في),تركيب وصفي(معرفة(الأراضي,مؤنث),معرفة(الجديدة,مؤنث)))))))]	4-1
[جملة فعلية(فعل يسبقه حرف(لا,يوصى,متعدي),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(نكرات متعددة(نكرة(استعمال,مذكر),نكرة(سماد,مذكر)),معرفة(اليوريا,مؤنث)))),مكمل جملة(شبه جملة(حرف جر(في),تركيب وصفي(معرفة(الأراضي,مؤنث),معرفة(الجديدة,مؤنث))))))]	4-2
[جملة فعلية(فعل(تتم,لازم),فاعل(تركيب إضافي(نكرة(إضافة,مؤنث),تركيب وصفي(معرفة(السماد,مذكر),معرفة(الآزوتي,مذكر)))),مكمل جملة(شبه جملة(حرف جر(في),تركيب وصفي(معرفة(الأراضي,مؤنث),معرفة(الجيرية,مؤنث))))]	4-3
[جملة فعلية(فعل(تروى,متعدي),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(نكرة(نظام,مذكر),معرفة(الغمر,مذكر)))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(على),تركيب إضافي(نكرات متعددة(نكرة(ثلاث,مذكر),نكرة(دفعات,مؤنث))))),مكمل جملة(تراكيب إضافية معطوفة(تركيب إضافي(ظرف مكان(عند),معرفة(الزراعة,مؤنث)),حرف وصل(و),تراكيب إضافية معطوفة(تركيب إضافي(ظرف مكان(عند),تركيب إضافي(نكرة(رية,مؤنث),معرفة(المحاياة,مؤنث))),حرف وصل(و),تركيب إضافي(ظرف زمان(قبل),تركيب إضافي(نكرات متعددة(نكرة(مرحلة,مؤنث),نكرة(طرد,مذكر)),معرفة(السنابل,مؤنث)))))))))]	4-4
[جملة فعلية يتصدرها مكمل جملة مقدم(مكمل جملة(شبه جملة(حرف جر(في),تركيب وصفي(معرفة(الأراضي,مؤنث),معرفة(الرملية,مؤنث)))),جملة فعلية(فعل(يقسم,متعدي),مفعول	4-5

	به(تركيب وصفي(معرفة(السماد,مذكر),معرفة(النتروجيني,مذكر))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(على),تركيب إضافي(وحدة رقمية(5-6),نكرة(دفعات,مؤنث))),مكمل جملة(نكرة(متساوية,مؤنث)))))]
4-6	[جملة فعلية(فعل(تروى,متعدي),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(نكرة(نظام,مذكر),معرفة(الري,مذكر))),مكمل جملة(شبه جملة(حرف جر(بال),معرفة(رش,مذكر)))))]
4-7	[جملة فعلية(فعل(تضاف,متعدي),مفعول به(تركيب إضافي(ظرف زمان(قبل),معرفة(الري,مذكر))),مكمل جملة مركب(مكمل جملة(نكرة(بداية,مؤنث)),مكمل جملة(شبه جملة(حرف جر(من),معرفة(الزراعة,مؤنث))))),جملة فعلية(فعل(تضاف,متعدي),مكمل جملة(تركيب إضافي(ظرف زمان(قبل),معرفة(الري,مذكر))),مفعول به(نكرة(بداية,مؤنث)),مكمل جملة(شبه جملة(حرف جر(من),معرفة(الزراعة,مؤنث))))]
4-8	[جملة فعلية(فعل(يفضل,متعدي),مفعول به(تركيب إضافي(نكرة(إضافة,مؤنث),معرفة(السماد,مذكر))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(مع),تركيب إضافي(نكرة(مياه,مؤنث),معرفة(الري,مذكر)))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(نكرة(استخدام,مذكر),معرفة(السمادة,مؤنث)))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(لل),معرفة(حصول,مذكر))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(على),نكرة(نتائج,مؤنث))),مكمل جملة(نكرة(أفضل,مذكر)))))))]
4-9	[جملة فعلية(فعل(يمكن,متعدي),مكمل جملة(شبه جملة(حرف جر(لل),معرفة(مزارع,مذكر))),مفعول به(جملة إن و أخواتها(حرف إن (إن),خبر(جملة فعلية(فعل(يتعرف,لازم),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(على),تركيب إضافي(نكرات متعددة(نكرة(مدى,مذكر),نكرات متعددة(نكرة(احتياج,مذكر),نكرة(نباتات,مؤنث))),معرفة(القمح,مذكر)))),مكمل جملة(شبه جملة(حرف جر(إلي),تركيب وصفي(معرفة(السماد,مذكر),معرفة(النتروجيني,مذكر)))))))))]
4-10	[جملة أسمية(مبتدأ(اسم اشارة(ذلك)),خبر(شبه جملة(حرف جر(عن),تركيب إضافي(نكرات متعددة(نكرة(طريق,مذكر),نكرات متعددة(نكرة(ملاحظة,مؤنث),نكرة(لون,مذكر))),معرفة(النباتات,مؤنث)))))]
4-11	[جملة فعلية يتصدرها مكمل جملة مقدم(مكمل جملة(شبه جملة(حرف جر(من),معرفة(المفروض,مذكر))), جملة إن و أخواتها(حرف إن (إن),خبر(جملة كان و أخواتها(فعل(تكون,متعدي),جملة أسمية(مبتدأ(معرفة(النباتات,مؤنث)),خبر(تركيب إضافي(نكرة(خضراء,مؤنث),معرفة(اللون,مذكر)))))))]
4-12	[جملة فعلية(فعل(يميل,لازم),فاعل(معرفة(اللون,مذكر)),مكمل جملة مركب(مكمل جملة(نكرة(قليلا,مذكر)),مكمل جملة(شبه جملة(حرف جر(إلي),معرفة(الزرقة,مؤنث)))))]
4-13	[جملة إن و أخواتها(حرف إن (إن),خبر(جملة كان و أخواتها(فعل(يكون,متعدي),خبر(خبر و متعلق به(نكرة(متجانسا,مذكر),مكمل جملة(شبه جملة(حرف جر(في),معرفة(الحقل,مذكر)))))))]
4-14	[جملة أسمية(مبتدأ(معرفة(النباتات,مؤنث)),خبر(خبر يسبقه مكمل جملة(مكمل جملة(تركيب إضافي(نكرة(ذات,مؤنث),تركيب وصفي(معرفة(اللون,مذكر),تركيب وصفي(معرفة(الأخضر,مذكر),معرفة(الشاحب,مذكر))))),جملة فعلية(فعل(فتدل,لازم),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(على),تركيب إضافي(نكرة(احتياج,مذكر),ضمير_متصل(ها)))),مكمل جملة(شبه جملة(حرف جر(لل),تركيب وصفي(معرفة(سماد,مذكر),معرفة(النتروجيني,مذكر)))))))),جملة فعلية يتصدرها مكمل جملة مقدم(مكمل جملة مركب(مكمل جملة(معرفة(النباتات,مؤنث)),مكمل جملة(تركيب إضافي(نكرة(ذات,مؤنث),تركيب وصفي(معرفة(اللون,مذكر),تركيب وصفي(معرفة(الأخضر,مذكر),معرفة(الشاحب,مذكر)))))),جملة فعلية(فعل(فتدل,لازم),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(على),تركيب إضافي(نكرة(احتياج,مذكر),ضمير_متصل(ها)))),مكمل جملة(شبه جملة(حرف جر(لل),تركيب وصفي(معرفة(سماد,مذكر),معرفة(النتروجيني,مذكر)))))))]
4-15	[جملة فعلية(فعل(تميل,لازم),مكمل جملة(شبه جملة(حرف

جر(إلي),معرفة(الصفرة,مؤنث)))),فاعل(تركيب إضافي(نكرة(ضعيفة,مؤنث),معرفة(النمو,مذكر)))))]	
[جملة فعلية يتصدرها مكمل جملة مقدم(مكمل جملة معطوفة(مكمل جملة مركب(مكمل جملة(نكرة(نظرا,مذكر)),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ل),تركيب إضافي(نكرات متعددة(نكرة(قلة,مؤنث),نكرة(احتفاظ,مذكر)),تركيب وصفي(معرفة(الأراضي,مؤنث),معرفة(الجديدة,مؤنث))))),مكمل جملة(نكرة(عامة,مؤنث)))),حرف وصل(و),مكمل جملة مركب(مكمل جملة(تركيب وصفي(معرفة(الأراضي,مؤنث),معرفة(الرملية,مؤنث))),مكمل جملة مركب(مكمل جملة(نكرة(خاصة,مؤنث)),مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(نكرة(مياه,مؤنث),معرفة(الري,مذكر)))))), جملة إن و أخواتها(فعل(فإن,متعدي),جملة أسمية(مبتدأ(تركيب بدل(اسم اشارة(تلك),معرفة(الأراضي,مؤنث))),خبر(جملة فعلية(فعل(تحتاج,لازم),مكمل جملة(شبه جملة(حرف جر(إلي),تركيب وصفي(نكرات متعددة(نكرة(فترات,مؤنث),نكرة(ري,مذكر)),نكرة(متقاربة,مؤنث)))))))))]	4-16
[جملة فعلية(فعل(يمكن,متعدي),مفعول به(تركيب إضافي(نكرة(توفير,مذكر),تركيب وصفي(معرفة(الرطوبة,مؤنث),تركيب وصفي(معرفة(الأرضية,مؤنث),معرفة(اللازمة,مؤنث))))),مكمل جملة(شبه جملة(حرف جر(ل),تركيب إضافي(نكرات متعددة(نكرة(نمو,مذكر),نكرة(نباتات,مؤنث)),معرفة(القمح,مذكر)))))]	4-17
[جملة فعلية يتصدرها مكمل جملة مقدم(مكمل جملة(شبه جملة(حرف جر(في),تركيب وصفي(معرفة(الأراضي,مؤنث),معرفة(الرملية,مؤنث)))),جملة فعلية(فعل(يراعى,متعدي),مفعول به(تركيب إضافي(نكرات متعددة(نكرة(إعطاء,مذكر),نكرة(رية,مؤنث))),مكمل جملة مركب(مكمل جملة(تركيب إضافي(ظرف زمان(بعد),تركيب إضافي(ظرف زمان(حوالي),نكرة(يوم,مذكر)))),مكمل جملة مركب(مكمل جملة(نكرة(واحد,مذكر)),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(من),تركيب إضافي(نكرة(رية,مؤنث),معرفة(الزراعة,مؤنث)))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ل),تركيب إضافي(نكرات متعددة(نكرة(ضمان,مذكر),نكرة(توفير,مذكر)),تركيب وصفي(معرفة(الرطوبة,مؤنث),معرفة(الكافية,مؤنث))))),مكمل جملة(شبه جملة(حرف جر(لل),معرفة(إنبات,مذكر)))))))]	4-18
[جملة فعلية(فعل(تتكرر,متعدي),مفعول به(تركيب وصفي(معرفة(الريات,مؤنث),معرفة(اليومية,مؤنث))),مكمل جملة مركب(مكمل جملة(تركيب إضافي(ظرف زمان(حتى),تركيب إضافي(نكرة(ظهور,مذكر),معرفة(البادرات,مؤنث)))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(مع),تركيب إضافي(نكرة(ضرورة,مؤنث),معرفة(الاهتمام,مذكر)))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(نكرات متعددة(نكرة(عدم,مذكر),نكرة(تعطيش,مذكر)),معرفة(النباتات,مؤنث)))),مكمل جملة(تراكيب إضافية معطوفة(تركيب إضافي(نكرات متعددة(نكرة(طول,مذكر),نكرة(فترات,مؤنث)),معرفة(النمو,مذكر)),حرف وصل(و),تركيب إضافي(ظرف زمان(حتى),معرفة(الحصاد,مذكر)))))))),جملة فعلية(فعل(تتكرر,متعدي),فاعل(تركيب وصفي(معرفة(الريات,مؤنث),معرفة(اليومية,مؤنث))),مفعول به(تركيب إضافي(ظرف زمان(حتى),تركيب إضافي(نكرة(ظهور,مذكر),معرفة(البادرات,مؤنث)))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(مع),تركيب إضافي(نكرة(ضرورة,مؤنث),معرفة(الاهتمام,مذكر)))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(نكرات متعددة(نكرة(عدم,مذكر),نكرة(تعطيش,مذكر)),معرفة(النباتات,مؤنث)))),مكمل جملة(تراكيب إضافية معطوفة(تركيب إضافي(نكرات متعددة(نكرة(طول,مذكر),نكرة(فترات,مؤنث)),معرفة(النمو,مذكر)),حرف وصل(و),تركيب إضافي(ظرف زمان(حتى),معرفة(الحصاد,مذكر)))))))]	4-19
[جملة فعلية(فعل(يروى,متعدي),مفعول به(معرفة(القمح,مذكر)),مكمل جملة مركب(مكمل جملة(شبه	4-20

جملة(حرف جر(في),تركيب وصفي(معرفة(الأراضي,مؤنث),معرفة(الرملية,مؤنث))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(على),نكرة(فترات,مؤنث))),مكمل جملة(شبه جملة(حرف جر(من),تركيب إضافي(وحدة رقمية(5-10),نكرة(أيام,مؤنث)))))]	
[جملة فعلية يتصدرها مكمل جملة مقدم(مكمل جملة(شبه جملة(حرف جر(في),تركيب وصفي(معرفة(الأراضي,مؤنث),معرفة(الجيرية,مؤنث))),جملة فعلية(فعل(يحتاج,لازم),فاعل(معرفة(القمح,مذكر)),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(إلي),تركيب إضافي(ظرف زمان(حوالي),تركيب إضافي(وحدة رقمية(7),نكرة(ريات,مؤنث))))),مكمل جملة مركب(مكمل جملة(تركيب إضافي(نكرة(خلال,حياد),معرفة(الموسم,مذكر))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(مع),تركيب إضافي(نكرات متعددة(نكرة(عدم,مذكر),نكرات متعددة(نكرة(تأخير,مذكر),نكرة(رية,مؤنث))),معرفة(المحاياة,مؤنث))),مكمل جملة(شبه جملة(حرف جر(عن),تركيب إضافي(وحدة رقمية(20-225),نكرة(يوما,مذكر)))))))]	4-21
[جملة فعلية(فعل(تروى,متعدي),مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(نكرة(نظام,مذكر),معرفة(الغمر,مذكر)))))]	4-22
[جملة فعلية(فعل(يوالي,متعدي),مفعول به(معرفة(الري,مذكر)),مكمل جملة مركب(مكمل جملة(تركيب إضافي(ظرف زمان(بعد),اسم اشارة(ذلك))),مكمل جملة(تركيب إضافي(ظرف زمان(كل),تركيب إضافي(وحدة رقمية(15-20),نكرة(يوما,مذكر)))))),جملة فعلية(فعل(يوالي,متعدي),فاعل(معرفة(الري,مذكر)),مفعول به(تركيب إضافي(ظرف زمان(بعد),اسم اشارة(ذلك))),مكمل جملة(تركيب إضافي(ظرف زمان(كل),تركيب إضافي(وحدة رقمية(15-20),نكرة(يوما,مذكر))))),جملة فعلية(فعل(يوالي,متعدي),فاعل(معرفة(الري,مذكر)),مكمل جملة(تركيب إضافي(ظرف زمان(بعد),اسم اشارة(ذلك))),مفعول به(تركيب إضافي(ظرف زمان(كل),تركيب إضافي(وحدة رقمية(15-20),نكرة(يوما,مذكر)))))]	4-23
[جملة فعلية(فعل(يمكن,متعدي),مكمل جملة(شبه جملة(حرف جر(لل),معرفة(مزارع,مذكر))),مفعول به(جملة إن و أخواتها(حرف إن (إن),خبر(جملة فعلية(فعل(يدرك,متعدي),مفعول به(تركيب إضافي(نكرات متعددة(نكرة(مدى,مذكر),نكرة(احتياج,مذكر)),معرفة(الحقل,مذكر))),مكمل جملة معطوفة(مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(إلي),معرفة(الري,مذكر))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(عن),تراكيب إضافية معطوفة(تركيب إضافي(نكرات متعددة(نكرة(طريق,مذكر),نكرة(جفاف,مذكر)),معرفة(التربة,مؤنث)),حرف وصل(و),تركيب إضافي(نكرة(ظهور,مذكر),معرفة(الشقوق,مؤنث))))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(على),تركيب إضافي(نكرة(سطح,مذكر),معرفة(التربة,مؤنث)))),مكمل جملة(شبه جملة(حرف جر(في),تركيب وصفي(معرفة(الأراضي,مؤنث),معرفة(الجيرية,مؤنث))))),حرف وصل(و),مكمل جملة مركب(مكمل جملة(تركيب إضافي(نكرة(جفاف,مذكر),تركيب وصفي(معرفة(الطبقة,مؤنث),معرفة(السطحية,مؤنث))),مكمل جملة(شبه جملة(حرف جر(ب),ضمير_متصل(ها)))))))))]	4-24
[جملة أسمية(مبتدأ(تراكيب إضافية معطوفة(تركيب إضافي(نكرات متعددة(نكرة(التفاف,مذكر),نكرة(أوراق,مؤنث)),معرفة(النباتات,مؤنث)),حرف وصل(و),تركيب إضافي(نكرات متعددة(نكرة(ضعف,مذكر),نكرة(نمو,مذكر)),ضمير_متصل(ها)))),خبر(خبر و متعلق به(شبه جملة(حرف جر(ك),نكرة(مظهر,مذكر)),مكمل جملة(شبه جملة(حرف جر(من),تركيب إضافي(نكرة(مظاهر,مؤنث),معرفة(العطش,مذكر)))))))]	4-25

Parse Tree structure	No.
[جملة أسمية(خبر مقدم(شبه جملة(حرف جر(من),معرفة(الضروري,مذكر))),مبتدأ(معرفة(العمل,مذكر)),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(على),تركيب إضافي(نكرة(مقاومة,مؤنث),معرفة(الحشائش,مؤنث)))),مكمل جملة(شبه جملة(حرف جر(في),تركيب إضافي(نكرة(حقول,مؤنث),معرفة(القمح,مذكر)))))]	5-1

5-2	[جملة فعلية(فعل يسبقه حرف(لا,تنافس,متعدي),مفعول به(معرفة(المحصول,مذكر)),مكمل جملة(شبه جملة(حرف جر(في),تركيب إضافي(تركيب إضافي(نكرة(احتياج,مؤنث),ضمير_متصل(ه)),معرفة(البيئية,مؤنث)))))]
5-3	[جملة فعلية(فعل(تتم,متعدي),مفعول به(معرفة(المقاومة,مؤنث)),مكمل جملة معطوفة(مكمل جملة مركب(مكمل جملة(نكرة(سواء,مذكر)),مكمل جملة(نكرة(يدوية,مؤنث))),حرف وصل(أو),مكمل جملة معطوفة(مكمل جملة(نكرة(مكانيكية,مؤنث)),حرف وصل(أو),مكمل جملة(نكرة(كيماوية,مؤنث)))))]
5-4	[جملة فعلية(فعل(لزم,متعدي),فاعل(معرفة(الأمر,مذكر))),جملة فعلية(فعل(لزم,متعدي),مفعول به(معرفة(الأمر,مذكر)))]
5-5	[جملة فعلية(فعل(يمكن,متعدي),مفعول به(تركيب إضافي(نكرة(استعمال,مذكر),تركيب وصفي(معرفة(المكافحة,مؤنث),معرفة(الكيماوية,مؤنث)))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(لل),معرفة(حشائش,مؤنث))),مكمل جملة مركب(مكمل جملة(تركيب إضافي(ظرف مكان(عند),تركيب إضافي(نكرة(انتشار,مذكر),معرفة(الحشائش,مؤنث)))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),نكرة(شدة,مؤنث))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(مع),نكرة(ضبط,مذكر))),مكمل جملة مركب(مكمل جملة(تركيب إضافي(نكرة(معدل,مذكر),معرفة(الرش,مذكر))),مكمل جملة(شبه جملة(حرف جر(ل),تركيب إضافي(نكرة(ضمان,مذكر),معرفة(الانتظام,مذكر))))))))),جملة فعلية(فعل(يمكن,متعدي),فاعل(تركيب إضافي(نكرة(استعمال,مذكر),تركيب وصفي(معرفة(المكافحة,مؤنث),معرفة(الكيماوية,مؤنث)))),مكمل جملة(شبه جملة(حرف جر(لل),معرفة(حشائش,مؤنث))),مفعول به(تركيب إضافي(ظرف مكان(عند),تركيب إضافي(نكرة(انتشار,مذكر),معرفة(الحشائش,مؤنث)))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),نكرة(شدة,مؤنث))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(مع),نكرة(ضبط,مذكر))),مكمل جملة مركب(مكمل جملة(تركيب إضافي(نكرة(معدل,مذكر),معرفة(الرش,مذكر))),مكمل جملة(شبه جملة(حرف جر(ل),تركيب إضافي(نكرة(ضمان,مذكر),معرفة(الانتظام,مذكر)))))))))]
5-6	[جملة إن و أخواتها(حرف إن (إن),خبر(جملة فعلية(فعل(يجرى,متعدي),مفعول به(معرفة(الرش,مذكر)),مكمل جملة(تركيب إضافي(ظرف زمان(بعد),تركيب إضافي(نكرة(تطاير,مذكر),معرفة(الندى,مذكر)))))))]
5-7	UnParseable
5-8	[جملة كان و أخواتها(فعل(يكون,متعدي),جملة أسمية(مبتدأ(تركيب إضافي(نكرة(تأثير,مذكر),معرفة(الرش,مذكر))),خبر(نكرة(فعال,مذكر))))]
5-9	[جملة أسمية(مبتدأ(تركيب إضافي(نكرة(سينال,مذكر),تركيب إضافي(نسبة مئوية(10%),معرفة_علم_لاتيني(SC,حياد)))),خبر(خبر و متعلق به(شبه جملة(حرف جر(ب),تركيب إضافي(نكرة(معدل,مذكر),تركيب إضافي(وحدة رقمية(40),نكرة(سم3/ف,مذكر)))),مكمل جملة مركب(مكمل جملة(نكرة(رشا,مذكر)),مكمل جملة مركب(مكمل جملة(تركيب إضافي(ظرف زمان(قبل),تركيب إضافي(نكرة(رية,مؤنث),معرفة(المحاياة,مؤنث)))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),تركيب وصفي(نكرة(يوم,مذكر),نكرة(واحد,مذكر)))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(مع),تركيب إضافي(وحدة رقمية(200),نكرات متعددة(نكرة(لتر,مذكر),نكرة(ماء,مذكر)))),مكمل جملة(شبه جملة(حرف جر(لل),معرفة(فدان,مذكر)))))))))]
5-10	[جملة مركبة(جملة أسمية(مبتدأ(تركيب إضافي(نكرة(سنكور,مذكر),تركيب إضافي(نسبة مئوية(70%),معرفة_علم_لاتيني(WP,حياد)))),خبر(خبر و متعلق به(شبه جملة(حرف جر(ب),تركيب إضافي(نكرة(معدل,مذكر),تركيب إضافي(وحدة رقمية(60),نكرة(جم/فدان,مذكر)))),مكمل جملة مركب(مكمل جملة(نكرة(رشا,مذكر)),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(في),تركيب إضافي(نكرة(طور,مذكر),تركيب إضافي(وحدة

	رقمية(2-4),نكرة(أوراق,مؤنث)))))),مكمل جملة(شبه جملة(حرف جر(لل),معرفة(قمح,مذكر)))))))),رابط جمل(حرف وصل(و)),جملة أسمية(مبتدأ(تركيب بدل(اسم اشارة(هذان),معرفة(المبيدان,مذكر))),خبر(جملة فعلية(فعل(يستخدمان,متعدي),مفعول به(تركيب إضافي(نكرة(ضد,مذكر),معرفة(الحشائش,مؤنث))),مكمل جملة مركب(مكمل جملة(تركيب إضافي(نكرة(عريضة,مؤنث),معرفة(الأوراق,مؤنث))),مكمل جملة(نكرة(فقط,حياد)))))))]
5-11	[جملة أسمية(مبتدأ(تركيب إضافي(نكرة(أريلون,مذكر),تركيب إضافي(نسبة مئوية(50%),معرفة_علم_لاتيني(FL,حياد)))),خبر(خبر يسبقه مكمل جملة(مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(نكرة(معدل,مذكر),تركيب إضافي(وحدة رقمية(125),نكرة(لتر/فدان,مذكر))))),مكمل جملة(شبه جملة(حرف جر(مع),تركيب إضافي(وحدة رقمية(200),نكرات متعددة(نكرة(لتر,مذكر),نكرة(ماء,مذكر))))))),جملة فعلية(فعل(يرش,متعدي),مفعول به(تركيب إضافي(ظرف مكان(عند),تركيب إضافي(نكرة(عمر,مذكر),تركيب إضافي(وحدة رقمية(3-4),نكرة(ورقات,مؤنث))))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ل),تركيب إضافي(نكرة(نبات,مذكر),معرفة(القمح,مذكر)))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(مع),تركيب إضافي(نكرة(رج,مذكر),معرفة(العبوة,مؤنث)))),مكمل جملة مركب(مكمل جملة(نكرة(جيدا,مذكر)),مكمل جملة(تركيب إضافي(ظرف زمان(قبل),معرفة(الاستعمال,مذكر)))))))),جملة فعلية يتصدرها مكمل جملة مقدم(مكمل جملة مركب(مكمل جملة(تركيب إضافي(نكرة(أريلون,مذكر),تركيب إضافي(نسبة مئوية(50%),معرفة_علم_لاتيني(FL,حياد)))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(نكرة(معدل,مذكر),تركيب إضافي(وحدة رقمية(125),نكرة(لتر/فدان,مذكر))))),مكمل جملة(شبه جملة(حرف جر(مع),تركيب إضافي(وحدة رقمية(200),نكرات متعددة(نكرة(لتر,مذكر),نكرة(ماء,مذكر))))))),جملة فعلية(فعل(يرش,متعدي),مكمل جملة مركب(مكمل جملة(تركيب إضافي(ظرف مكان(عند),تركيب إضافي(نكرة(عمر,مذكر),تركيب إضافي(وحدة رقمية(3-4),نكرة(ورقات,مؤنث))))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ل),تركيب إضافي(نكرة(نبات,مذكر),معرفة(القمح,مذكر)))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(مع),تركيب إضافي(نكرة(رج,مذكر),معرفة(العبوة,مؤنث)))),مكمل جملة مركب(مكمل جملة(نكرة(جيدا,مذكر)),مكمل جملة(تركيب إضافي(ظرف زمان(قبل),معرفة(الاستعمال,مذكر)))))))),جملة فعلية يتصدرها مكمل جملة مقدم(مكمل جملة مركب(مكمل جملة(تركيب إضافي(نكرة(أريلون,مذكر),تركيب إضافي(نسبة مئوية(50%),معرفة_علم_لاتيني(FL,حياد)))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(نكرة(معدل,مذكر),تركيب إضافي(وحدة رقمية(125),نكرة(لتر/فدان,مذكر))))),مكمل جملة(شبه جملة(حرف جر(مع),تركيب إضافي(وحدة رقمية(200),نكرات متعددة(نكرة(لتر,مذكر),نكرة(ماء,مذكر))))))),جملة فعلية(فعل(يرش,متعدي),مفعول به(تركيب إضافي(ظرف مكان(عند),تركيب إضافي(نكرة(عمر,مذكر),تركيب إضافي(وحدة رقمية(3-4),نكرة(ورقات,مؤنث))))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ل),تركيب إضافي(نكرة(نبات,مذكر),معرفة(القمح,مذكر)))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(مع),تركيب إضافي(نكرة(رج,مذكر),معرفة(العبوة,مؤنث)))),مكمل جملة مركب(مكمل جملة(نكرة(جيدا,مذكر)),مكمل جملة(تركيب إضافي(ظرف زمان(قبل),معرفة(الاستعمال,مذكر)))))))]
5-12	[جملة أسمية(مبتدأ(معرفة_علم_لاتيني(IPFLOW,حياد)),خبر(خبر يسبقه مكمل جملة(مكمل جملة مركب(مكمل جملة(تركيب إضافي(نسبة مئوية(50%),معرفة_علم_لاتيني(FL,حياد))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(نكرة(معدل,مذكر),تركيب إضافي(وحدة رقمية(125),نكرة(لتر,مذكر))))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(في),تركيب إضافي(وحدة رقمية(200),نكرات متعددة(نكرة(لتر,مذكر),نكرة(ماء,مذكر))))),مكمل جملة(شبه

	جملة(حرف جر(لل),معرفة(فدان,مذكر))))))),جملة فعلية(فعل(يرش,متعدي),مفعول به(تركيب إضافي(ظرف مكان(عند),تركيب إضافي(نكرة(عمر,مذكر),تركيب إضافي(وحدة رقمية(3-4),نكرة(ورقات,مؤنث))))),مكمل جملة(شبه جملة(حرف جر(ل),تركيب إضافي(نكرة(نبات,مذكر),معرفة(القمح,مذكر)))))))),جملة فعلية يتصدرها مكمل جملة مقدم(مكمل جملة مركب(مكمل جملة(معرفة_علم_لاتيني(IPFLOW,حياد)),مكمل جملة مركب(مكمل جملة(تركيب إضافي(نسبة مئوية(50%),معرفة_علم_لاتيني(FL,حياد))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(نكرة(معدل,مذكر),تركيب إضافي(وحدة رقمية(125),نكرة(لتر,مذكر))))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(في),تركيب إضافي(وحدة رقمية(200),نكرات متعددة(نكرة(لتر,مذكر),نكرة(ماء,مذكر))))),مكمل جملة(شبه جملة(حرف جر(لل),معرفة(فدان,مذكر)))))),جملة فعلية(فعل(يرش,متعدي),مكمل جملة مركب(مكمل جملة(تركيب إضافي(ظرف مكان(عند),تركيب إضافي(نكرة(عمر,مذكر),تركيب إضافي(وحدة رقمية(3-4),نكرة(ورقات,مؤنث))))),مكمل جملة(شبه جملة(حرف جر(ل),تركيب إضافي(نكرة(نبات,مذكر),معرفة(القمح,مذكر)))))),جملة فعلية يتصدرها مكمل جملة مقدم(مكمل جملة مركب(مكمل جملة(معرفة_علم_لاتيني(IPFLOW,حياد)),مكمل جملة مركب(مكمل جملة(تركيب إضافي(نسبة مئوية(50%),معرفة_علم_لاتيني(FL,حياد))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(نكرة(معدل,مذكر),تركيب إضافي(وحدة رقمية(125),نكرة(لتر,مذكر))))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(في),تركيب إضافي(وحدة رقمية(200),نكرات متعددة(نكرة(لتر,مذكر),نكرة(ماء,مذكر))))),مكمل جملة(شبه جملة(حرف جر(لل),معرفة(فدان,مذكر)))))),جملة فعلية(فعل(يرش,متعدي),مفعول به(تركيب إضافي(ظرف مكان(عند),تركيب إضافي(نكرة(عمر,مذكر),تركيب إضافي(وحدة رقمية(3-4),نكرة(ورقات,مؤنث))))),مكمل جملة(شبه جملة(حرف جر(ل),تركيب إضافي(نكرة(نبات,مذكر),معرفة(القمح,مذكر)))))))]
5-13	[جملة فعلية(فعل يسبقه حرف(لا يستخدم,متعدي),مفعول به(نكرة(أي,حياد)),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(من),تركيب بدل(اسم اشارة(هذين),معرفة(المبيدين,مذكر)))),مكمل جملة(شبه جملة(حرف جر(في),تراكيب وصفية معطوفة(تركيب وصفي(معرفة(الأراضي,مؤنث),معرفة(الرملية,مؤنث)),حرف وصل(أو),معرفة(الكلسية,مؤنث)))))]
5-14	[جملة فعلية(فعل(يتم,لازم),فاعل(تركيب إضافي(نكرة(استخدام,مذكر),ضمير_متصل(ها))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(في),تركيب وصفي(معرفة(الأراضي,مؤنث),معرفة(الطينية,مؤنث)))),مكمل جملة(نكرة(فقط,حياد))))]
5-15	[جملة فعلية(فعل(يستخدم,متعدي),مفعول به(تركيب إضافي(نكرة(مبيد,مذكر),تركيب إضافي(نكرة(سافيكس,مذكر),تركيب إضافي(نسبة مئوية(20%),معرفة_علم_لاتيني(EC,حياد))))),مكمل جملة معطوفة(مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(نكرة(معدل,مذكر),تركيب إضافي(وحدة رقمية(125),نكرة(لتر,مذكر))))),مكمل جملة(شبه جملة(حرف جر(ل),معرفة(فدان,مذكر)))),حرف وصل(أو),مكمل جملة مركب(مكمل جملة(تركيب إضافي(نكرة(مبيد,مذكر),تركيب إضافي(نكرة(جراسب,مذكر),تركيب إضافي(نسبة مئوية(10%),معرفة_علم_لاتيني(EC,حياد))))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),تركيب وصفي(نكرة(معدل,مذكر),نكرة(واحد,مذكر)))),مكمل جملة مركب(مكمل جملة(نكرة(لتر,مذكر)),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(لل),معرفة(فدان,مذكر))),مكمل جملة(شبه جملة(حرف جر(في),تركيب إضافي(ظرف زمان(حوالي),تركيب إضافي(وحدة رقمية(200),نكرات متعددة(نكرة(لتر,مذكر),نكرة(ماء,مذكر)))))))))]
5-16	[جملة فعلية(فعل(يرش,متعدي),مكمل جملة مركب(مكمل جملة(تركيب إضافي(ظرف مكان(عند),تركيب إضافي(نكرة(عمر,مذكر),تركيب إضافي(وحدة رقمية(4-5),نكرة(ورقات,مؤنث))))),مكمل جملة(شبه جملة(حرف جر(ل),تركيب

	إضافي(نكرة(نبات,مذكر),معرفة(القمح,مذكر))))),جملة فعلية(فعل(يرش,متعدي),مفعول به(تركيب إضافي(ظرف مكان(عند),تركيب إضافي(نكرة(عمر,مذكر),تركيب إضافي(وحدة رقمية(4-5),نكرة(ورقات,مؤنث))))),مكمل جملة(شبه جملة(حرف جر(ل),تركيب إضافي(نكرة(نبات,مذكر),معرفة(القمح,مذكر)))))]
5-17	[جملة أسمية(مبتدأ(تركيب إضافي(نكرة(توبيك,مذكر),تركيب إضافي(نسبة مئوية(24%),معرفة_علم_لاتيني(EC,حياد))),خبر(خبر و متعلق به(شبه جملة(حرف جر(ب),تركيب إضافي(نكرة(معدل,مذكر),تركيب إضافي(وحدة رقمية(100),نكرة(سم3,مذكر))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ل),معرفة(فدان,مذكر))),مكمل جملة مركب(مكمل جملة(نكرة(رشا,مذكر)),مكمل جملة مركب(مكمل جملة(نكرة(خلال,حياد)),مكمل جملة مركب(مكمل جملة(نكرة(شهر,مذكر)),مكمل جملة(تركيب إضافي(ظرف زمان(بعد),معرفة(المحاياة,مؤنث)))))))))]
5-18	[جملة أسمية(مبتدأ(تركيب إضافي(نكرة(اسيرت,مذكر),تركيب إضافي(نسبة مئوية(25%),معرفة_علم_لاتيني(EC,حياد))),خبر(خبر و متعلق به(شبه جملة(حرف جر(ب),تركيب إضافي(نكرة(معدل,مذكر),تركيب إضافي(وحدة رقمية(850),نكرة(سم3,مذكر))),مكمل جملة معطوفة(مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ل),معرفة(فدان,مذكر))),مكمل جملة مركب(مكمل جملة(نكرة(رشا,مذكر)),مكمل جملة مركب(مكمل جملة(تركيب إضافي(ظرف زمان(بعد),تركيب إضافي(وحدة رقمية(30-35),نكرة(يوما,مذكر))),مكمل جملة(شبه جملة(حرف جر(من),معرفة(الزراعة,مؤنث)))))),حرف وصل(أو),مكمل جملة مركب(مكمل جملة(تركيب إضافي(نكرة(بوناسوبر,مذكر),تركيب إضافي(نسبة مئوية(75%),معرفة_علم_لاتيني(EW,حياد))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(نكرة(معدل,مذكر),تركيب إضافي(وحدة رقمية(500),نكرة(سم3,مذكر))))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ل),معرفة(فدان,مذكر))),مكمل جملة مركب(مكمل جملة(نكرة(رشا,مذكر)),مكمل جملة مركب(مكمل جملة(تركيب إضافي(ظرف زمان(بعد),تركيب إضافي(وحدة رقمية(30-35),نكرة(يوما,مذكر))),مكمل جملة(شبه جملة(حرف جر(من),معرفة(الزراعة,مؤنث)))))))))))]

Parse Tree structure	No.
[جملة فعلية يتصدرها مكمل جملة مقدم(مكمل جملة(تركيب إضافي(نكرة(اتباع,مذكر),تركيب وصفي(نكرة(دورة,مؤنث),نكرة(زراعية,مؤنث)))),جملة فعلية(فعل و ضمير متصل(تخلل,متعدي,ضمير_متصل(ها)),فاعل(تركيب إضافي(نكرة(محصول,مذكر),معرفة(البرسيم,مذكر)))))]	6-1
[جملة فعلية(فعل(يساهم,لازم),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(في),تركيب إضافي(نكرات متعددة(نكرة(تقليل,مذكر),نكرات متعددة(نكرة(أعداد,مؤنث),نكرة(نباتات,مؤنث))),معرفة(الزمير,مذكر)))),مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(نكرات متعددة(نكرة(تكرار,مذكر),نكرة(حش,مذكر)),معرفة(البرسيم,مذكر)))))]	6-2
UnParseable	6-3
UnParseable	6-4
[جملة فعلية(فعل(يزيد,متعدي),مفعول به(معرفة(القدرة,مؤنث)),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(على),تراكيب إضافية معطوفة(تركيب إضافي(نكرة(منافسة,مؤنث),معرفة(الحشائش,مؤنث)),حرف وصل(و),معرفة(التفوق,مذكر)))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(إلي),ضمير_متصل(ها))),مكمل جملة(شبه جملة(حرف جر(في),معرفة(النمو,مذكر)))))]	6-5
UnParseable	6-6
UnParseable	6-7

6-8	[جملة فعلية(فعل(تعتبر,متعدي),مفعول به(نكرة(حشرات,مؤنث)),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(من),معرفة(القمح,مذكر))),مكمل جملة(شبه جملة(حرف جر(من),تركيب إضافي(نكرات متعددة(نكرة(أهم,مذكر),نكرة(مجاميع,مؤنث)),تركيب وصفي(معرفة(الحشرات,مؤنث),تركيب وصفي(معرفة(الثاقبة,مؤنث),معرفة(الماصة,مؤنث)))))))]
6-9	[جملة فعلية(فعل(تصيب,متعدي),فاعل(تركيب وصفي(معرفة(المحاصيل,مؤنث),معرفة(النجيلية,مؤنث))),مفعول به(نكرة(عموما,مذكر))),جملة فعلية(فعل(تصيب,متعدي),مفعول به(تركيب وصفي(معرفة(المحاصيل,مؤنث),معرفة(النجيلية,مؤنث))),مكمل جملة(نكرة(عموما,مذكر)))]
6-10	[جملة فعلية(فعل(تتأثر,لازم),فاعل(تركيب وصفي(معرفة(النباتات,مؤنث),معرفة(المصابة,مؤنث))),مكمل جملة(نكرة(نتيجة,مؤنث)))]
6-11	[جملة أسمية(مبتدأ(تركيب وصفي(معرفة(السحب,مذكر),معرفة(المباشر,مذكر))),خبر(خبر و متعلق به(شبه جملة(حرف جر(ل),تركيب وصفي(معرفة(عصير,مذكر),معرفة(الخلوي,مذكر))),مكمل جملة مركب(مكمل جملة(تركيب إضافي(نكرة(نتيجة,مؤنث),معرفة(التغذية,مؤنث))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),تركيب وصفي(نكرة(أعداد,مؤنث),نكرة(كبيرة,مؤنث)))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(من),معرفة(الحشرات,مؤنث))),مكمل جملة(شبه جملة(حرف جر(على),تركيب وصفي(معرفة(الأنسجة,مؤنث),معرفة(النباتية,مؤنث)))))))))]
6-12	[جملة أسمية(مبتدأ(تركيب إضافي(نكرة(تعجيز,مذكر),تركيب وصفي(معرفة(الأوراق,مؤنث),معرفة(المصابة,مؤنث)))),خبر(خبر و متعلق به(نكرة(نتيجة,مؤنث),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ل),تراكيب إضافية معطوفة(تركيب إضافي(نكرة(تجمع,مذكر),تركيب وصفي(معرفة(الإفرازات,مؤنث),معرفة(العسلية,مؤنث))),حرف وصل(و),تراكيب إضافية معطوفة(تركيب إضافي(نكرة(نمو,مذكر),معرفة(الأعفان,مؤنث)),حرف وصل(و),تركيب إضافي(نكرة(قفل,مذكر),تركيب وصفي(معرفة(الثغور,مؤنث),معرفة(التنفسية,مؤنث))))))),مكمل جملة(شبه جملة(حرف جر(ل),معرفة(أوراق,مؤنث)))))]
6-13	[جملة أسمية(مبتدأ(تركيب إضافي(نكرة(تاثر,مذكر),تركيب وصفي(معرفة(الأنسجة,مؤنث),معرفة(النباتية,مؤنث)))),خبر(خبر و متعلق به(نكرة(نتيجة,مؤنث),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ل),تركيب وصفي(معرفة(إفراز,مؤنث),معرفة(اللعابية,مؤنث)))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ل),تركيب إضافي(نكرة(من,مذكر),معرفة(القمح,مذكر)))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(في),تركيب وصفي(نكرات متعددة(نكرة(صورة,مؤنث),نكرة(بقع,مؤنث)),نكرة(حمراء,مؤنث)))),مكمل جملة(تركيب إضافي(نكرة(ذات,مؤنث),تركيب وصفي(نكرة(مراكز,مؤنث),نكرة(سوداء,مؤنث)))))))))]
6-14	UnParseable
6-15	[جملة فعلية(فعل و ضمير متصل(عقب,متعدي,ضمير_متصل(ها)),فاعل(نكرة(تدمير,مذكر)),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ل),تركيب وصفي(معرفة(مادة,مؤنث),معرفة(الخضراء,مؤنث)))),مكمل جملة(شبه جملة(حرف جر(بال),معرفة(أوراق,مؤنث)))))]
6-16	[جملة فعلية(فعل(تموت,لازم),فاعل(تركيب وصفي(معرفة(الأنسجة,مؤنث),معرفة(المصابة,مؤنث))))]
6-17	[جملة فعلية(فعل(تلاحظ,متعدي),مفعول به(تركيب بدل(اسم اشارة(هذه),معرفة(الظاهرة,مؤنث))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(على),تركيب وصفي(معرفة(الأوراق,مؤنث),معرفة(السفلية,مؤنث)))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(بال),معرفة(قرب,مذكر))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(من),تركيب إضافي(نكرة(سطح,مذكر),تراكيب وصفية معطوفة(معرفة(الأرض,مؤنث),حرف وصل(و),تركيب

وصفي(معرفة(المنطقة,مؤنث),معرفة(الوسطى,مؤنث))))))),مكمل جملة(شبه جملة(حرف جر(من),معرفة(النبات,مذكر)))))))]	
UnParseable	6-18
[جملة فعلية(فعل(أثر,لازم),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(بال),معرفة(سلب,مذكر))),مكمل جملة(شبه جملة(حرف جر(على),تركيب إضافي(نكرة(وفرة,مؤنث),تركيب وصفي(معرفة(الأعداء,مؤنث),معرفة(الحيوية,مؤنث)))))))]	6-19
[جملة أسمية(مبتدأ(تركيب وصفي(معرفة(الاكتشاف,مذكر),معرفة(المتأخر,مذكر))),خبر(شبه جملة(حرف جر(ل),معرفة(إصابة,مؤنث)))))]	6-20
[جملة أسمية(مبتدأ(تراكيب إضافية معطوفة(تركيب إضافي(نكرات متعددة(نكرة(تداخل,مذكر),نكرة(مواعيد,مؤنث)),معرفة(الزراعة,مؤنث)),حرف وصل(و),تراكيب إضافية معطوفة(تركيب إضافي(نكرات متعددة(نكرة(صغر,مذكر),نكرة(حجم,مذكر)),معرفة(الحيزات,مؤنث)),حرف وصل(و),تركيب إضافي(نكرة(عدم,مذكر),معرفة(الاعتناء,مذكر)))))),خبر(شبه جملة(حرف جر(ب),تراكيب إضافية معطوفة(تركيب إضافي(نكرات متعددة(نكرة(إزالة,مؤنث),نكرة(بقايا,مؤنث)),معرفة(المحصول,مذكر)),حرف وصل(و),تركيب إضافي(نكرة(تنقية,مؤنث),معرفة(الحشائش,مؤنث))))))]	6-21
[جملة فعلية يتصدرها مكمل جملة مقدم(مكمل جملة معطوفة(مكمل جملة(تركيب وصفي(معرفة(التسميد,مذكر),تركيب وصفي(معرفة(النتروجيني,مذكر),معرفة(الغزير,مذكر)))),حرف وصل(و),مكمل جملة(شبه جملة(حرف جر(في),تركيب وصفي(نكرة(صورة,مؤنث),نكرة(مجزئة,مؤنث))))),جملة فعلية(فعل(يأتي,لازم),مكمل جملة(شبه جملة(حرف جر(في),تراكيب إضافية معطوفة(تركيب إضافي(نكرات متعددة(نكرة(مصلحة,مؤنث),نكرة(انتشار,مذكر))),حرف وصل(و),تركيب إضافي(نكرات متعددة(نكرة(تزايد,مذكر),نكرة(حشرات,مؤنث)),معرفة(المن,مذكر)))))))]	6-22

Parse Tree structure	No.
[جملة أسمية(مبتدأ(تركيب إضافي(نكرة(زراعة,مؤنث),معرفة(الأصناف,مؤنث))),خبر(تركيب إضافي(نكرة(ذات,مؤنث),تركيب وصفي(معرفة(الاحتياجات,مؤنث),تركيب وصفي(معرفة(السمادية,مؤنث),معرفة(العالية,مؤنث))))))]	7-1
[جملة فعلية يتصدرها مكمل جملة مقدم(مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(من),معرفة(الجدير,مذكر))),مكمل جملة(شبه جملة(حرف جر(بال),معرفة(ذكر,مذكر)))), جملة إن و أخواتها(حرف إن (إن),جملة أسمية(مبتدأ(اسم اشارة(هناك)),خبر(خبر يسبقه مكمل جملة(مكمل جملة(تركيب إضافي(ظرف مكان(بعض),تركيب وصفي(معرفة(التوصيات,مؤنث),معرفة(الهامة,مؤنث)))),جملة فعلية(فعل(يجب,متعدي),فاعل(تركيب إضافي(نكرة(اتباع,مذكر),ضمير_متصل(ها)))))))))]	7-2
[جملة مركبة(جملة أسمية(مبتدأ(تركيب إضافي(نكرة(تكبير,مذكر),تركيب وصفي(معرفة(المساحات,مؤنث),معرفة(المنزرعة,مؤنث)))),خبر(شبه جملة(حرف جر(بال),معرفة(قمح,مذكر)))),رابط جمل(حرف وصل(و)),جملة أسمية(مبتدأ(تراكيب إضافية معطوفة(تركيب إضافي(نكرة(تجميع,مذكر),ضمير_متصل(ها)),حرف وصل(و),تركيب إضافي(نكرات متعددة(نكرة(توحيد,مذكر),نكرة(ميعاد,مذكر)),معرفة(الزراعة,مؤنث)))),خبر(شبه جملة(حرف جر(ب),تركيب إضافي(نكرة(قدر,مذكر),معرفة(الإمكان,مذكر)))))]	7-3
[جملة فعلية(فعل(تعمل,لازم),فاعل(تراكيب وصفية معطوفة(تركيب وصفي(معرفة(المساحات,مؤنث),معرفة(الصغيرة,مؤنث)),حرف وصل(و),معرفة(المتبادلة,مؤنث))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(مع),تركيب وصفي(نكرة(محاصيل,مؤنث),نكرة(أخرى,مؤنث)))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(على),تركيب وصفي(نكرات	7-4

متعددة(نكرة(عكس,مذكر),نكرة(طول,مذكر)),نكرة(موجي,مذكر))))),مكمل جملة(تركيب وصفي(نكرة(لوني,مذكر),نكرة(معين,مذكر))))))]	
[جملة فعلية(فعل(تستجيب,لازم),مكمل جملة(شبه جملة(حرف جر(إلي),ضمير_متصل(ه))),فاعل(تركيب إضافي(نكرة(حشرات,مؤنث),معرفة(المن,مذكر))),مكمل جملة(تركيب إضافي(ظرف زمان(أثناء),معرفة(الطيران,مذكر))))]	7-5
[جملة فعلية(فعل(فتهبط,لازم),مكمل جملة(شبه جملة(حرف جر(على),معرفة(القمح,مذكر))))]	7-6
[جملة فعلية يتصدرها مكمل جملة مقدم(مكمل جملة(شبه جملة(حرف جر(من),تركيب وصفي(نكرة(ناحية,مؤنث),نكرة(أخرى,مؤنث)))),جملة فعلية(فعل(يعمل,لازم),فاعل(تركيب إضافي(نكرات متعددة(نكرة(تداخل,مذكر),نكرة(مواعيد,مؤنث)),معرفة(الزراعة,مؤنث))),مكمل جملة معطوفة(مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(على),تركيب إضافي(نكرات متعددة(نكرة(تحرك,مذكر),نكرة(حشرات,مؤنث)),معرفة(المن,مذكر)))),مكمل جملة مركب(مكمل جملة(نكرة(بحثا,مذكر)),مكمل جملة(شبه جملة(حرف جر(عن),تركيب وصفي(معرفة(الأوراق,مؤنث),معرفة(الغضة,مؤنث))))),حرف وصل(و),مكمل جملة(تركيب وصفي(معرفة(العمر,مذكر),معرفة(الأصغر,مذكر))))))]	7-7
[جملة فعلية يتصدرها مكمل جملة مقدم(مكمل جملة مركب(مكمل جملة(نكرة(نتيجة,مؤنث)),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ل),تراكيب إضافية معطوفة(تركيب إضافي(نكرة(تغير,مذكر),تركيب وصفي(معرفة(الظروف,مؤنث),معرفة(البيئية,مؤنث))),حرف وصل(و),معرفة(الميل,مذكر)))),مكمل جملة مركب(مكمل جملة(معرفة(التدريجي,مذكر)),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ل),نكرة(ارتفاع,مذكر))),مكمل جملة مركب(مكمل جملة(تركيب إضافي(نكرة(درجات,مؤنث),معرفة(الحرارة,مؤنث))),مكمل جملة(تركيب إضافي(نكرة(خلال,حياد),تركيب وصفي(معرفة(السنوات,مؤنث),معرفة(السابقة,مؤنث)))))))),جملة فعلية(فعل(لوحظ,متعدي),مفعول به(تركيب إضافي(نكرات متعددة(نكرة(تواجد,مذكر),نكرة(حشرات,مؤنث)),معرفة(المن,مذكر))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(على),معرفة(البادرات,مؤنث))),مكمل جملة مركب(مكمل جملة(تركيب إضافي(ظرف زمان(عقب),معرفة(الإنبات,مذكر))),مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(ظرف زمان(حوالي),تركيب إضافي(وحدة رقمية(10),نكرة(أيام,مؤنث)))))))),جملة فعلية يتصدرها مكمل جملة مقدم(مكمل جملة مركب(مكمل جملة(نكرة(نتيجة,مؤنث)),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ل),تراكيب إضافية معطوفة(تركيب إضافي(نكرة(تغير,مذكر),تركيب وصفي(معرفة(الظروف,مؤنث),معرفة(البيئية,مؤنث))),حرف وصل(و),معرفة(الميل,مذكر)))),مكمل جملة مركب(مكمل جملة(معرفة(التدريجي,مذكر)),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ل),نكرة(ارتفاع,مذكر))),مكمل جملة مركب(مكمل جملة(تركيب إضافي(نكرة(درجات,مؤنث),معرفة(الحرارة,مؤنث))),مكمل جملة(تركيب إضافي(نكرة(خلال,حياد),تركيب وصفي(معرفة(السنوات,مؤنث),معرفة(السابقة,مؤنث)))))))),جملة فعلية(فعل(لوحظ,متعدي),فاعل(تركيب إضافي(نكرات متعددة(نكرة(تواجد,مذكر),نكرة(حشرات,مؤنث)),معرفة(المن,مذكر))),مكمل جملة(شبه جملة(حرف جر(على),معرفة(البادرات,مؤنث))),مفعول به(تركيب إضافي(ظرف زمان(عقب),معرفة(الإنبات,مذكر))),مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(ظرف زمان(حوالي),تركيب إضافي(وحدة رقمية(10),نكرة(أيام,مؤنث)))))))]	7-8
[جملة فعلية(فعل(تظل,متعدي),مفعول به(تركيب إضافي(نكرة(أعداد,مؤنث),معرفة(الحشرات,مؤنث))),مكمل جملة مركب(مكمل جملة(نكرة(منخفضة,مؤنث)),مكمل جملة(تركيب إضافي(نكرة(حسب,مذكر),تركيب وصفي(معرفة(الضغوط,مؤنث),معرفة(البيئية,مؤنث)))))),جملة فعلية(فعل(تظل,متعدي),فاعل(تركيب إضافي(نكرة(أعداد,مؤنث),معرفة(الحشرات,مؤنث))),مفعول به(نكرة(منخفضة,مؤنث)),مكمل جملة(تركيب إضافي(نكرة(حسب,مذكر),تركيب وصفي(معرفة(الضغوط,مؤنث),معرفة(البيئية,مؤنث))))),جملة فعلية(فعل(تظل,متعدي),فاعل(تركيب إضافي(نكرة(أعداد,مؤنث),معرفة(الحشرات,مؤنث))),مكمل جملة(نكرة(منخفضة,مؤنث)),مفعول	7-9

به(تركيب إضافي(نكرة(حسب,مذكر),تركيب وصفي(معرفة(الضغوط,مؤنث),معرفة(البيئية,مؤنث)))))]	
[جملة فعلية(فعل(تتعرض,لازم),مكمل جملة(شبه جملة(حرف جر(إلي),ضمير_متصل(ها))))]	7-10
UnParseable	7-11
UnParseable	7-12
[جملة فعلية(فعل(تبدأ,متعدي),مكمل جملة مركب(مكمل جملة(تراكيب إضافية معطوفة(تركيب إضافي(ظرف مكان(عند),معرفة(الحواف,مؤنث)),حرف وصل(و),معرفة(التعامل,مذكر))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(مع),ضمير_متصل(ها))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),نكرة(سرعة,مؤنث))),مكمل جملة(تركيب إضافي(ظرف زمان(قبل),تركيب إضافي(نكرة(انتشار,مذكر),ضمير_متصل(ها)))))))),جملة فعلية(فعل(تبدأ,متعدي),مفعول به(تراكيب إضافية معطوفة(تركيب إضافي(ظرف مكان(عند),معرفة(الحواف,مؤنث)),حرف وصل(و),معرفة(التعامل,مذكر))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(مع),ضمير_متصل(ها))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),نكرة(سرعة,مؤنث))),مكمل جملة(تركيب إضافي(ظرف زمان(قبل),تركيب إضافي(نكرة(انتشار,مذكر),ضمير_متصل(ها)))))))]	7-13
UnParseable	7-14
[جملة فعلية(فعل يسبقه حرف(لا,يستدعي,متعدي),مفعول به(معرفة(الأمر,مذكر)),مكمل جملة مركب(مكمل جملة(تركيب إضافي(نكرة(استخدام,مذكر),تركيب وصفي(نكرة(مبيدات,مؤنث),نكرة(سريعة,مؤنث)))),مكمل جملة(معرفة(التأثير,مذكر))))]	7-15
[جملة فعلية يتصدرها مكمل جملة مقدم(مكمل جملة(تركيب إضافي(ظرف مكان(عند),تركيب إضافي(نكرة(استخدام,مذكر),تركيب وصفي(معرفة(المكافحة,مؤنث),معرفة(الكيماوية,مؤنث)))),جملة فعلية(فعل(ينصح,متعدي),مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(نكرة(أحد,مذكر),تركيب وصفي(معرفة(المواد,مؤنث),معرفة(الآتية,مؤنث)))))]	7-16
[جملة أسمية(مبتدأ(تركيب إضافي(نكرة(سوبر,مذكر),تركيب إضافي(نكرة(مصرونا,مذكر),تركيب إضافي(نسبة مئوية(94%),نكرة(مستحلب,مذكر))))),خبر(خبر و متعلق به(شبه جملة(حرف جر(ب),تركيب إضافي(نكرة(معدل,مذكر),تركيب إضافي(وحدة رقمية(1),نكرة(لتر,مذكر)))),مكمل جملة(شبه جملة(حرف جر(ل),تركيب إضافي(ظرف زمان(كل),تركيب إضافي(وحدة رقمية(100),نكرات متعددة(نكرة(لتر,مذكر),نكرة(ماء,مذكر)))))))]	7-17
[جملة أسمية(مبتدأ(اسم اشارة(هذا)),خبر(خبر و متعلق به(شبه جملة(حرف جر(مع),نكرة(مراعاة,مؤنث)),مكمل جملة مركب(مكمل جملة(تركيب إضافي(نكرة(استخدام,مذكر),تركيب وصفي(نكرة(رشاشة,مؤنث),نكرة(ظهرية,مؤنث)))),مكمل جملة مركب(مكمل جملة(تركيب إضافي(نكرة(ذات,مؤنث),معرفة_علم(بشبوري,مذكر))),مكمل جملة مركب(مكمل جملة(نكرة(منحني,مذكر)),مكمل جملة(شبه جملة(حرف جر(ل),نكرة(أسفل,مذكر)))))))))]	7-18
[جملة فعلية(فعل(يصل,لازم),فاعل(تركيب إضافي(نكرة(محلول,مذكر),معرفة(الرش,مذكر))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ل),تركيب وصفي(معرفة(سطوح,مذكر),معرفة(السفلي,مذكر)))),مكمل جملة(شبه جملة(حرف جر(ل),معرفة(أوراق,مؤنث)))))]	7-19

Parse Tree structure	No.
[جملة فعلية(فعل(يراعى,متعدي),مفعول به(تركيب إضافي(نكرة(استخدام,مذكر),تركيب وصفي(معرفة(المواد,مؤنث),معرفة(البديلة,مؤنث)))),مكمل جملة(شبه جملة(حرف جر(ل),معرفة(مبيد,مؤنث))))]	8-1
[جملة فعلية(فعل(يؤدي,متعدي),مفعول به(تركيب	8-2

	إضافي(نكرة(تساقط,مذكر),معرفة(الأمطار,مؤنث))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(إلي),معرفة(الإقلال,مذكر))),مكمل جملة(شبه جملة(حرف جر(من),تركيب إضافي(نكرة(فاعلية,مؤنث),ضمير_متصل(ها)))))]
8-3	[جملة فعلية(فعل(يعتبر,متعدي),مفعول به(معرفة(العلاج,مذكر)),مكمل جملة مركب(مكمل جملة(تركيب إضافي(ظرف زمان(بعد),تركيب إضافي(نكرة(طرد,مذكر),معرفة(السنابل,مؤنث)))),مكمل جملة(تركيب إضافي(نكرة(عديم,مذكر),معرفة(الجدوى,مؤنث)))),جملة فعلية(فعل(يعتبر,متعدي),فاعل(معرفة(العلاج,مذكر)),مفعول به(تركيب إضافي(ظرف زمان(بعد),تركيب إضافي(نكرة(طرد,مذكر),معرفة(السنابل,مؤنث)))),مكمل جملة(تركيب إضافي(نكرة(عديم,مذكر),معرفة(الجدوى,مؤنث)))),جملة فعلية(فعل(يعتبر,متعدي),فاعل(معرفة(العلاج,مذكر)),مكمل جملة(تركيب إضافي(ظرف زمان(بعد),تركيب إضافي(نكرة(طرد,مذكر),معرفة(السنابل,مؤنث)))),مفعول به(تركيب إضافي(نكرة(عديم,مذكر),معرفة(الجدوى,مؤنث))))]
8-4	[جملة أسمية(مبتدأ(تركيب تمييزي(نكرة(أفضل,مذكر),نكرة(طريقة,مؤنث))),خبر(خبر يسبقه مكمل جملة(مكمل جملة(شبه جملة(حرف جر(ل),تركيب إضافي(نكرة(مكافحة,مؤنث),معرفة(الفئران,مؤنث)))),جملة أسمية(مبتدأ(ضمير_منفصل(هي)),خبر(خبر و متعلق به(تركيب إضافي(نكرة(استخدام,مذكر),تركيب وصفي(معرفة(المقاومة,مؤنث),معرفة(الجماعية,مؤنث))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(نكرة(أحد,مذكر),تركيب وصفي(معرفة(الطعوم,مؤنث),معرفة(السامة,مؤنث)))),مكمل جملة مركب(مكمل جملة(تركيب إضافي(ظرف مكان(مثل),معرفة_علم - أعلام متتابعة(فوسفيد,معرفة_علم(الزنك,مذكر),مذكر))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(مع),تركيب إضافي(نكرة(جريش,مذكر),تراكيب وصفية معطوفة(معرفة(الذرة,مؤنث),حرف وصل(أو),تركيب وصفي(معرفة(المبيدات,مؤنث),معرفة(المسيلة,مؤنث)))))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ل),معرفة(دم,مذكر))),مكمل جملة(شبه جملة(حرف جر(بال),تركيب وصفي(معرفة(نسب,مؤنث),معرفة(المقررة,مؤنث)))))))))))))]
8-5	UnParseable
8-6	[جملة فعلية(فعل(تتناسب,لازم),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(مع),تركيب وصفي(معرفة(الكثافة,مؤنث),معرفة(العددية,مؤنث))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ل),معرفة(فئران,مؤنث))),مكمل جملة(شبه جملة(حرف جر(مع),تركيب إضافي(تركيب إضافي(نكرة(متابعة,مؤنث),ضمير_متصل(ها)),معرفة(المستمرة,مؤنث)))))))]
8-7	[جملة فعلية يتصدرها مكمل جملة مقدم(مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(بال),معرفة(نسبة,مؤنث))),مكمل جملة(شبه جملة(حرف جر(ل),معرفة(عصافير,مؤنث))),جملة إن و أخواتها(حرف إن (إن),ضمير_متصل(ه),خبر(جملة فعلية(فعل(يفضل,متعدي),مفعول به(جملة إن و أخواتها(حرف إن (إن),خبر(جملة فعلية(فعل(تتم,متعدي),مفعول به(تركيب إضافي(نكرة(زراعة,مؤنث),معرفة(القمح,مذكر))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(في),تركيب وصفي(نكرة(تجمعات,مؤنث),نكرة(كبيرة,مؤنث))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(في),معرفة(المواعيد,مؤنث))),مكمل جملة مركب(مكمل جملة(معرفة(الموصي,مذكر)),مكمل جملة(شبه جملة(حرف جر(ب),ضمير_متصل(ها)))))))))))]
8-8	[جملة إن و أخواتها(حرف إن (إن),خبر(جملة كان و أخواتها(فعل(تكون,متعدي),جملة أسمية(مبتدأ(تركيب إضافي(نكرة(مواعيد,مؤنث),معرفة(الزراعة,مؤنث))),خبر(خبر و متعلق به(شبه جملة(حرف جر(في),تركيب وصفي(معرفة(الحقول,مؤنث),معرفة(المجاورة,مؤنث))),مكمل جملة مركب(مكمل جملة(نكرة(متقاربة,مؤنث)),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ل),تركيب إضافي(نكرات متعددة(نكرة(تقليل,مذكر),نكرات متعددة(نكرة(أضرار,مؤنث),نكرة(مهاجمة,مؤنث))),معرفة(العصافير,مؤنث)))),مكمل جملة(شبه

	جملة(حرف جر(مع),تركيب إضافي(نكرة(إزالة,مؤنث),معرفة(الأعشاش,مؤنث)))))))))))]
8-9	[جملة كان و أخواتها(فعل(كانت,متعدي),خبر(خبر و متعلق به(نكرة(متواجدة,مؤنث),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(على),تركيب وصفي(معرفة(الأشجار,مؤنث),معرفة(المحيطة,مؤنث))))),مكمل جملة(شبه جملة(حرف جر(ل),معرفة(حقل,مذكر)))))))]
8-10	[جملة فعلية(فعل(ينصح,متعدي),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(في),تركيب وصفي(معرفة(الأراضي,مؤنث),معرفة(الموبوءة,مؤنث))))),مكمل جملة(شبه جملة(حرف جر(بال),معرفة(قواقع,مؤنث))))),مفعول به(تركيب إضافي(نكرات متعددة(نكرة(زراعة,مؤنث),نكرة(داير,مذكر))))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(من),معرفة(البرسيم,مذكر))),مكمل جملة(تركيب إضافي(نكرات متعددة(نكرة(حول,مذكر),نكرة(حقل,مذكر)),معرفة(القمح,مذكر)))))]
8-11	[جملة كان و أخواتها(فعل(يكون,متعدي),خبر(خبر و متعلق به(شبه جملة(حرف جر(ب),تركيب إضافي(نكرات متعددة(نكرة(مثابة,مؤنث),نكرة(مصيدة,مؤنث))))),مكمل جملة(شبه جملة(حرف جر(ل),معرفة(قواقع,مؤنث)))))]
8-12	[جملة أسمية(خبر مقدم(شبه جملة(حرف جر(من),تركيب إضافي(نكرة(أهم,مذكر),ضمير_متصل(ها))))),مبتدأ(تركيب إضافي(نكرة(أصداء,مؤنث),معرفة(القمح,مذكر))),مكمل جملة معطوفة(مكمل جملة مركب(مكمل جملة(معرفة(الثلاثة,مؤنث)),مكمل جملة(معرفة(الأسود,مذكر))),حرف وصل(و),مكمل جملة معطوفة(مكمل جملة(معرفة_علم(البرتقالي,مذكر)),حرف وصل(و),مكمل جملة معطوفة(مكمل جملة(معرفة(الأصفر,مذكر)),حرف وصل(و),مكمل جملة(تركيب إضافي(نكرة(مرض,مذكر),تركيب وصفي(معرفة(التفحم,مذكر),معرفة(السائب,مذكر)))))))]
8-13	UnParseable
8-14	[جملة فعلية يتصدرها مكمل جملة مقدم(مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(في),تركيب وصفي(نكرات متعددة(نكرة(حالة,مؤنث),نكرات متعددة(نكرة(ظهور,مذكر),نكرة(طرز,مؤنث))),نكرة(جديدة,مؤنث))))),مكمل جملة(شبه جملة(حرف جر(من),تركيب وصفي(معرفة(الأمراض,مؤنث),معرفة(الفطرية,مؤنث)))))),جملة فعلية(فعل(تصيب,متعدي),مكمل جملة(تركيب إضافي(ظرف مكان(بعض),تركيب وصفي(معرفة(الأصناف,مؤنث),معرفة(المنزرعة,مؤنث))))),مفعول به(جملة إن و أخواتها(فعل(فإن,متعدي),خبر(خبر و متعلق به(تركيب إضافي(نكرات متعددة(نكرة(قسم,مذكر),نكرة(بحوث,مؤنث)),معرفة(القمح,مذكر)),جملة فعلية(فعل(يعمل,لازم),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(على),تركيب إضافي(نكرة(إحلال,مذكر),ضمير_متصل(ها))))),مكمل جملة(شبه جملة(حرف جر(ب),تركيب وصفي(نكرة(أصناف,مؤنث),نكرة(جديدة,مؤنث)))))),مكمل جملة معطوفة(مكمل جملة(تركيب إضافي(نكرة(عالية,مؤنث),معرفة(المحصول,مذكر))),حرف وصل(و),مكمل جملة مركب(مكمل جملة(نكرة(مقاومة,مؤنث)),مكمل جملة(شبه جملة(حرف جر(ل),معرفة(مرض,مذكر)))))))))),جملة فعلية يتصدرها مكمل جملة مقدم(مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(في),تركيب وصفي(نكرات متعددة(نكرة(حالة,مؤنث),نكرات متعددة(نكرة(ظهور,مذكر),نكرة(طرز,مؤنث))),نكرة(جديدة,مؤنث))))),مكمل جملة(شبه جملة(حرف جر(من),تركيب وصفي(معرفة(الأمراض,مؤنث),معرفة(الفطرية,مؤنث)))))),جملة فعلية(فعل(تصيب,متعدي),مكمل جملة(تركيب إضافي(ظرف مكان(بعض),تركيب وصفي(معرفة(الأصناف,مؤنث),معرفة(المنزرعة,مؤنث))))),مفعول به(جملة إن و أخواتها(فعل(فإن,متعدي),جملة أسمية(مبتدأ(تركيب إضافي(نكرات متعددة(نكرة(قسم,مذكر),نكرة(بحوث,مؤنث)),معرفة(القمح,مذكر))),خبر(جملة فعلية(فعل(يعمل,لازم),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(على),تركيب إضافي(نكرة(إحلال,مذكر),ضمير_متصل(ها))))),مكمل جملة(شبه جملة(حرف جر(ب),تركيب وصفي(نكرة(أصناف,مؤنث),نكرة(جديدة,مؤنث)))))))))),مكمل جملة معطوفة(مكمل جملة(تركيب

إضافي(نكرة(عالية,مؤنث),معرفة(المحصول,مذكر))),حرف وصل(و),مكمل جملة مركب(مكمل جملة(نكرة(مقاومة,مؤنث)),مكمل جملة(شبه جملة(حرف جر(ل),معرفة(مرض,مذكر)))))]	
[جملة فعلية يتصدرها مكمل جملة مقدم(مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(بال),معرفة(نسبة,مؤنث))),مكمل جملة(شبه جملة(حرف جر(ل),تركيب إضافي(نكرة(مرض,مذكر),تركيب وصفي(معرفة(التفحم,مذكر),معرفة(السائب,مذكر))))), جملة إن و أخواتها(فعل(فإن,متعدي),جملة أسمية(مبتدأ(معرفة(التقاوي,مؤنث)),خبر(خبر يسبقه مكمل جملة(مكمل جملة مركب(مكمل جملة(معرفة(الموصي,مذكر)),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),ضمير_متصل(ها))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(من),تراكيب إضافية معطوفة(تركيب إضافي(ظرف زمان(قبل),تركيب إضافي(نكرات متعددة(نكرة(قسم,مذكر),نكرة(بحوث,مؤنث)),معرفة(القمح,مذكر))),حرف وصل(و),معرفة(الموزعة,مؤنث)))),مكمل جملة(شبه جملة(حرف جر(عن),تركيب إضافي(نكرات متعددة(نكرة(طريق,مذكر),نكرة(وزارة,مؤنث)),معرفة(الزراعة,مؤنث)))))),جملة فعلية(فعل(يتم,متعدي),مفعول به(تركيب إضافي(نكرة(معاملة,مؤنث),ضمير_متصل(ها))),مكمل جملة مركب(مكمل جملة(تركيب إضافي(نكرة(ضد,مذكر),معرفة(المرض,مذكر))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(بال),تركيب وصفي(معرفة(مطهر,مؤنث),تركيب وصفي(معرفة(الفطرية,مؤنث),معرفة(الجهازية,مؤنث))))),مكمل جملة(شبه جملة(حرف جر(ل),تركيب إضافي(نكرات(نكرة(منع,مذكر),نكرات معطوفة(نكرة(ظهور,مذكر),حرف وصل(و),نكرة(انتشار,مذكر))),معرفة(المرض,مذكر)))))))))]	8-15
[جملة إن و أخواتها(حرف إن (إن),ضمير_متصل(ه),خبر(جملة فعلية(فعل(يوصى,متعدي),مكمل جملة(جملة اعتراضية(كلمة عارضة(دائما)),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(نكرة(استعمال,مذكر),تركيب وصفي(معرفة(التقاوي,مؤنث),معرفة(المعتمدة,مؤنث))))),مكمل جملة(شبه جملة(حرف جر(ل),تركيب وصفي(معرفة(أصناف,مؤنث),معرفة(الجديدة,مؤنث)))))),مفعول به(تراكيب إضافية معطوفة(تركيب إضافي(نكرة(عالية,مؤنث),معرفة(المحصول,مذكر)),حرف وصل(و),معرفة(الموزعة,مؤنث))),مكمل جملة معطوفة(مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(عن),تراكيب إضافية معطوفة(تركيب إضافي(نكرات متعددة(نكرة(طريق,مذكر),نكرة(وزارة,مؤنث)),معرفة(الزراعة,مؤنث)),حرف وصل(و),تركيب إضافي(نكرة(عدم,مذكر),معرفة(اللجوء,مذكر))))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(إلي),تراكيب إضافية معطوفة(تركيب إضافي(نكرة(زراعة,مؤنث),تركيب وصفي(معرفة(الأصناف,مؤنث),معرفة(القديمة,مؤنث))),حرف وصل(و),تركيب إضافي(ظرف زمان(غير),معرفة(الموصي,مذكر))))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),ضمير_متصل(ها))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ل),تراكيب إضافية معطوفة(تركيب إضافي(نكرات متعددة(نكرة(شدة,مؤنث),نكرة(إصابة,مؤنث)),ضمير_متصل(ها)),حرف وصل(و),تراكيب إضافية معطوفة(تركيب إضافي(نكرات متعددة(نكرة(انخفاض,مذكر),نكرة(محصول,مذكر)),ضمير_متصل(ها)),حرف وصل(و),معرفة(الزراعة,مؤنث))))),مكمل جملة(شبه جملة(حرف جر(في),تركيب وصفي(معرفة(المواعيد,مؤنث),معرفة(المقررة,مؤنث)))))),حرف وصل(و),مكمل جملة مركب(مكمل جملة(نكرة(اتباع,مذكر)),مكمل جملة مركب(مكمل جملة(تركيب إضافي(نكرة(كافة,مؤنث),تركيب وصفي(معرفة(التوصيات,مؤنث),معرفة(الفنية,مؤنث)))),مكمل جملة(شبه جملة(حرف جر(ل),تركيب إضافي(نكرة(إنتاج,مذكر),معرفة(القمح,مذكر)))))))))]	8-16
[جملة فعلية(فعل(تظهر,لازم),فاعل(معرفة(العلامات,مؤنث)),مكمل جملة معطوفة(مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(على),تركيب إضافي(نكرة(شكل,مذكر),تركيب وصفي(نكرة(بقع,مؤنث),نكرة(صفراء,مؤنث))))),مكمل جملة مركب(مكمل جملة(نكرة(منفصلة,مؤنث)),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(إلي),ضمير_متصل(ها))),مكمل جملة مركب(مكمل جملة(تركيب	8-17

	وصفي(نكرة(مظهر,مذكر),نكرة(مسحوقي,مذكر))),مكمل جملة مركب(مكمل جملة(نكرة(مرتبة,مؤنث)),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(في),تركيب وصفي(نكرة(صفوف,مؤنث),نكرة(طولية,مؤنث)))),مكمل جملة(شبه جملة(حرف جر(مع),تركيب إضافي(نكرة(محور,مذكر),معرفة(الورقة,مؤنث)))))))))),حرف وصل(و),مكمل جملة(نكرة(متوازية,مؤنث)))))]
8-18	[جملة فعلية(فعل(تظهر,لازم),فاعل(معرفة(الإصابة,مؤنث)),مكمل جملة معطوفة(مكمل جملة(شبه جملة(حرف جر(على),معرفة(الأوراق,مؤنث),حرف وصل(و),معرفة(الأغماد,مؤنث))),حرف وصل(و),مكمل جملة(معرفة(القنابع,مؤنث)))))]
8-19	[جملة فعلية يتصدرها مكمل جملة مقدم(مكمل جملة(شبه جملة(حرف جر(في),تراكيب إضافية معطوفة(تركيب إضافي(نكرة(نهاية,مؤنث),معرفة(الموسم,مذكر)),حرف وصل(أو),تركيب إضافي(ظرف مكان(عند),تركيب إضافي(نكرة(اشتداد,مذكر),معرفة(الحرارة,مؤنث))))))),جملة فعلية(فعل(يتحول,لازم),فاعل(تركيب وصفي(معرفة(اللون,مذكر),معرفة(الأصفر,مذكر))),مكمل جملة(شبه جملة(حرف جر(إلي),تركيب وصفي(معرفة(اللون,مذكر),تركيب وصفي(معرفة(المسود,مذكر),معرفة(اللامع,مذكر)))))))]

Parse Tree structure	No.
[جملة فعلية(فعل(يناسب,متعدي),مفعول به(معرفة(المرض,مذكر)),مكمل جملة مركب(مكمل جملة(تركيب إضافي(نكرة(درجات,مؤنث),تركيب وصفي(معرفة(الحرارة,مؤنث),معرفة(المنخفضة,مؤنث)))),مكمل جملة(نكرة(نهارا,مذكر)))),جملة فعلية(فعل(يناسب,متعدي),فاعل(معرفة(المرض,مذكر)),مفعول به(تركيب إضافي(نكرة(درجات,مؤنث),تركيب وصفي(معرفة(الحرارة,مؤنث),معرفة(المنخفضة,مؤنث)))),مكمل جملة(نكرة(نهارا,مذكر))),جملة فعلية(فعل(يناسب,متعدي),فاعل(معرفة(المرض,مذكر)),مكمل جملة(تركيب إضافي(نكرة(درجات,مؤنث),تركيب وصفي(معرفة(الحرارة,مؤنث),معرفة(المنخفضة,مؤنث)))),مفعول به(نكرة(نهارا,مذكر)))]	9-1
[جملة إن و أخواتها(حرف إن (إن),خبر(جملة كان و أخواتها(فعل(يكون,متعدي),جملة أسمية(مبتدأ(معرفة(الفرق,مذكر)),خبر(خبر و متعلق به(تركيب إضافي(ظرف مكان(بين),تركيب إضافي(نكرة(درجات,مؤنث),معرفة(الحرارة,مؤنث))),مكمل جملة مركب(مكمل جملة(تراكيب إضافية معطوفة(تركيب إضافي(ظرف مكان(بين),معرفة(الليل,مذكر)),حرف وصل(و),معرفة(النهار,مذكر))),مكمل جملة(نكرة(كبيرا,مذكر)))))))))]	9-2
[جملة فعلية(فعل(ينتشر,لازم),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(في),تركيب إضافي(نكرات(نكرة(مناطق,مؤنث),نكرات معطوفة(نكرة(شمال,مذكر),حرف وصل(و),نكرات معطوفة(نكرة(وسط,مذكر),حرف وصل(و),نكرة(جنوب,مذكر)))),معرفة_علم(الدلتا,مؤنث)))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(عن),تركيب وصفي(معرفة(المناطق,مؤنث),معرفة(الأخرى,مؤنث)))),مكمل جملة(شبه جملة(حرف جر(في),تركيب إضافي(نكرات(نكرات معطوفة(نكرة(وسط,مذكر),حرف وصل(و),نكرة(جنوب,مذكر))),معرفة(الوادي,مذكر)))))))]	9-3
[جملة فعلية(فعل(تظهر,لازم),فاعل(معرفة(العلامات,مؤنث)),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(على),تركيب وصفي(نكرة(هيئة,مؤنث),تركيب وصفي(نكرة(بقع,مؤنث),نكرة(مسحوقية,مؤنث))))),مكمل جملة مركب(مكمل جملة(تركيب إضافي(نكرة(لون,مذكر),ضمير_متصل(ها))),مكمل جملة مركب(مكمل جملة(تركيب وصفي(نكرة(بني,مذكر),نكرة(فاتح,مذكر))),مكمل جملة مركب(مكمل جملة(تركيب وصفي(نكرة(مستديرة,مؤنث),نكرة(مبعثرة,مؤنث))),مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(نكرات متعددة(نكرة(دون,حياد),نكرة(نظام,مذكر)))))))))]	9-4
[جملة فعلية(فعل(يمكن,متعدي),مفعول به(جملة إن و أخواتها(حرف إن (إن),خبر(جملة	9-5

	فعلية(فعل(تظهر,لازم),مكمل جملة(شبه جملة(حرف جر(على),تركيب إضافي(نكرة(سطحي,مذكر),معرفة(الورقة,مؤنث)))))))))]
9-6	[جملة فعلية(فعل يسبقه حرف(لا,تظهر,لازم),فاعل(معرفة(الإصابة,مؤنث)),مكمل جملة(جملة اعتراضية(كلمة عارضة(إلا)),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(على),معرفة(الأوراق,مؤنث))),مكمل جملة(نكرة(فقط,حياد)))))]
9-7	[جملة فعلية(فعل(يناسب,متعدي),مفعول به(معرفة(المرض,مذكر)),مكمل جملة مركب(مكمل جملة(تركيب إضافي(نكرة(درجات,مؤنث),تركيب وصفي(معرفة(الحرارة,مؤنث),معرفة(المتوسطة,مؤنث)))),مكمل جملة(نكرة(نوعا,مذكر)))),جملة فعلية(فعل(يناسب,متعدي),فاعل(معرفة(المرض,مذكر)),مفعول به(تركيب إضافي(نكرة(درجات,مؤنث),تركيب وصفي(معرفة(الحرارة,مؤنث),معرفة(المتوسطة,مؤنث)))),مكمل جملة(نكرة(نوعا,مذكر))),جملة فعلية(فعل(يناسب,متعدي),فاعل(معرفة(المرض,مذكر)),مكمل جملة(تركيب إضافي(نكرة(درجات,مؤنث),تركيب وصفي(معرفة(الحرارة,مؤنث),معرفة(المتوسطة,مؤنث)))),مفعول به(نكرة(نوعا,مذكر)))]
9-8	[جملة فعلية(فعل(يسود,لازم),مكمل جملة(شبه جملة(حرف جر(في),نكرة(مناطق,مؤنث))),فاعل(تراكيب إضافية معطوفة(تركيب إضافي(نكرات(نكرات معطوفة(نكرة(غرب,مذكر),حرف وصل(و),نكرات معطوفة(نكرة(وسط,مذكر),حرف وصل(و),نكرة(جنوب,مذكر)))),معرفة_علم(الدلتا,مؤنث)),حرف وصل(و),تركيب وصفي(معرفة_علم(مصر,مؤنث),معرفة(الوسطى,مؤنث)))))),جملة فعلية(فعل(يسود,لازم),مكمل جملة(شبه جملة(حرف جر(في),نكرة(مناطق,مؤنث))),فاعل(تراكيب إضافية معطوفة(تركيب إضافي(نكرات(نكرات معطوفة(نكرة(غرب,مذكر),حرف وصل(و),نكرات معطوفة(نكرة(وسط,مذكر),حرف وصل(و),نكرة(جنوب,مذكر)))),معرفة_علم(الدلتا,مؤنث)),حرف وصل(و),معرفة_علم(مصر,مؤنث))),مكمل جملة(معرفة(الوسطى,مؤنث)))]
9-9	[جملة فعلية مركبة(جملة فعلية(فعل(تظهر,لازم),فاعل(معرفة(الإصابة,مؤنث)),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(على),تركيب وصفي(نكرة(هيئة,مؤنث),تركيب وصفي(نكرة(بقع,مؤنث),نكرة(مسحوقية,مؤنث))))),مكمل جملة مركب(مكمل جملة(تركيب إضافي(نكرة(لون,مذكر),ضمير_متصل(ها))),مكمل جملة(تركيب وصفي(نكرة(بني,مذكر),نكرة(داكن,مذكر)))))),حرف وصل(أو),جملة فعلية يتصدرها مكمل جملة مقدم(مكمل جملة مركب(مكمل جملة(نكرة(مسود,مذكر)),مكمل جملة(تركيب إضافي(ظرف زمان(غير),نكرة(منتظمة,مؤنث)))),جملة فعلية(فعل(تلتحم,لازم),مكمل جملة(شبه جملة(حرف جر(مع),تركيب إضافي(ظرف مكان(بعض),ضمير_متصل(ها)))))))]
9-10	[جملة فعلية(فعل(تظهر,لازم),فاعل(معرفة(الإصابة,مؤنث)),مكمل جملة معطوفة(مكمل جملة(شبه جملة(حرف جر(على),معرفة(الساق,مؤنث),حرف وصل(و),معرفة(الأوراق,مؤنث))),حرف وصل(و),مكمل جملة(معرفة(السنابل,مؤنث)))))]
9-11	[جملة فعلية(فعل(تسبب,متعدي),مفعول به(تركيب وصفي(معرفة(الإصابة,مؤنث),معرفة(الشديدة,مؤنث))),مكمل جملة مركب(مكمل جملة(نكرة(تهتك,مذكر)),مكمل جملة(شبه جملة(حرف جر(في),تراكيب وصفية معطوفة(تركيب وصفي(معرفة(الأنسجة,مؤنث),معرفة(الدعامية,مؤنث)),حرف وصل(و),معرفة(الناقلة,مؤنث)))))),جملة فعلية(فعل(تسبب,متعدي),فاعل(تركيب وصفي(معرفة(الإصابة,مؤنث),معرفة(الشديدة,مؤنث))),مفعول به(نكرة(تهتك,مذكر)),مكمل جملة(شبه جملة(حرف جر(في),تراكيب وصفية معطوفة(تركيب وصفي(معرفة(الأنسجة,مؤنث),معرفة(الدعامية,مؤنث)),حرف وصل(و),معرفة(الناقلة,مؤنث)))))]
9-12	[جملة فعلية(فعل(تسبب,متعدي),مفعول به(تراكيب إضافية معطوفة(تركيب إضافي(نكرة(رقاد,مذكر),معرفة(النباتات,مؤنث)),حرف وصل(و),تركيب إضافي(نكرة(ضعف,مذكر),معرفة(المحصول,مذكر)))))]
9-13	[جملة فعلية(فعل(يناسب,متعدي),مفعول به(معرفة(المرض,مذكر)),مكمل جملة مركب(مكمل

	جملة(تركيب إضافي(نكرة(درجات,مؤنث),تركيب وصفي(معرفة(الحرارة,مؤنث),معرفة(العالية,مؤنث)))),مكمل جملة(نكرة(نوعا,مذكر)))),جملة فعلية(فعل(يناسب,متعدي),فاعل(معرفة(المرض,مذكر)),مفعول به(تركيب إضافي(نكرة(درجات,مؤنث),تركيب وصفي(معرفة(الحرارة,مؤنث),معرفة(العالية,مؤنث)))),مكمل جملة(نكرة(نوعا,مذكر))),جملة فعلية(فعل(يناسب,متعدي),فاعل(معرفة(المرض,مذكر)),مكمل جملة(تركيب إضافي(نكرة(درجات,مؤنث),تركيب وصفي(معرفة(الحرارة,مؤنث),معرفة(العالية,مؤنث)))),مفعول به(نكرة(نوعا,مذكر)))]
9-14	[جملة أسمية(مبتدأ(معرفة(المرض,مذكر)),خبر(جملة فعلية(فعل يسبقه حرف(لا,يشكل,متعدي),مفعول به(نكرة(خطورة,مؤنث)),مكمل جملة مركب(مكمل جملة(نكرة(نظرا,مذكر)),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ل),تركيب إضافي(نكرة(درجة,مؤنث),تركيب وصفي(معرفة(المقاومة,مؤنث),معرفة(العالية,مؤنث)))),مكمل جملة(شبه جملة(حرف جر(في),تركيب إضافي(نكرة(معظم,مذكر),تركيب وصفي(معرفة(الأصناف,مؤنث),تركيب وصفي(معرفة(التجارية,مؤنث),معرفة(المنزرعة,مؤنث))))))))),جملة فعلية يتصدرها مكمل جملة مقدم(مكمل جملة(معرفة(المرض,مذكر)),جملة فعلية(فعل يسبقه حرف(لا,يشكل,متعدي),مفعول به(نكرة(خطورة,مؤنث)),مكمل جملة مركب(مكمل جملة(نكرة(نظرا,مذكر)),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ل),تركيب إضافي(نكرة(درجة,مؤنث),تركيب وصفي(معرفة(المقاومة,مؤنث),معرفة(العالية,مؤنث)))),مكمل جملة(شبه جملة(حرف جر(في),تركيب إضافي(نكرة(معظم,مذكر),تركيب وصفي(معرفة(الأصناف,مؤنث),تركيب وصفي(معرفة(التجارية,مؤنث),معرفة(المنزرعة,مؤنث))))))))),جملة فعلية يتصدرها مكمل جملة مقدم(مكمل جملة(معرفة(المرض,مذكر)),جملة فعلية(فعل يسبقه حرف(لا,يشكل,متعدي),مفعول به(نكرة(خطورة,مؤنث)),مكمل جملة مركب(مكمل جملة(نكرة(نظرا,مذكر)),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ل),تركيب إضافي(نكرة(درجة,مؤنث),تركيب وصفي(معرفة(المقاومة,مؤنث),معرفة(العالية,مؤنث)))),مكمل جملة(شبه جملة(حرف جر(في),تركيب إضافي(نكرة(معظم,مذكر),تركيب وصفي(معرفة(الأصناف,مؤنث),تركيب وصفي(معرفة(التجارية,مؤنث),معرفة(المنزرعة,مؤنث))))))))))]
9-15	UnParseable
9-16	[جملة فعلية(فعل(تظهر,لازم),فاعل(تركيب إضافي(نكرة(علامات,مؤنث),معرفة(الإصابة,مؤنث))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(على),معرفة(النباتات,مؤنث))),مكمل جملة(تركيب إضافي(ظرف مكان(عند),تركيب إضافي(نكرة(طرد,مذكر),معرفة(السنابل,مؤنث)))))]
9-17	UnParseable
9-18	[جملة فعلية(فعل(تتطاير,لازم),فاعل(تركيب إضافي(نكرات متعددة(نكرة(نتيجة,مؤنث),نكرة(اهتزاز,مذكر)),معرفة(النباتات,مؤنث))),مكمل جملة(شبه جملة(حرف جر(ب),تراكيب إضافية معطوفة(تركيب إضافي(نكرة(فعل,مذكر),معرفة(الرياح,مؤنث)),حرف وصل(أو),تركيب إضافي(ظرف زمان(غير),ضمير_متصل(ها)))))]
9-19	UnParseable
9-20	[جملة فعلية(فعل(تنبت,متعدي),فاعل(معرفة(الجرثومة,مؤنث))),جملة فعلية(فعل(تنبت,متعدي),مفعول به(معرفة(الجرثومة,مؤنث)))]
9-21	[جملة فعلية(فعل(تسلك,متعدي),مفعول به(تركيب إضافي(نكرات متعددة(نكرة(نفس,مذكر),نكرات متعددة(نكرة(سلوك,مذكر),نكرة(حبة,مؤنث))),معرفة(اللقاح,مذكر)))]
9-22	[جملة فعلية(فعل(تصل,لازم),مكمل جملة(شبه جملة(حرف جر(إلي),معرفة(المبيض,مذكر)))]
9-23	[جملة فعلية(فعل(يسكن,لازم),فاعل(معرفة(الفطر,مذكر)),مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(نكرة(جوار,مذكر),معرفة(الجنين,مذكر)))))]
9-24	[جملة فعلية يتصدرها مكمل جملة مقدم(مكمل جملة(تراكيب إضافية معطوفة(تركيب إضافي(ظرف زمان(بعد),معرفة(الحصاد,مذكر)),حرف وصل(و),معرفة(الدراس,مذكر))),جملة فعلية(فعل يسبقه حرف(لا,يظهر,لازم),مكمل جملة(شبه جملة(حرف جر(على),معرفة(الحبوب,مؤنث))),فاعل(تركيب إضافي(نكرة(أي,حياد),تركيب وصفي(نكرة(أعراض,مؤنث),نكرة(مرضية,مؤنث)))))]
9-25	[جملة فعلية يتصدرها مكمل جملة مقدم(مكمل جملة مركب(مكمل جملة(تركيب إضافي(ظرف مكان(عند),تركيب إضافي(نكرة(زراعة,مؤنث),تركيب وصفي(معرفة(الحبوب,مؤنث),معرفة(المصابة,مؤنث))))),مكمل جملة(شبه جملة(حرف جر(في),تركيب وصفي(معرفة(الموسم,مذكر),معرفة(التالي,مذكر))))),جملة

	فعلية(فعل(ينشط,لازم),فاعل(معرفة(الفطر,مذكر)))]
9-26	[جملة فعلية(فعل(يستطيل,لازم),مكمل جملة(شبه جملة(حرف جر(مع),تركيب إضافي(نكرة(استطالة,مؤنث),معرفة(النبات,مذكر)))]
9-27	[جملة فعلية يتصدرها مكمل جملة مقدم(مكمل جملة(تركيب إضافي(ظرف مكان(عند),تركيب إضافي(نكرة(تكوين,مذكر),معرفة(السنبلة,مؤنث))),جملة فعلية(فعل(يقضي,لازم),مكمل جملة(شبه جملة(حرف جر(على),معرفة(الحبوب,مؤنث)))]
9-28	UnParseable
9-29	[جملة فعلية(فعل(تظهر,لازم),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(على),تركيب إضافي(نكرة(هيئة,مؤنث),تركيب وصفي(نكرة(مسحوق,مذكر),نكرة(أسود,مذكر))),مكمل جملة(تراكيب إضافية معطوفة(تركيب إضافي(ظرف مكان(عند),تركيب إضافي(نكرة(تكشف,مذكر),معرفة(السنبلة,مؤنث))),حرف وصل(أو),تركيب إضافي(نكرة(طرد,مذكر),معرفة(السنابل,مؤنث)))))]
9-30	[جملة فعلية(فعل(تعيد,متعدي),فاعل(تركيب إضافي(نكرة(دورة,مؤنث),معرفة(الحياة,مؤنث))),جملة فعلية(فعل(تعيد,متعدي),مفعول به(تركيب إضافي(نكرة(دورة,مؤنث),معرفة(الحياة,مؤنث)))]

No.	Parse Tree structure
10-1	[جملة فعلية(فعل(يلاحظ,متعدي),مفعول به(جملة إن و أخواتها(حرف إن (إن),جملة أسمية(مبتدأ(تركيب وصفي(معرفة(السنابل,مؤنث),معرفة(المتفحمة,مؤنث))),خبر(جملة فعلية(فعل(تظهر,لازم),مكمل جملة مركب(مكمل جملة(نكرة(مبكرا,مذكر)),مكمل جملة مركب(مكمل جملة(تركيب إضافي(ظرف زمان(قبل),تركيب إضافي(نكرة(بقية,مؤنث),معرفة(السنابل,مؤنث))),مكمل جملة(شبه جملة(حرف جر(ب),نكرة(يوم,مذكر),حرف وصل(أو),نكرة(ثلاثة,مؤنث)))))))))))]
10-2	[جملة فعلية يتصدرها مكمل جملة مقدم(مكمل جملة(شبه جملة(حرف جر(ل),تركيب إضافي(نكرة(مقاوم,مؤنث),تركيب بدل(اسم اشارة(هذا),معرفة(المرض,مذكر))),جملة فعلية(فعل(يجب,متعدي),مفعول به(تراكيب إضافية معطوفة(تركيب إضافي(نكرة(استعمال,مذكر),تركيب وصفي(معرفة(التقاوي,مؤنث),معرفة(المعتمدة,مؤنث))),حرف وصل(و),معرفة(المعاملة,مؤنث))),مكمل جملة(شبه جملة(حرف جر(بال),تركيب وصفي(معرفة(مطهر,مؤنث),معرفة(الفطرية,مؤنث)))))]
10-3	[جملة أسمية(مبتدأ(تركيب إضافي(نكرة(معاملة,مؤنث),معرفة(التقاوي,مؤنث))),خبر(خبر و متعلق به(تركيب إضافي(ظرف زمان(قبل),معرفة(الزراعة,مؤنث)),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(بال),تركيب وصفي(معرفة(مبيد,مذكر),معرفة(الفطري,معرفة_علم - أعلام متتابعة(سومي,معرفة_علم(ايت,مذكر),مذكر),مذكر))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(نكرة(معدل,مذكر),تركيب إضافي(وحدة رقمية(3),نكرة(جم,مذكر)))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ل),تركيب إضافي(ظرف زمان(كل),تركيب إضافي(نكرة(كيلو,مذكر),تركيب وصفي(نكرة(جرام,مذكر),نكرة(واحد,مذكر))))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(من),معرفة(التقاوي,مؤنث))),مكمل جملة(شبه جملة(حرف جر(مع),تركيب وصفي(معرفة(الخلط,مذكر),معرفة(الجيد,مذكر)))))))))))]
10-4	[جملة فعلية(فعل(تظهر,لازم),فاعل(تركيب إضافي(نكرة(علامات,مؤنث),معرفة(الإصابة,مؤنث))),مكمل جملة معطوفة(مكمل جملة(شبه جملة(حرف جر(على),معرفة(الأوراق,مؤنث),حرف وصل(و),معرفة(السيقان,مؤنث))),حرف وصل(و),مكمل جملة مركب(مكمل جملة(معرفة(السنابل,مؤنث)),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(على),تركيب وصفي(نكرات متعددة(نكرة(هيئة,مؤنث),نكرة(بقع,مؤنث)),نكرة(بيضاء,مؤنث))),مكمل جملة(تركيب إضافي(ظرف زمان(غير),نكرة(منتظمة,مؤنث)))))))]
10-5	[جملة فعلية(فعل(تتحد,لازم),مكمل جملة(شبه جملة(حرف جر(مع),تركيب إضافي(ظرف

	مكان(بعض,ضمير_متصل(ها)))))]
10-6	[جملة كان و أخواتها(فعل(يكون,متعدي),خبر(خبر و متعلق به(شبه جملة(حرف جر(ل),ضمير_متصل(ها)),مكمل جملة(تركيب وصفي(نكرة(ملمس,مذكر),نكرة(قطني,مذكر))))))]
10-7	[جملة فعلية(فعل(يتحول,لازم),فاعل(معرفة(اللون,مذكر)),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(إلي),معرفة(الرمادي,مذكر))),مكمل جملة(شبه جملة(حرف جر(مع),تراكيب إضافية معطوفة(تركيب إضافي(نكرة(تقدم,مذكر),معرفة(الإصابة,مؤنث)),حرف وصل(و),تركيب إضافي(نكرة(اصفرار,مذكر),معرفة(الأوراق,مؤنث)))))))]
10-8	UnParseable
10-9	[جملة فعلية يتصدرها مكمل جملة مقدم(مكمل جملة(شبه جملة(حرف جر(ل),تركيب إضافي(نكرة(مقاوم,مؤنث),معرفة(المرض,مذكر)))),جملة فعلية(فعل(يجب,لازم),فاعل(تركيب إضافي(نكرة(استعمال,مذكر),تركيب وصفي(معرفة(التقاوي,مؤنث),معرفة(المعتمدة,مؤنث)))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ال),معرفة(أصناف,مؤنث))),مكمل جملة مركب(مكمل جملة(معرفة(الموصي,مذكر)),مكمل جملة(شبه جملة(حرف جر(ب),ضمير_متصل(ها)))))))]
10-10	[جملة أسمية(مبتدأ(تركيب إضافي(نكرات متعددة(نكرة(فيروس,مذكر),نكرة(تقزم,مذكر)),تركيب وصفي(معرفة(الشعير,مذكر),معرفة(الأصفر,مذكر)))),خبر(شبه جملة(حرف جر(على),معرفة(القمح,مذكر))))]
10-11	[جملة فعلية يتصدرها مكمل جملة مقدم(مكمل جملة(شبه جملة(حرف جر(في),ضمير_متصل(ه))),جملة فعلية(فعل(تتلون,لازم),فاعل(تركيب إضافي(نكرات متعددة(نكرة(قمة,مؤنث),نكرات متعددة(نكرة(أوراق,مؤنث),نكرة(نبات,مذكر))),معرفة(القمح,مذكر))),مكمل جملة(شبه جملة(حرف جر(بال),تراكيب وصفية معطوفة(تركيب وصفي(معرفة(لون,مذكر),معرفة(الأصفر,مذكر)),حرف وصل(أو),معرفة(القرمزي,مذكر)))))]
10-12	UnParseable
10-13	[جملة فعلية(فعل(ينتقل,لازم),فاعل(معرفة(المرض,مذكر)),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(عن),تركيب إضافي(نكرة(طريق,مذكر),معرفة(الإصابة,مؤنث)))),مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(نكرة(حشرة,مؤنث),معرفة(المن,مذكر)))))]
10-14	[جملة فعلية(فعل(تعتبر,متعدي),مفعول به(تركيب وصفي(معرفة(الناقل,مذكر),معرفة(الرئيسي,مذكر))),مكمل جملة(شبه جملة(حرف جر(ل),معرفة(مرض,مذكر))))]
10-15	[جملة فعلية يتصدرها مكمل جملة مقدم(مكمل جملة(شبه جملة(حرف جر(ل),تركيب إضافي(نكرة(مقاوم,مؤنث),تركيب بدل(اسم اشارة(هذا),معرفة(المرض,مذكر)))),جملة فعلية(فعل(تتبع,متعدي),مفعول به(تركيب إضافي(نكرات متعددة(نكرة(نفس,مذكر),نكرات متعددة(نكرة(خطوات,مؤنث),نكرات متعددة(نكرة(مقاومة,مؤنث),نكرة(حشرة,مؤنث)))),معرفة(المن,مذكر)))))]
10-16	[جملة فعلية(فعل يسبقه حرف(لا,ينتقل,لازم),فاعل(معرفة(المرض,مذكر)),مكمل جملة(شبه جملة(حرف جر(إلي),تركيب إضافي(نكرة(حقول,مؤنث),معرفة(القمح,مذكر)))))]
10-17	[جملة فعلية(فعل(يبدأ,متعدي),مفعول به(معرفة(الحصاد,مذكر)),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(في),نكرة(أواخر,مؤنث))),مكمل جملة(تراكيب إضافية معطوفة(تركيب إضافي(نكرة(شهر,مذكر),معرفة_علم(أبريل,مذكر)),حرف وصل(و),تركيب إضافي(نكرة(أوائل,مؤنث),معرفة_علم(مايو,مذكر))))))),جملة فعلية(فعل(يبدأ,متعدي),فاعل(معرفة(الحصاد,مذكر)),مكمل جملة(شبه جملة(حرف جر(في),نكرة(أواخر,مؤنث))),مفعول به(تراكيب إضافية معطوفة(تركيب إضافي(نكرة(شهر,مذكر),معرفة_علم(أبريل,مذكر)),حرف وصل(و),تركيب إضافي(نكرة(أوائل,مؤنث),معرفة_علم(مايو,مذكر)))))]
10-	[جملة فعلية(فعل(يجب,متعدي),مفعول به(جملة إن و أخواتها(حرف إن (إن),خبر(جملة

18	فعلية(فعل(يتم,متعدي),مفعول به(معرفة(الحصاد,مذكر)),مكمل جملة مركب(مكمل جملة(تركيب إضافي(ظرف زمان(بعد),تركيب وصفي(معرفة(النضج,مذكر),معرفة(التام,مذكر))))),مكمل جملة(نكرة(مباشرة,مؤنث)))))))]
10-19	[جملة فعلية(فعل(يتميز,لازم),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ب),تركيب إضافي(نكرة(اصفرار,مذكر),تركيب وصفي(معرفة(السلامية,مؤنث),تركيب وصفي(معرفة(العليا,مؤنث),معرفة(الحاملة,مؤنث)))))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ل),معرفة(سنبلة,مؤنث))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(في),تركيب إضافي(ظرف زمان(حوالي),تركيب إضافي(نسبة مئوية(50%))))),مكمل جملة(شبه جملة(حرف جر(من),تركيب إضافي(نكرة(نباتات,مؤنث),معرفة(الحقل,مذكر)))))))]
10-20	UnParseable
10-21	[جملة فعلية(فعل يسبقه حرف(لا,يحدث,متعدي),مفعول به(نكرة(فرط,مذكر)),مكمل جملة معطوفة(مكمل جملة(شبه جملة(حرف جر(ل),معرفة(حبوب,مؤنث))),حرف وصل(أو),مكمل جملة مركب(مكمل جملة(نكرة(تكسير,مذكر)),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(ل),معرفة(سنابل,مؤنث))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(مع),تركيب إضافي(نكرة(تقليل,مذكر),معرفة(الفاقد,مذكر)))),مكمل جملة مركب(مكمل جملة(شبه جملة(حرف جر(بال),معرفة(عناية,مؤنث))),مكمل جملة(شبه جملة(حرف جر(ب),تراكيب إضافية معطوفة(تركيب إضافي(نكرة(عمل,مؤنث),معرفة(النقل,مذكر)),حرف وصل(و),معرفة(الدراس,مذكر)))))))))]

APPENDIX (C)
THE ARABIC GRAMMAR

Adjective Constituent (التركيب الوصفي)

We did classify the adjective constituent into two classes:

(1) Simple:

Also this class divided into two subclasses:

(1-1) Defined adjective constituent (تركيب وصفي معرفة), which has the following forms:

<Name: Defined adjective constituent>
<Form1>:
defined+ defined
<Conditions >:

- Last defined noun must be has ability to adjective in its morph indicators.
- Gender of two nouns must be agreed.

<Examples>:
" الفاكهة الحمضية"
"المكتبة العربية"
"الخليج العربي"

<Name: Defined adjective constituent>
<Form2>:
defined+ connected pronoun + defined
<Conditions >:

- Last defined noun must be has ability to adjective in its morph indicators.
- Gender of two nouns must be agreed.

<Examples>:
"مركزه الاقتصادي"

<Name: Defined adjective constituent>
<Form3>:
defined+ <Defined adjective constituent>
<Conditions >:

- Last defined noun (in second constituent) must be has ability to adjective in its morph indicators.
- Gender of two (noun, and second constituent) must be agreed.

<Examples>:
"الإمبراطورية العربية الإسلامية"

(1-2) Indeterminate adjective constituent (تركيب وصفي نكرة), which has the following forms:

<Name: Indeterminate adjective constituent>
<Form1>:
indeterminate + indeterminate
<Conditions >:

- Last indeterminate noun must be has ability to adjective in its morph indicators.
- Morphological pattern of first indeterminate noun , must be not equal to "أفعل".
- Gender of two nouns must be agreed.

<Examples>:
"عرض واضح"
"صفة عامة"
"أكثر شمولا" false, because second condition failed

<Name: Indeterminate adjective constituent>
<Form2>:
indeterminate + <Indeterminate adjective constituent >
<Conditions >:

- Last indeterminate noun (in second constituent) must be has ability to adjective in its morph indicators.
- Morphological pattern of first indeterminate noun , must be not equal to "أفعل".
- Gender of two nouns must be agreed.

<Examples>:
"أرض طينية خصبة"

(2) Connected:

<Name: Connected adjective constituent>
<Form1>:
<Indeterminate adjective constituent> + connector +
<Indeterminate adjective constituent >
Or:
< Defined adjective constituent > + connector + < Defined adjective constituent >

<Conditions >:

- Gender of two constituents must be agreed.

<Examples>:
"أرض طينية و خصبة غنية "

<Name: Connected adjective constituent>
<Form2>:
< Defined adjective constituent > + connector + defined
<Conditions >:
- Last defined noun must be has ability to adjective in its morph indicators.
- Gender of two units (constituent and defined noun) must be agreed.

<Examples>:
"الأراضي الصحراوية أو الرملية"

<Name: Connected adjective constituent>
<Form3>:
< Defined adjective constituent > + connector + <Connected Defines>
<Conditions >:
- Gender of two constituents must be agreed.

<Examples>:
"الأراضي الصحراوية أو الرملية أو الجيرية"

<Name: Connected adjective constituent>
<Form4>:
defined + connector + < Defined adjective constituent >
<Conditions >:
- Gender of two (defined noun and constituent) must be agreed.

<Examples>:
"البذور و المحاصيل الشتوية"

Annexation Constituent (تركيب إضافى):

We did classify the annexation constituent into two classes:

(1) Simple:

Also this class divided into four subclasses:

(1-1) Annexation-pronoun constituent, who has the following forms:

<Name: Annexation-pronoun constituent>
<Form1>:
< Compound indeterminate constituent > + connected pronoun
<Conditions >:
- No conditions.
<Examples>:
"إنتاج و تجميعها"

<Name: Annexation-pronoun constituent>
<Form2>:

< Contiguous indeterminate constituent > + connected pronoun

<Conditions >:

- No conditions.

<Examples>:

"خصوبة أرضها"

<Name: Annexation-pronoun constituent>

<Form3>:

indeterminate + connected pronoun

<Conditions >:

- No conditions.

<Examples>:

"تربتها"

<Name: Annexation-pronoun constituent>

<Form4>:

<Annexation-pronoun constituent> + <Defined adjective constituent>

<Conditions >:

- No conditions.

<Examples>:

"تربتها الطينية الخصبة"

<Name: Annexation-pronoun constituent>

<Form5>:

<Annexation-pronoun constituent> + defined

<Conditions >:

- defined noun must be has ability to adjective in its morph indicators.

<Examples>:

"تربتها الطينية"

(1-2) Annexation-digit constituent, who has the following forms:

<Name: Annexation-digit constituent>

<Form1>:

<digit constituent> + <Contiguous indeterminate constituent>

<Conditions >:

- No conditions.

<Examples>:

"30 كيلو جرام تقاوي"

<Name: Annexation-digit constituent>

<Form2>:

<digit constituent> + indeterminate

<Conditions >:

- No conditions.

<Examples>:

"12 سم"

<Name: Annexation-digit constituent>

<Form3>:

<percentage digit constituent>

<Conditions >:

- No conditions.

<Examples>:

" % 30"

(1-3) Annexation-accusative constituent, who has the following forms:

<Name: Annexation-accusative constituent>

<Form1>:

accusative (of time or place) + connected pronoun

<Conditions >:

- No conditions.

<Examples>:

"قبلها"

<Name: Annexation-accusative constituent>

<Form2>:

accusative (of time or place) + <adjective constituent>

<Conditions >:

- No conditions.

<Examples>:

"بعد الزراعة الشتوية"

<Name: Annexation-accusative constituent>

<Form3>:

accusative (of time or place) + demonstrative

<Conditions >:

- No conditions.

<Examples>:

"قبل هذا"

<Name: Annexation-accusative constituent>

<Form4>:

accusative (of time or place) + defined

<Conditions >:

- No conditions.

<Examples>:

"قبل الري"

<Name: Annexation-accusative constituent>
<Form5>:
accusative (of time or place) + indeterminate
<Conditions >:
- No conditions.
<Examples>:
"بعد الرش"

(1-4) Annexation-normal constituent, who has the following forms:
<Name: Annexation-normal constituent>
<Form1>:
<Compound indeterminate >+ defined
<Conditions >:
- No conditions.
<Examples>:
"زراعة و حرث الأرض"

<Name: Annexation-normal constituent>
<Form2>:
<Compound indeterminate >+ <Connected defined>
<Conditions >:
- No conditions.
<Examples>:
"زراعة و حرث الأرض و التربة"

<Name: Annexation-normal constituent>
<Form3>:
<Contiguous indeterminate >+ <Defined adjective constituent>
<Conditions >:
- No conditions.
<Examples>:
"باقي أنواع الزراعات الشتوية"

<Name: Annexation-normal constituent>
<Form4>:
<Contiguous indeterminate >+ defined
<Conditions >:
- No conditions.
<Examples>:
"ورود ذكر البرتقال"
"استعمال عصير الليمون"

<Name: Annexation-normal constituent>
<Form5>:
indeterminate + <Defined adjective constituent>

<Conditions >:
- No conditions.
<Examples>:
"انتشار الموالح التدريجي"

<Name: Annexation-normal constituent>
<Form6>:
indeterminate + <Substitution Constituent>
<Conditions >:
- No conditions.
<Examples>:
"صفحات هذه الطبعة"
<Name: Annexation-normal constituent>
<Form6>:
indeterminate + <Indeterminate adjective constituent>
<Conditions >:
- No conditions.
<Examples>:
"قمح أراضي صحراوية"

<Name: Annexation-normal constituent>
<Form7>:
<contiguous indeterminate>
<Conditions >:
- No conditions.
<Examples>:
"ظروف غير ملائمة"

<Name: Annexation-normal constituent>
<Form8>:
indeterminate + defined
<Conditions >:
- No conditions.
<Examples>:
"أنواع الموالح"
"عالية المحصول"

(2) Connected:

<Name: Connected Annexation constituent >
<Form1>:
<Simple Annexation constituent > + connector + <Connected Defines>
<Conditions >:
- No conditions.
<Examples>:
"زراعة الأرضي الرملية و الجيرية"

<Name: Connected Annexation constituent >
<Form2>:
<Simple Annexation constituent > + connector +
<Connected Annexation constituent>
<Conditions >:
- No conditions.
<Examples>:
"ورود ذكر و وصف الأراضي الرملية"

<Name: Connected Annexation constituent >
<Form3>:
<Simple Annexation constituent > + connector + <Simple Annexation constituent >
<Conditions >:
- No conditions.
<Examples>:
"ورود ذكر و وصف أنواع الموالح"

<Name: Connected Annexation constituent >
<Form4>:
<Simple Annexation constituent > + connector + defined + connector + <Simple Annexation constituent >
<Conditions >:
- No conditions.
<Examples>:
"جمع الثمار و البذور و تصديرها"

<Name: Connected Annexation constituent >
<Form5>:
<Simple Annexation constituent > + connector + defined
<Conditions >:
- No conditions.
<Examples>:
"جمع الثمار و البذور"
<Name: Connected Annexation constituent >
<Form6>:
<Simple Annexation constituent > + connector + <Defined adjective constituent>
<Conditions >:
- No conditions.
<Examples>:
"نواحي الإنتاج و الزراعة الحديثة"

<Name: Connected Annexation constituent >
<Form7>:

<Simple Annexation constituent > + connector + <indeterminate adjective constituent>

<Conditions >:

- No conditions.

<Examples>:

"استعمال بذور و ثمار غنية"

Substitution Constituent (تركيب بدل):

This constituent was declared only with two rules in our grammar as the following:

<Name: substitution constituent >
<Form1>:
demonstrative pronoun + <Defined adjective constituent>
<Conditions >:

- Gender of two (demonstrative pronoun and second constituent) must be agreed.

<Examples>:

"هذا البرتقال الكبير"
"هذه الطريقة الحديثة"

<Name: substitution constituent >
<Form2>:
demonstrative pronoun + defined
<Conditions >:

- Gender of two (demonstrative pronoun and defined noun) must be agreed.

<Examples>:

"هذا الرجل"
"هذه الطريقة"

Distinguish Constituent (تركيب تمييزى):

This constituent was declared only with two rules in our grammar as the following:

<Name: distinguish constituent >
<Form1>:
indeterminate + <Annexation constituent>
<Conditions >:
- Morphological pattern of first indeterminate noun , must be equal to "أفعل".
<Examples>:

"أكثر بذور الموالح"
"أطول جذور النباتات"

<Name: distinguish constituent >

<Form2>:
indeterminate + indeterminate
<Conditions >:
- Morphological pattern of first indeterminate noun , must be equal to "أفعل".
<Examples>:
"أكثر شمولا"

Inchoative (المبتدأ):

This constituent was declared with seven rules in our grammar as the following:

<Name: inchoative constituent >
<Form1>:
<Defined adjective constituent>
<Conditions >:
- No conditions.
<Examples>:
"الزراعة الحديثة"
"القمح الجديد"

<Name: inchoative constituent >
<Form2>:
<Annexation constituent>
<Conditions >:
- First noun of annexation constituent must be not belonging to accusative of place or accusative of time.
<Examples>:
"زراعة القمح"
"بذور النارنج"
"مصادرها"
"مركزه الاقتصادي"
"بين الأوراق" not accepted

<Name: inchoative constituent >
<Form3>:
<distinguish constituent>
<Conditions >:
- No conditions.
<Examples>:
"أكثر زراعة"
"أفضل طريقة"

<Name: inchoative constituent >
<Form4>:
<substitution constituent>

<Conditions >:
- No conditions.
<Examples>:
"هذه الزراعة"
"هذا القمح"

<Name: inchoative constituent >
<Form5>:
defined.
<Conditions >:
- No conditions.
<Examples>:
"الزراعة"
"القمح"

<Name: inchoative constituent >
<Form6>:
separated pronoun.
<Conditions >:
- No conditions.
<Examples>:
"هو"
"هي"

<Name: inchoative constituent >
<Form7>:
demonstrative pronoun.
<Conditions >:
- No conditions.
<Examples>:
"هذا"
"هذه"

Enunciative (الخبر):

We did classify the enunciative constituent into four classes:

(1) Simple:

Also this class divided into two classes:

(1-1) Default simple enunciative without any violation of default rules of enunciative ,

The following are the declaration of the six rules represent it:

<Name: simple enunciative constituent >
<Form1>:
quasi-proposition

<Conditions >:
- No conditions.
<Examples>:
"في البيت"
"في الأرض الرملية"
"في الأراضي الجديدة"

<Name: simple enunciative constituent >
<Form2>:
<indeterminate adjective constituent>
<Conditions >:
- No conditions.
<Examples>:
"زراعة اقتصادية"
"ثمرة صغيرة"
"زراعة مكلفة"

<Name: simple enunciative constituent >
<Form3>:
<annexation constituent>
<Conditions >:
- No conditions.
<Examples>:

"سريع الإنتاج"
"زراعة حديثة"

<Name: simple enunciative constituent >
<Form4>:
<distinguish constituent>
<Conditions >:
- No conditions.
<Examples>:
"أكثر شمولا"
"أفضل زراعة"
"أحدث طريقة"

<Name: simple enunciative constituent >
<Form5>:
indeterminate + <conjunctive sentence>
<Conditions >:
- Gender of two (indeterminate noun and conjunctive noun) must be agreed.
<Examples>:
"صنف الذي يزرع في الأراضي الرملية"

"صنف التي يزرع في الأراضي الرملية"

<Name: simple enunciative constituent >
<Form6>:
indeterminate
<Conditions >:
- No conditions.
<Examples>:
"جميل"
"حديث"
"اقتصادي"

(1-2) free simple enunciative which (under some special cases) will violate default rules of enunciative , The following are the declaration of the two rules represent it:

<Name: simple enunciative constituent >
<Form1>:
defined
<Conditions >:
- The nominal sentence which contain this enunciative , must call this type of simple enunciative class explicate.
<Examples>:
from the following large sentence "هو الصحراء"
"موطنها الأصلي هو الصحراء"
Note: in this example, we violate two of default rules of the enunciative
(1) First, there is no gender agreement between the inchoative "هو" Which is "male", and its enunciative "الصحراء" which is "female".
(2) Second, the violation of the linguistic rule of that, the enunciative must be indeterminate (see chapter 3).

<Name: simple enunciative constituent >
<Form2>:
<indeterminate adjective constituent>
<Conditions >:
- The nominal sentence which contain this enunciative, must call this type of simple enunciative class explicate.
<Examples>:
from the following large sentence "هو المنطقة الاستوائية"
"موطنها الأصلي هو المنطقة الاستوائية"
Note: the same previous notes.

(2) Enunciative as sentence:

The following are the two-declaration rules of this class:

<Name: enunciative as sentence constituent >
<Form1>:
<Nominal sentence>
<Conditions >:

- The large nominal sentence which contains this embedded nominal sentence as enunciative, must call this type of enunciative class explicate.
- The inchoative of the embedded nominal sentence must be containing separated or connected pronoun.

<Examples>:
enunciative for the following large sentence "زراعته سهلة"
"القمح زراعته سهلة"

<Name: enunciative as sentence constituent >
<Form2>:
<Verbal sentence>
<Conditions >:

- The large nominal sentence which contains this embedded verbal sentence as enunciative, must call this type of enunciative class explicate.
- The embedded verbal sentence must be missing its Subject.

<Examples>:
enunciative for the following large sentence "يزرع الأرض"
"الفلاح يزرع الأرض"

(3) Enunciative with complementary:

The following are the rules whose declare these class forms:

<Name: enunciative with complementary constituent >
<Form1>:
<simple enunciative constituent> + <Complementary>
<Conditions >:

- No conditions.

<Examples>:
"القمح بخير في البلاد العربية "
in this sentence , the grammar will recognize the constituent "بخير"
as the enunciative of the sentence, but the following constituent " في البلاد العربية"
will recognized as complementary of the sentence.

<Name: enunciative with complementary constituent >
<Form2>:

<simple enunciative constituent> + <enunciative as sentence constituent> + <Complementary>

<Conditions >:

- No conditions.

<Examples>:

"القمح زراعته قليلة في البلاد العربية "

in this sentence , the grammar will recognize the sentence " زراعته قليلة"

as the enunciative of the sentence, but the following constituent " في البلاد العربية"

will recognized as complementary of the sentence.

(4) Compound enunciative:

<Name: compound enunciative constituent >

<Form1>:

<Complementary> + <enunciative as sentence constituent>

<Conditions >:

- No conditions.

<Examples>:

"القمح في الدول العربية يزرع في الأراضي الرملية"

In this sentence, our grammar will recognize the constituent "في الدول العربية" as "Complementary constituent" , then the grammar will continue to recognize the verbal sentence " يزرع في الأراضي الرملية" as the enunciative of the sentence.

"القمح في الدول العربية زراعته قليلة"

In this sentence, our grammar will recognize the constituent "في الدول العربية" as "Complementary constituent" , then the grammar will continue to recognize the nominal sentence " زراعته قليلة" as the enunciative of the sentence.

Quasi-proposition (شبه جملة):

The grammar declare this constituent in the following six forms:

<Name: Quasi-proposition constituent>

<Form1>:

preposition + <Defined adjective constituent>

Or

preposition + <indeterminate adjective constituent>

<Conditions >:

- No conditions.

<Examples>:

"في الدول العربية"
"من أنواع هامة"

<Name: Quasi-proposition constituent>
<Form2>:
preposition + <annexation constituent>
<Conditions >:
- No conditions.
<Examples>:
"في بلاد العالم"
"من ثمار الموالح"

<Name: Quasi-proposition constituent>
<Form3>:
preposition + < substitution constituent>
<Conditions >:
- No conditions.
<Examples>:
"في هذا الوطن"
"من هذه الثمرة"

<Name: Quasi-proposition constituent>
<Form4>:
preposition + connected pronoun
<Conditions >:
- No conditions.
<Examples>:
"منها"
"فيه"
"إليهم"

<Name: Quasi-proposition constituent>
<Form5>:
preposition + <noun> + connector + <noun>
<Conditions>:
- The determination of the two nouns must be agreed.
<Examples>:
"على التربة و السطح"
"في ورق و سيقان"
not accepted "في التربة و سيقان"

<Name: Quasi-proposition constituent>
<Form6>:
preposition + <noun>
<Conditions>:

- No conditions.

<Examples>:

"في الأرض"

"من الثمار"

Subject (فاعل):

The grammar classified the Subject constituent into two classes as the followed:

(1) Simple:

These simple Subject classes declared into five forms:

(1-1)

<Name: simple Subject>

<Form1>:

<annexation constituent>

<Conditions>:

- First noun of annexation constituent must be not belonging to accusative of place or accusative of time.

<Examples>:

"زراعة القمح"

"بذور النارنج"

not accepted "تحت التربة"

(1-2)

<Name: simple Subject>

<Form2>:

<Substitution Constituent>

<Conditions>:

- no conditions.

<Examples>:

"هذه الزراعة"

"هذا القمح"

(1-3)

<Name: simple Subject>

<Form3>:

<Defined adjective constituent>

or

<indeterminate adjective Constituent>

<Conditions>:

- no conditions.

<Examples>:

"الرية الشتوية "

"مكافحة قوية"

(1-4)

<Name: simple Subject>
<Form4>:
defined
<Conditions>:
- no conditions.
<Examples>:
"القمح"
"البيت"
"التربة"

(1-5)
<Name: simple Subject>
<Form5>:
indeterminate
<Conditions>:
- no conditions.
<Examples>:
"قمح"
"بيت"
"تربة"

(2) Connected:

<Name: Connected Subject>
<Form1>:
defined + connector + defined
<Conditions>:
- no conditions.
<Examples>:
"العرب و الرومان"
"القمح و الموالح"

<Name: Connected Subject>
<Form2>:
defined + connector + <Connected Subject>
<Conditions>:
- no conditions.
<Examples>:
"العرب و الأسبان و الرومان"
"القمح و الموالح و البذور"

Direct Object (مفعول به):

The grammar declared this constituent into the following forms:
<Name: direct object>
<Form1>:

<’Inna sentence>
<Conditions>:
- no conditions.
<Examples>:
direct object in the following large sentence "أن المنغوليين اكتشفوا الليمون"
"أوضح تاناكا أن المنغوليين اكتشفوا الليمون"

<Name: direct object>
<Form2>:
<annexation constituent>
<Conditions>:
- no conditions.
<Examples>:
direct object in the following large sentence "طرق تكاثرها"
"ذكر طرق تكاثرها"

direct object in the following large sentence "كتاب هسبريدزا"
"نشر كتاب هسبريدزا"

<Name: direct object>
<Form3>:
<defined adjective Constituent>
or
<indeterminate adjective Constituent>
<Conditions>:
- no conditions.
<Examples>:
direct object in the following large sentence "عرض واضح"
"شملت صفحات هذه الطبعة عرض واضح"

direct object in the following large sentence "المناطق المعتدلة الدافئة"
"تشمل هذه المنطقة المناطق المعتدلة الدافئة"

<Name: direct object>
<Form4>:
<distinguish constituent>
<Conditions>:
- no conditions.
<Examples>:
direct object in the following large sentence "أفضل نتائج"
"أوضحت الدراسات أفضل نتائج"

<Name: direct object>

<Form5>:
<substitution constituent>
<Conditions>:
- no conditions.
<Examples>:
direct object in the following large sentence "هذا النوع"
"زرع الصينيون هذا النوع"

<Name: direct object>
<Form6>:
<Connected Defines>
<Conditions>:
- no conditions.
<Examples>:
direct object in the following large sentence "البرتقال و اليوسفي"
"خصوا بالذكر البرتقال و اليوسفي"

<Name: direct object>
<Form7>:
defined
<Conditions>:
- no conditions.
<Examples>:
direct object in the following large sentence "الأرض"
"تحرث الأرض"

<Name: direct object>
<Form8>:
indeterminate
<Conditions>:
- no conditions.
<Examples>:
direct object in the following large sentence "بقع"
"يظهر هذا المرض بقع عل الأوراق"

Complement (مكمل جملة):

The grammar divide this constituent into two classes as follow:

(1) Unit:

The declarations of this class include four forms:

<Name: unit complementary>
<Form1>:
<compound complementary>
<Conditions>:
- no conditions.
<Examples>:
compound complementary in the following large sentence "في كثير من الدول الغربية"

"ينتشر في كثير من الدول الغربية"

compound complementary in the following large sentence "من الدراسات المطروحة للبحث"

"تظهر نتائجه من الدراسات المطروحة للبحث"

<Name: unit complementary>
<Form2>:
<simple complementary>
<Conditions>:
- no conditions.
<Examples>:
simple complementary in the following large sentence "مؤخرا"

"اكتشفوا الليمون مؤخرا"

(2) Connected:

<Name: connected complementary>
<Form1>:
<unit complementary> + connector + <unit complementary>
<Conditions>:

1- If first unit complementary was " 'Inna sentence" , then reject this form.

2- If first unit complementary include "annexation constituent " in the end of it,
Then , second unit complementary must not include "annexation constituent " in
The start of it.

<Examples>:

"في البلاد العربية و في جنوب آسيا"

"من أنواع الموالح و أصنافها"

not accepted (condition 2) "خصائص نموها و إنتاجها"

this constituent will not recognize here as "connected complementary" ("إنتاجها" + "و" + "خصائص نموها"), but will recognized as a single "annexation constituent" ("خصائص نموها و إنتاجها") which is include embedded "connected annexation constituents".

not accepted (condition 1) "أن تاريخ الموالح من المواضيع المهمة و عظيمة الأهمية"

this constituent will not recognize here as "connected complementary" ("عظيمة الأهمية" + "و" + "أن تاريخ الموالح من المواضيع المهمة") , but will recognized as a single "'Inna sentence" ("أن تاريخ الموالح من المواضيع المهمة و عظيمة الأهمية") which is include embedded "connected complementary" ("من المواضيع المهمة و عظيمة الأهمية").

Simple Complementary:

The grammar divide this constituent into two classes as follow:

(1-1) This subclass will refer to it by "simple_1 complementary", in this class we define the constituents which has no way to be any thing (Subject or Direct Object) except "complementary".

The forms of it as follow:

<Name: simple_1 complementary>
<Form1>:
preposition + <'Inna sentence>
<Conditions>:
- No conditions.
<Examples>:
"إلي أن الموالح قسم من أشجار الفاكهة"

<Name: simple_1 complementary>
<Form2>:
<Quasi-proposition Constituent>
<Conditions>:
- No conditions.
<Examples>:
"في الدول العربية"

<Name: simple_1 complementary>
<Form3>:
<separated pronoun>
<Conditions>:
- No conditions.
<Examples>:
occurred in the following large sentence "...هي....."
"تعتبر الزراعة العفير **هي** أكثر طرق الزراعة استخداما"
The grammar will recognize this "separated pronoun" as a complementary
of the verbal sentence, and call it as "ضمير شأن للتوكيد" .

<Name: simple_1 complementary>
<Form4>:
<annexation constituent>
<Conditions>:
- First noun of annexation constituent must belong to accusative of place or accusative of time.
<Examples>:
"تحت التربة"
"فوق الرمال"
"خصوبة الأرض" not accepted

<Name: simple_1 complementary>
<Form5>:
<Defined adjective constituent>
Or
<indeterminate adjective constituent>
<Conditions>:
- The adjective constituent must has explicate subjunctive or genitive case (علامة نصب أو جر ظاهرة).
<Examples>:
" ارتفاعا نسبيا" (الألف الأخيرة تمثل علامة نصب)
"شمولا واضحا" (الألف الأخيرة تمثل علامة نصب)
"شمول واضح" not accepted

<Name: simple_1 complementary>
<Form6>:
indeterminate
<Conditions>:
- The indeterminate must have explicated subjunctive or genitive case (جر أو نصب علامة ظاهرة).
<Examples>:
"نسبيا" (الألف الأخيرة تمثل علامة نصب)
"شمولا" (الألف الأخيرة تمثل علامة نصب)
"شمول" not accepted

(1-2) This subclass will refer to it by "simple_2 complementary", in this class we define the constituents whose can be any recognized as (Subject or Direct Object).
The forms of it as follow:

<Name: simple_2 complementary>
<Form1>:
<annexation constituent>
<Conditions>:
- The sentence call this "simple_2 constituent" rule, must be sure from that , the Subject and the Direct Object can't be occur in the reminder of the sentence.
<Examples>:
"....عبر الحقب الزمنية" included in the following large sentence
"انتشر النارنج عبر الحقب الزمنية"

<Name: simple_2 complementary>
<Form2>:
<Defined adjective constituent>
Or
<indeterminate adjective constituent>

<Conditions>:

- The sentence call this “simple_2 constituent” rule, must be sure from that , the Subject and the Direct Object can’t be occur in the reminder of the sentence.

<Examples>:

"صنعوا صوبات حولها زجاجية كبيرة"

<Name: simple_2 complementary>

<Form3>:

<distinguish constituent>

<Conditions>:

- The sentence call this “simple_2 constituent” rule, must be sure from that , the Subject and the Direct Object can’t be occur in the reminder of the sentence.

<Examples>:

included in the following large sentence "…….أفضل تقاوي….."

"تنتج بذور القمح سنابل هي أفضل تقاوي للزراعة مرة أخرى"

<Name: simple_2 complementary>

<Form4>:

indeterminate

<Conditions>:

- The sentence call this “simple_2 constituent” rule, must be sure from that , the Subject and the Direct Object can’t be occur in the reminder of the sentence.

<Examples>:

included in the following large sentence "…..عامة"

"شملت صفحات هذه الطبعة دراسة لأنواع الموالح عامة"

<Name: simple_2 complementary>

<Form5>:

defined

<Conditions>:

- The sentence call this “simple_2 constituent” rule, must be sure from that , the Subject and the Direct Object can’t be occur in the reminder of the sentence.

<Examples>:

""

Compound Complementary:

This constituent has two forms:

<Name: compound complementary>

<Form1>:

<simple complementary> + <simple complementary>

<Conditions>:

- No conditions.

<Examples>:

"في الدول العربية من فترات بعيدة"
"كبيرة نسبيا في حجمها"

<Name: compound complementary>
<Form2>:
<simple complementary> + <compound complementary>
<Conditions>:
- No conditions.
<Examples>:

"في الدول العربية من الشرق إلي الغرب"

Digit Constituent (تركيب رقمي):

This constituent has two forms:

<Name: digit constituent>
<Form1>:
<percentage constituent>
<Conditions>:
- No conditions.
<Examples>:

"33% آزوت"
"% 20"

<Name: digit constituent>
<Form2>:
<simple digit constituent>
<Conditions>:
- No conditions.
<Examples>:

"15"
"15 إلي 30"

Simple Digit Constituent:

This constituent has two forms:

<Name: simple digit constituent>
<Form1>:
digit
<Conditions>:
- No conditions.
<Examples>:

"14"
"200"

<Name: simple digit constituent>
<Form2>:

digit + preposition + digit

<Conditions>:

- No conditions.

<Examples>:

"14 إلي 50"

Percentage Digit Constituent (تركيب رقم نسبى):

This constituent has two forms:

<Name: simple digit constituent>

<Form1>:

percentage digit

<Conditions>:

- No conditions.

<Examples>:

"%14"

"%200"

<Name: simple digit constituent>

<Form2>:

percentage digit + preposition + percentage digit

<Conditions>:

- No conditions.

<Examples>:

"%14 إلي %50"

Grammar of Sentences:

The proposed grammar recognize the following sentences:

Simple Sentences :

We assume that the simple sentence in the grammar will classified into three classes:

Nominal Sentences (جملة اسمية):

Nominal sentence declared in the following forms:

<Name: Nominal sentence>

<Form1>:

<Inchoative> + <Enunciative>

<Conditions>:

- Number of two both Inchoative and Enunciative must be agreed.

<Examples>:

"زراعة القمح جيدة"

"إنتاج الفاكهة الحمضية رغم أهميته البالغة لايزال من الموضوعات المهمة"

<Name: Nominal sentence>

<Form2>:

<Quasi-proposition Constituent> + <Inchoative>

<Conditions>:

- No conditions.

<Examples>:

"من المهم اتباع التوصيات الفنية"

"من الضروري العمل"

<Name: Nominal sentence>

<Form3>:

<Quasi-proposition Constituent> + <Inchoative> + <complementary>

<Conditions>:

- No conditions.

<Examples>:

"من الضروري العمل على مقاومة الحشائش"

"من المهم اتباع التوصيات الفنية الخاصة بإنتاج القمح"

Verbal Sentences (جملة فعلية):

(1) Simple Verbal Sentence:

The simple verbal sentence has the following forms in our grammar:

<Name: Simple Verbal sentence>

<Form1>:

verb + <Subject> + <complementary> + <direct object> + <complementary>

<Conditions>:

1- Gender of Subject must be agree with the gender indicator of the verb.

2- Verb doesn't contain morph indicator of Direct Object.

3- Verb doesn't contain morph indicator of Subject.

4- Verb must be transitive.

<Examples>:

<Name: Simple Verbal sentence>

<Form2>:

verb + <Subject> + <direct object> + <complementary>

<Conditions>:

1- Gender of Subject must be agree with the gender indicator of the verb.

2- Verb doesn't contain morph indicator of Direct Object.

3- Verb doesn't contain morph indicator of Subject.

4- Verb must be transitive.

<Examples>:

<Name: Simple Verbal sentence>

<Form3>:

verb + <Subject> + <complementary> + <direct object>

<Conditions>:

1- Gender of Subject must be agree with the gender indicator of the verb.

2- Verb doesn't contain morph indicator of Direct Object.

3- Verb doesn't contain morph indicator of Subject.

4- Verb must be transitive.

<Examples>:

<Name: Simple Verbal sentence>

<Form4>:

verb + <complementary> + <Subject> + <complementary>

<Conditions>:

1- Gender of Subject must be agree with the gender indicator of the verb.

2- Verb doesn't contain morph indicator of Subject.

3- Verb must be intransitive.

<Examples>:

<Name: Simple Verbal sentence>

<Form5>:

verb + <Subject> + <direct object>

<Conditions>:

1- Gender of Subject must be agree with the gender indicator of the verb.

2- Verb doesn't contain morph indicator of Direct Object.

3- Verb doesn't contain morph indicator of Subject.

4- Verb must be transitive.

<Examples>:

<Name: Simple Verbal sentence>

<Form6>:

verb + <Subject> + <complementary>

<Conditions>:

1- Gender of Subject must be agreed with the gender indicator of the verb.

2- Verb doesn't contain morph indicator of Subject.

3- Verb must be intransitive.

<Examples>:

<Name: Simple Verbal sentence>
<Form7>:
verb + <complementary> + <Subject>
<Conditions>:
1- Gender of Subject must be agreed with the gender indicator of the verb.
2- Verb doesn't contain morph indicator of Subject.
3- Verb must be intransitive.
<Examples>:

<Name: Simple Verbal sentence>
<Form8>:
verb + <direct object> + <complementary>
<Conditions>:
1- Verb doesn't contain morph indicator of Direct Object.
2- Verb must be transitive.
<Examples>:

<Name: Simple Verbal sentence>
<Form9>:
verb + <complementary> + <direct object>
<Conditions>:
1- Verb doesn't contain morph indicator of Direct Object.
2- Verb must be transitive.
<Examples>:

<Name: Simple Verbal sentence>
<Form10>:
verb + <complementary> + <direct object> + <complementary>
<Conditions>:
1- Verb doesn't contain morph indicator of Direct Object.
2- Verb must be transitive.
<Examples>:

<Name: Simple Verbal sentence>
<Form11>:
verb + <complementary>
<Conditions>:
- No conditions.
<Examples>:

<Name: Simple Verbal sentence>

<Form12>:
verb + <Subject>
<Conditions>:
1- Gender of Subject must be agreed with the gender indicator of the verb.
2- Verb must contain morph indicator of Direct Object.
3- Verb doesn't contain morph indicator of Subject.
<Examples>:

<Name: Simple Verbal sentence>
<Form13>:
verb + <direct object>
<Conditions>:
1- Gender of Subject must be agreed with the gender indicator of the verb.
2- Verb doesn't contain morph indicator of Direct Object.
3- Verb doesn't contain morph indicator of Subject.
4- Verb must be transitive.
<Examples>:

(2) Prefixed Verbal Sentence:

<Name: Prefixed Verbal sentence>
<Form1>:
<complementary> + <Simple Verbal sentence>
<Conditions>:
- No conditions.
<Examples>:
"فيها تبذر التقاوي عقب حرث و خدمة الأرض"
"نظرا لأهمية محصول القمح باعتباره المحصول الغذائي الأول في مصر فإنه يعتمد على تلك الأراضي في إنتاج القمح لتقليل الفجوة بين المنتج المحلي و المستهلك من حبوب القمح"

Special Verbal Sentences (جملة فعلية خاصة):

(1) 'Inna Sentence (جملة إن و أخواتها):

We assume that the 'Inna sentence has the following forms:
<Name: 'Inna sentence>
<Form1>:
verb 'Inna + connected pronoun + <Enunciative>
<Conditions>:
- No conditions.
<Examples>:

"أنه من الموالح"
"أنه يوجد في بعض المقاطعات"

<Name: ‘Inna sentence>
<Form2>:
verb ‘Inna + <Nominal sentence>
<Conditions>:
- No conditions.
<Examples>:
"أن الموالح بصفة عامة نشأت في هذه المنطقة الواسعة"
"إن تاريخ الموالح و قصة انتشارها من المواضيع المهمة"

<Name: ‘Inna sentence>
<Form3>:
verb ‘Inna + <Enunciative>
<Conditions>:
- No conditions.
<Examples>:
"أن يثمر في الشتاء"

(2) Kana Sentence (جملة كان و أخواتها):

We assume that the ‘Inna sentence has the following forms:

<Name: Kana sentence>
<Form1>:
verb Kana + connected pronoun + <Enunciative>
<Conditions>:
- No conditions.
<Examples>:
""

<Name: Kana sentence>
<Form2>:
verb Kana + <Nominal sentence>
<Conditions>:
- No conditions.
<Examples>:
"لا يزال مرض الأسقربوط يظهر بأعراض طفيفة"

<Name: Kana sentence>
<Form3>:
verb Kana + <Enunciative>
<Conditions>:
- No conditions.
<Examples>:
"لتكون أكثر شمولا"
"لا يزال من الموضوعات المهمة"

Compound Sentences (جملة مركبة):

We declare a compound sentence in our grammar in two forms:

<Name: Compound sentence>
<Form1>:
<Nominal sentence> + connector + <Nominal sentence>
<Conditions>:
- No conditions.
<Examples>:
"انتظام توزيع التقاوي في الحقل و انتظام عمق الزراعة و ضمان تغطية الحبوب عقب الزراعة"

<Name: Compound sentence>
<Form2>:
<Verbal sentence> + connector + <Prefixed Verbal sentence>
<Conditions>:
- No conditions.
<Examples>:
"تعتبر الزراعة العفير بدار هي أكثر طرق الزراعة استخداما و فيها تبذر التقاوي عقب حرث و خدمة الأرض ."

<Name: Compound sentence>
<Form3>:
<Nominal sentence> + connector + <Compound sentence>
<Conditions>:
- No conditions.
<Examples>:
This form declared for recursive example of compound sentence.

APPENDIX(D)
THE PROLOG CODE

```
/*********************************************************************
 *
 *
 *      [Word, Type, Sex, Number, Known, Function, Trans, VerbSepcial, NounSpecial, Wazn,
 *      AdVerb, Modaf, VerbPrefix, VerbSuffix].
 *
 *      Function ==> علامة اعراب
 *  AdVerb      ==> القابلية النعتية
 *
 *********************************************************************/


/*******************************************
 *
 *                           *SENTENCE*
 *
 *******************************************/

s(Tree) --> simple_sentence(Tree,_NextConnectionDegree).
s(Tree) --> compund_sentence(Tree).

simple_sentence(Tree,NextConnectionDegree)--> gomla_essmya(Tree,_, 'no', 'yes',
_LastUnitType,  _FirstUnitType,NextConnectionDegree).
simple_sentence(Tree,NextConnectionDegree)--> gomla_feealia(Tree,NextConnectionDegree,
'yes', 'no').

compund_sentence('جملة مركبة'Tree1,RG,Tree2 <-- ((
       gomla_essmya(Tree1,_, 'no', 'yes', _LastUnitType1,
_FirstUnitType1,NextConnectionDegree1),
       rabet_goml(RG,NextConnectionDegree1),
       gomla_essmya(Tree2,_, 'no', 'yes', _LastUnitType2,
_FirstUnitType2,_NextConnectionDegree2).

compund_sentence('جملة مركبة'Tree1,RG,Tree2<-- ((
       gomla_essmya(Tree1,_, 'no', 'yes', _LastUnitType1,
_FirstUnitType1,NextConnectionDegree1),
       rabet_goml(RG,NextConnectionDegree1),
       compund_sentence(Tree2).

/*******************************************
 *
 *                           *GOMLA_ESSMYA*
 *
 *******************************************/
gomla_essmya_simple('جملة أسمية'TreeMob, TreeKhbr), Mobtadaa_Contain_Dameer, KhabarFree,
Agreement, LastUnitType2,  FirstUnitType1,NextConnectionDegree<--(
       mobtadaa(TreeMob, _SEX , NUM , _, _, _,_,_, Mobtadaa_Contain_Dameer,
_LastUnitType1,  FirstUnitType1),
              {validate_KhabarFree(KhabarFree,FirstUnitType1)},
       khabar(TreeKhbr,  _SEX1 , NUM1 , _, _, _,_,_, KhabarFree, LastUnitType2,
_FirstUnitType2,NextConnectionDegree),
              {validate_NumAgreement(Agreement,NUM,NUM1)} ,!.

gomla_essmya_simple('جملة أسمية'TreeKhbr), TreeMob , TreeComplemntry),
Mobtadaa_Contain_Dameer, KhabarFree, _Agreement, LastUnitType3,
FirstUnitType1,NextConnectionDegree)-->('خبر مقدم')
       shepeh_gomla_single_or_compund(TreeKhbr, _SEX1 , _NUM1, _Known, _, _,_,_,
LastUnitType1, FirstUnitType1, 'all' ,_LastTermType),
```

```
       mobtadaa(TreeMob, _SEX , _NUM , _, _, _,_,_, Mobtadaa_Contain_Dameer,
LastUnitType2,  FirstUnitType2),
               {refused_unit_decompos(LastUnitType1, FirstUnitType2)},
               {validate_KhabarFree(KhabarFree,FirstUnitType1)},
       complentry(TreeComplemntry, _,_,_,_,_,_,_, LastUnitType3, FirstUnitType3,'yes' ,
_RefuseFaaelInFollowingParse,NextConnectionDegree),
               {refused_unit_decompos(LastUnitType2, FirstUnitType3)} ,!.

gomla_essmya_simple('خبر مقدم')'جملة أسمية'(TreeKhbr), TreeMob), Mobtadaa_Contain_Dameer,
KhabarFree, _Agreement, LastUnitType2,  FirstUnitType1,NextConnectionDegree)-->
       shepeh_gomla_single_or_compund(TreeKhbr, _SEX1 , _NUM1, _Known, _, _,_,_,
LastUnitType1, FirstUnitType1, 'all' ,_LastTermType),
       mobtadaa(TreeMob, _SEX , _NUM , _, _, _,_,_, Mobtadaa_Contain_Dameer,
LastUnitType2,  FirstUnitType2),
               {refused_unit_decompos(LastUnitType1, FirstUnitType2)},
               {validate_KhabarFree(KhabarFree,FirstUnitType1)},
               {NextConnectionDegree = 'all'}.


gomla_essmya(Tree , Mobtadaa_Contain_Dameer, KhabarFree, Agreement, LastUnitType2,
FirstUnitType1,NextConnectionDegree)-->
       gomla_essmya_simple(Tree,Mobtadaa_Contain_Dameer, KhabarFree, Agreement,
LastUnitType2,  FirstUnitType1,NextConnectionDegree).

/*********************************************
 *
 *                          *GOMLA_FEEALIA*
 *
 *********************************************/

       /*      ============================================        */
       /*      =========      gomla_feealia_simple   =========      */
       /*      ============================================        */

       /*      ================================================================================
       */
       /*      =========      gomla_feealia_simple
               =========      */
       /*      =========      (verb, faael, complentry, mafool_behe, complentry)
       =========      */
       /*      ================================================================================
       */
 gomla_feealia_simple(' جملة
)'فعلية'(TreeVerb,TreeFaael,TreeComplemntry,TreeMafool,TreeComplemntry2),NextConnectionDegr
ee,IsGomlaWithoutFaael<--(
               {IsGomlaWithoutFaael \== 'yes'},
       verb(TreeVerb, _, _, _ , _, VerbTrans,null,_ ,VerbPrefix,
_VerbSuffix,VerbRefuseFaaelInFollowingParse , IsContainMafool),
               {
                       IsContainMafool = 'no',
                       VerbTrans = 'متعدي',
                       VerbRefuseFaaelInFollowingParse = 'no'
               },
       faael(TreeFaael, SexFaael, _ , _ , _Function ,_,_,_, LastUnitType1,
_FirstUnitType1),
               {validate_faael_sex_agreement(VerbPrefix, SexFaael)},
       complentry(TreeComplemntry, _,_,_,_,_,_,_, LastUnitType2, FirstUnitType2,'yes' ,
_RefuseFaaelInFollowingParse,_NextConnectionDegree2),
               {refused_unit_decompos(LastUnitType1, FirstUnitType2)},
       mafool_behe(TreeMafool, _,_,_,_,_,_,_, LastUnitType3, FirstUnitType3),
               {refused_unit_decompos(LastUnitType2, FirstUnitType3)},
       complentry(TreeComplemntry2, _,_,_,_,_,_,_, _LastUnitType4, FirstUnitType4,'yes' ,
_RefuseFaaelInFollowingParse4,NextConnectionDegree),
               {refused_unit_decompos(LastUnitType3, FirstUnitType4)}.

       /*      ============================================================        */
       /*      =========      gomla_feealia_simple
       =========      */
       /*      =========      (verb, faael, mafool_behe,complentry) =========      */
```

```
	/*	================================================================	*/
 gomla_feealia_simple(' جملة
)'فعلية'TreeVerb,TreeFaael,TreeMafool,TreeComplemntry),NextConnectionDegree,IsGomlaWithout
Faael<--(
		{IsGomlaWithoutFaael \== 'yes'},
	verb(TreeVerb, _, _, _ , _, VerbTrans,null,_ ,VerbPrefix,
_VerbSuffix,VerbRefuseFaaelInFollowingParse , IsContainMafool),
		{
			IsContainMafool = 'no',
			VerbTrans = ',متعدي'
			VerbRefuseFaaelInFollowingParse = 'no'
		},
	faael(TreeFaael, SexFaael, _ , _ , _Function ,_,_,_, LastUnitType1,
_FirstUnitType1),
		{validate_faael_sex_agreement(VerbPrefix, SexFaael)},
	mafool_behe(TreeMafool, _,_,_,_,_,_,_, LastUnitType2, FirstUnitType2),
		{refused_unit_decompos(LastUnitType1, FirstUnitType2)},
	complentry(TreeComplemntry, _,_,_,_,_,_,_, _LastUnitType3, FirstUnitType3,'yes' ,
_RefuseFaaelInFollowingParse,NextConnectionDegree),
		{refused_unit_decompos(LastUnitType2, FirstUnitType3)}.

	/*	================================================================	*/
	/*	=========	gomla_feealia_simple
	=========	*/
	/*	=========	(verb, faael, complentry, mafool_behe)	=========
	*/
	/*	================================================================	*/
 gomla_feealia_simple(' جملة
)'فعلية'TreeVerb,TreeFaael,TreeComplemntry,TreeMafool),NextConnectionDegree,IsGomlaWithout
Faael<--(
		{IsGomlaWithoutFaael \== 'yes'},
	verb(TreeVerb, _, _, _ , _, VerbTrans,null,_ ,VerbPrefix,
_VerbSuffix,VerbRefuseFaaelInFollowingParse , IsContainMafool),
		{
			IsContainMafool = 'no',
			VerbTrans = ',متعدي'
			VerbRefuseFaaelInFollowingParse = 'no'
		},
	faael(TreeFaael, SexFaael, _ , _ , _Function ,_,_,_, LastUnitType1,
_FirstUnitType1),
		{validate_faael_sex_agreement(VerbPrefix, SexFaael)},
	complentry(TreeComplemntry, _,_,_,_,_,_,_, LastUnitType2, FirstUnitType2,'yes' ,
_RefuseFaaelInFollowingParse,NextConnectionDegree),
		{refused_unit_decompos(LastUnitType1, FirstUnitType2)},
	mafool_behe(TreeMafool, _,_,_,_,_,_,_, _LastUnitType3, FirstUnitType3),
		{refused_unit_decompos(LastUnitType2, FirstUnitType3)}.

	/*	================================================================	*/
	/*	=========	gomla_feealia_simple
	=========	*/
	/*	=========	(verb, complentry, faael,complentry)	=========	*/
	/*	================================================================	*/
gomla_feealia_simple(' جملة
)'فعلية'TreeVerb,TreeComplemntry1,TreeFaael,TreeComplemntry2),NextConnectionDegree2,IsGoml
aWithoutFaael<--(
		{IsGomlaWithoutFaael \== 'yes'},
	verb(TreeVerb, _, _, _ , _, VerbTrans,null,_ ,VerbPrefix,
_VerbSuffix,VerbRefuseFaaelInFollowingParse , _IsContainMafool),
		{
			VerbTrans = ',لازم'
			VerbRefuseFaaelInFollowingParse = 'no'
		},
	complentry(TreeComplemntry1, _,_,_,_,_,_,_, LastUnitType1, _FirstUnitType1,'no' ,
RefuseFaaelInFollowingParse,_NextConnectionDegree1),
		{RefuseFaaelInFollowingParse = 'no'},
```

```
       faael(TreeFaael, SexFaael, _Num , _Known , _Function ,_,_,_, _LastUnitType2,
FirstUnitType2),
              {validate_faael_sex_agreement(VerbPrefix, SexFaael)},
              {refused_unit_decompos(LastUnitType1, FirstUnitType2)},
       complentry(TreeComplemntry2, _,_,_,_,_,_,_, _LastUnitType3, _FirstUnitType3,'yes'
, _RefuseFaaelInFollowingParse,NextConnectionDegree2),
              {NextConnectionDegree2 = 'all'}.

       /*     ================================================    */
       /*     =========      gomla_feealia_simple      =========  */
       /*     =========      (verb, faael, mafool_behe)    =========      */
       /*     ================================================    */
 gomla_feealia_simple(' جملة
'فعلية)TreeVerb,TreeFaael,TreeMafool),NextConnectionDegree,IsGomlaWithoutFaael<--(
              {IsGomlaWithoutFaael \== 'yes'},
       verb(TreeVerb, _, _, _ , _, VerbTrans,null,_ ,VerbPrefix,
_VerbSuffix,VerbRefuseFaaelInFollowingParse ,IsContainMafool),
              {
                     IsContainMafool = 'no',
                     VerbTrans = ',متعدي'
                     VerbRefuseFaaelInFollowingParse = 'no'
              },
       faael(TreeFaael, SexFaael, _ , _ , _Function,_,_,_, LastUnitType1,
_FirstUnitType1),
              {validate_faael_sex_agreement(VerbPrefix, SexFaael)},
       mafool_behe(TreeMafool, _,_,_,_,_,_,_,_, _LastUnitType2, FirstUnitType2),
              {refused_unit_decompos(LastUnitType1, FirstUnitType2)},
              {NextConnectionDegree = 'all'}.

       /*     ================================================    */
       /*     =========      gomla_feealia_simple          =========      */
       /*     =========      (verb, faael, complentry)     =========      */
       /*     ================================================    */
gomla_feealia_simple(' جملة
'فعلية)TreeVerb,TreeFaael,TreeComplemntry),NextConnectionDegree,IsGomlaWithoutFaael<--(
              {IsGomlaWithoutFaael \== 'yes'},
       verb(TreeVerb, _, _, _ , _, VerbTrans,null,_ ,VerbPrefix,
_VerbSuffix,VerbRefuseFaaelInFollowingParse , IsContainMafool),
              {
                     VerbTrans = '<-'لازم
                            VerbRefuseFaaelInFollowingParse = 'no'
                     ;
                            IsContainMafool = 'yes'
              },
       faael(TreeFaael, SexFaael, _ , _ , _Function ,_,_,_, _LastUnitType1,
_FirstUnitType1),
              {validate_faael_sex_agreement(VerbPrefix, SexFaael)},
       complentry(TreeComplemntry, _,_,_,_,_,_,_,_, _LastUnitType2, _FirstUnitType2,'yes',
_RefuseFaaelInFollowingParse,NextConnectionDegree).

       /*     ================================================    */
       /*     =========      gomla_feealia_simple          =========      */
       /*     =========      (verb, complentry, faael)     =========      */
       /*     ================================================    */
gomla_feealia_simple(' جملة
'فعلية)TreeVerb,TreeComplemntry,TreeFaael),NextConnectionDegree,IsGomlaWithoutFaael<--(
              {IsGomlaWithoutFaael \== 'yes'},
       verb(TreeVerb, _, _, _ , _, VerbTrans,null,_ ,VerbPrefix,
_VerbSuffix,VerbRefuseFaaelInFollowingParse , _IsContainMafool),
              {
                     VerbTrans = ',لازم'
                     VerbRefuseFaaelInFollowingParse = 'no'
              },
       complentry(TreeComplemntry, _,_,_,_,_,_,_,_, LastUnitType1, _FirstUnitType1,'no' ,
RefuseFaaelInFollowingParse,NextConnectionDegree),
              {RefuseFaaelInFollowingParse = 'no'},
```

```
	faael(TreeFaael, SexFaael, _Num , _Known , _Function ,_,_,_, _LastUnitType2,
FirstUnitType2),
		{validate_faael_sex_agreement(VerbPrefix, SexFaael)},
		{refused_unit_decompos(LastUnitType1, FirstUnitType2)},
		{NextConnectionDegree = 'all'}.

	/*	=====================================================	*/
	/*	=========	gomla_feealia_simple		=========	*/
	/*	=========	(verb, mafool_behe,complentry)	=========	*/
	/*	=====================================================	*/
gomla_feealia_simple(' جملة
فعلية'TreeVerb,TreeMafool,TreeComplemntry),NextConnectionDegree,_IsGomlaWithoutFaael<--(
	verb(TreeVerb, _, _, _ , _, VerbTrans,null,_ ,_VerbPrefix,
_VerbSuffix,_VerbRefuseFaaelInFollowingParse , IsContainMafool),
		{
			IsContainMafool = 'no',
			VerbTrans = 'متعدي'
		},
	mafool_behe(TreeMafool, _,_,_,_,_,_,_, LastUnitType2, _FirstUnitType2),
	complentry(TreeComplemntry, _,_,_,_,_,_,_, _LastUnitType3, FirstUnitType3,'yes',
_RefuseFaaelInFollowingParse,NextConnectionDegree),
		{refused_unit_decompos(LastUnitType2, FirstUnitType3)}.

	/*	=====================================================	*/
	/*	=========	gomla_feealia_simple		=========	*/
	/*	=========	(verb, mafool_behe,complentry)	=========	*/
	/*	=====================================================	*/
/*
gomla_feealia_simple(' جملة
فعلية'TreeVerb,TreeMafool,TreeComplemntry),NextConnectionDegree,_IsGomlaWithoutFaael<--(
	verb(TreeVerb, _, _, _ , _, VerbTrans,null,_ ,_VerbPrefix,
_VerbSuffix,VerbRefuseFaaelInFollowingParse , IsContainMafool),
		{
			IsContainMafool = 'no',
			VerbTrans = 'متعدي',
			VerbRefuseFaaelInFollowingParse = 'yes'
		},
	mafool_behe(TreeMafool, _,_,_,_,_,_,_, LastUnitType2, _FirstUnitType2),
	complentry(TreeComplemntry, _,_,_,_,_,_,_, _LastUnitType3, FirstUnitType3,'yes',
_RefuseFaaelInFollowingParse,NextConnectionDegree),
		{refused_unit_decompos(LastUnitType2, FirstUnitType3)}.
*/


	/*	=====================================================	*/
	/*	=========	gomla_feealia_simple		=========	*/
	/*	=========	(verb, complentry, mafool_behe)	=========	*/
	/*	=====================================================	*/
gomla_feealia_simple(' جملة
فعلية'TreeVerb,TreeComplemntry,TreeMafool),NextConnectionDegree,_IsGomlaWithoutFaael<--(
	verb(TreeVerb, _, _, _ , _, VerbTrans,null,_ ,_VerbPrefix,
_VerbSuffix,_VerbRefuseFaaelInFollowingParse , IsContainMafool),
		{
			IsContainMafool = 'no',
			VerbTrans = 'متعدي'
		},
	complentry(TreeComplemntry, _,_,_,_,_,_,_, LastUnitType1, _FirstUnitType1,'no',
_RefuseFaaelInFollowingParse,NextConnectionDegree),
	mafool_behe(TreeMafool, _,_,_,_,_,_,_, _LastUnitType2, FirstUnitType2),
		{refused_unit_decompos(LastUnitType1, FirstUnitType2)},
		{NextConnectionDegree = 'all'}.

	/*	=================================================================	*/
	/*	=========	gomla_feealia_simple
	=========	*/
	/*	=========	(verb, complentry, mafool_behe,complentry)	=========
	*/
```

```
	/*	================================================================	*/
gomla_feealia_simple(' جملة
)'فعلية TreeVerb,TreeComplemntry,TreeMafool,TreeComplemntry2),NextConnectionDegree,_IsGoml
aWithoutFaael<--(
	verb(TreeVerb, _, _, _ , _, VerbTrans,null,_ ,_VerbPrefix,
_VerbSuffix,_VerbRefuseFaaelInFollowingParse , IsContainMafool),
		{
			IsContainMafool = 'no',
			VerbTrans = 'متعدي'
		},
	complentry(TreeComplemntry, _,_,_,_,_,_,_, LastUnitType1, _FirstUnitType1,'no',
_RefuseFaaelInFollowingParse,_NextConnectionDegree1),
	mafool_behe(TreeMafool, _,_,_,_,_,_,_, LastUnitType2, FirstUnitType2),
		{refused_unit_decompos(LastUnitType1, FirstUnitType2)},
	complentry(TreeComplemntry2, _,_,_,_,_,_,_, _LastUnitType3, FirstUnitType3,'yes',
_RefuseFaaelInFollowingParse,_NextConnectionDegree2),
		{refused_unit_decompos(LastUnitType2, FirstUnitType3)},
		{NextConnectionDegree = 'all'}.


	/*	================================================	*/
	/*	=========	gomla_feealia_simple	=========	*/
	/*	=========	(verb, complentry)		=========	*/
	/*	================================================	*/
gomla_feealia_simple(' جملة
)'فعلية TreeVerb,TreeComplemntry),NextConnectionDegree,_IsGomlaWithoutFaael<--(
	verb(TreeVerb, _, _, _ , _, _VerbTrans,null,_ ,_VerbPrefix,
_VerbSuffix,VerbRefuseFaaelInFollowingParse , _IsContainMafool),
		{	VerbRefuseFaaelInFollowingParse = 'yes' ->
				AcceptAllComplentryTypes = 'yes'
			;
				AcceptAllComplentryTypes = 'no'
		},
	complentry(TreeComplemntry, _,_,_,_,_,_,_, _LastUnitType,
_FirstUnitType,AcceptAllComplentryTypes ,
_RefuseFaaelInFollowingParse,NextConnectionDegree).

	/*	================================================	*/
	/*	=========	gomla_feealia_simple	=========	*/
	/*	=========	(verb, faael)			=========	*/
	/*	================================================	*/
gomla_feealia_simple(' جملة
)'فعلية TreeVerb,TreeFaael),NextConnectionDegree,IsGomlaWithoutFaael<--(
		{IsGomlaWithoutFaael \== 'yes'},
	verb(TreeVerb, _, _, _ , _, _VerbTrans,null,_ ,VerbPrefix,
_VerbSuffix,VerbRefuseFaaelInFollowingParse , IsContainMafool),
		{VerbRefuseFaaelInFollowingParse = 'no' ; IsContainMafool = 'yes'},
/*		{VerbTrans \== 'متعدي'/*,{
	faael(TreeFaael, SexFaael, _ , _ , _ ,_,_,_, _LastUnitType1, _FirstUnitType1),
		{validate_faael_sex_agreement(VerbPrefix, SexFaael)},
		{NextConnectionDegree = 'all'}.

	/*	================================================	*/
	/*	=========	gomla_feealia_simple	=========	*/
	/*	=========	(verb, mafool_behe)		=========	*/
	/*	================================================	*/
gomla_feealia_simple(' جملة
)'فعلية TreeVerb,TreeMafool),NextConnectionDegree,_IsGomlaWithoutFaael<--(
	verb(TreeVerb, _, _, _ , _, VerbTrans,null,_ ,_VerbPrefix,
_VerbSuffix,_VerbRefuseFaaelInFollowingParse , IsContainMafool),
		{
			IsContainMafool = 'no',
			VerbTrans = 'متعدي'
		},
	mafool_behe(TreeMafool, _,_,_,_,_,_,_, _LastUnitType2, _FirstUnitType2),
		{NextConnectionDegree = 'all'}.
```

```
        /*      ================================================    */
        /*      =========      gomla_feealia_simple          =========      */
        /*      =========      (ENN's & KAN's brothors)      =========      */
        /*      ================================================    */
gomla_feealia_simple(TreeGomlaKan,NextConnectionDegree,_IsGomlaWithoutFaael)-->
        gomla_kan(TreeGomlaKan,NextConnectionDegree).

gomla_feealia_simple(TreeGomlaKan,NextConnectionDegree,_IsGomlaWithoutFaael)-->
        gomla_enn(TreeGomlaKan,NextConnectionDegree).


        /*      ================================================      */
        /*      =========      gomla_feealia_prefixed    =========   */
        /*      ================================================      */
gomla_feealia_prefixed('جملة فعلية يتصدرها مكمل جملة مقدم'(TreeComplemntry,
Tree),NextConnectionDegree, IsPrefixComplementryAccepted, IsGomlaWithoutFaael)-->
                {IsPrefixComplementryAccepted = 'yes'},
        complentry(TreeComplemntry, _,_,_,_,_,_,_, _LastUnitType1, _FirstUnitType1,'yes',
_RefuseFaaelInFollowingParse,_NextConnectionDegree1),
        gomla_feealia_simple(Tree,NextConnectionDegree,IsGomlaWithoutFaael).

        /*      ========================================    */
        /*      =========      gomla_feealia      =========  */
        /*      ==== can be considered as compound  ====     */
        /*      ========================================    */

gomla_feealia('جملة فعلية مركبة'(Tree1,TreeConnect,Tree2),NextConnectionDegree2,
IsPrefixComplementryAccepted, IsGomlaWithoutFaael)<--
        gomla_feealia_simple(Tree1,_NextConnectionDegree1,IsGomlaWithoutFaael),
        connector(TreeConnect,'all'),
        gomla_feealia_prefixed(Tree2,NextConnectionDegree2,IsPrefixComplementryAccepted,
_IsGomlaWithoutFaael2).

gomla_feealia(Tree,NextConnectionDegree, _IsPrefixComplementryAccepted,
IsGomlaWithoutFaael)-->
        gomla_feealia_simple(Tree,NextConnectionDegree,IsGomlaWithoutFaael).

gomla_feealia(Tree,NextConnectionDegree, IsPrefixComplementryAccepted,
IsGomlaWithoutFaael)-->
        gomla_feealia_prefixed(Tree,NextConnectionDegree,IsPrefixComplementryAccepted,
IsGomlaWithoutFaael).

/*********************************************
 *
 *                          *GOMLA_FEEALIA_KAN*
 *
 *********************************************/
gomla_kan('< أخوات كان'(TreeVerb,TreeKhabr),NextConnectionDegree<--(جملة فعلية
        verb(TreeVerb, _, _, _ , _, _,'_,_كان'(VerbPrefix,
_VerbSuffix,_RefuseFaaelInFollowingParse , _IsContainMafool,(
        dameer_motasel(_TreeDameerMotasel, _,_, _, _, _,_,_,_RefuseFaaelInFollowingParse
,_AcceptAdVerb , _IsMafool),
        khabar(TreeKhabr,  _ , _ , _, _, _,_,_, 'no', _,_,NextConnectionDegree) , !.

gomla_kan('< أخوات كان'(TreeVerb,TreeGomlaEssmya),NextConnectionDegree<--(جملة فعلية
        verb(TreeVerb, _, _, _ , _, _,'_,_كان'(VerbPrefix,
_VerbSuffix,_RefuseFaaelInFollowingParse , _IsContainMafool,(
        gomla_essmya(TreeGomlaEssmya,_, 'no', 'yes', _LastUnitType,
_FirstUnitType,NextConnectionDegree) , !.

gomla_kan('< أخوات كان'(TreeVerb,TreeKhabr),NextConnectionDegree<--(جملة فعلية
        verb(TreeVerb, _, _, _ , _, _,'_,_كان'(VerbPrefix,
_VerbSuffix,_RefuseFaaelInFollowingParse , _IsContainMafool,(
```

```
       khabar(TreeKhabr,  _ , _ , _, _, _,_,_, 'no', _,_,NextConnectionDegree).

/*********************************************
 *
 *                            *GOMLA_FEEALIA_ENN*
 *
 *********************************************/
gomla_enn('< جملة فعلية >أخوات
')'إن TreeVerb,TreeDameerMotasel,TreeKhabr),NextConnectionDegree<--(
       verb(TreeVerb, _, _, _ , _, _,'_,_'أن VerbPrefix,
_VerbSuffix,_RefuseFaaelInFollowingParse , _IsContainMafool,(
       dameer_motasel(TreeDameerMotasel, _,_, _, _,
_,_,_,_RefuseFaaelInFollowingParse,_AcceptAdVerb , _IsMafool),
       khabar(TreeKhabr,  _ , _ , _, _, _,_,_, 'no', _,_,NextConnectionDegree),!.

gomla_enn('< )'إن أخوات > جملة فعلية TreeVerb,TreeGomlaEssmya),NextConnectionDegree<--(
       verb(TreeVerb, _, _, _ , _, _,'_,_'أن VerbPrefix,
_VerbSuffix,_RefuseFaaelInFollowingParse , _IsContainMafool,(
       gomla_essmya(TreeGomlaEssmya,_, 'no', 'yes', _LastUnitType,
_FirstUnitType,NextConnectionDegree), !.

gomla_enn('< )'إن أخوات > جملة فعلية TreeVerb,TreeKhabr),NextConnectionDegree<--(
       verb(TreeVerb, _, _, _ , _, _,'_,_'أن VerbPrefix,
_VerbSuffix,_RefuseFaaelInFollowingParse , _IsContainMafool,(
       khabar(TreeKhabr,  _ , _ , _, _, _,_,_, 'no', _,_,NextConnectionDegree).

/*********************************************
 *
 *                            *GOMLA_RELATED*
 *
 *********************************************/
gomla_related(')'جملة متعلقة Tree1, TreeShepehGomla1, TreeConnector, TreeTarkeebSwaa),
LastUnitType2, FirstUnitType1, NextConnectionDegree<--(
       essm_swaa(Tree1),
       shepeh_gomla(TreeShepehGomla1, _, _, _Known, _, _,_,_, _LastUnitType1,
FirstUnitType1, 'not or' ,_LastTermType),
       or_connector(TreeConnector),
       tarkeeb_swaa(TreeTarkeebSwaa,_, _, _Known, _, _,_,_, LastUnitType2,
_FirstUnitType2),
              {NextConnectionDegree = 'not or'}.


/*********************************************
 *
 *                            *TARKEEB_SWAA*
 *
 *********************************************/
tarkeeb_swaa_simple(')'تركيب مكمل لسواء TreeTarkeeb), _, _, _Known, _, _,_,_,
LastUnitType, FirstUnitType<--(
       shepeh_gomla(TreeTarkeeb, _, _, _Known, _, _,_,_, LastUnitType, FirstUnitType,
'not or' ,_LastTermType);
       tarkeeb_eddafy(TreeTarkeeb, _, _ , _Known,_,_,_,_, _Contain_Dameer, LastUnitType,
FirstUnitType,'not or');
       gomla_essmya(TreeTarkeeb,_, 'no', 'yes', LastUnitType,
FirstUnitType,_NextConnectionDegree).

tarkeeb_swaa_simple(')'تركيب مكمل لسواء TreeTarkeeb), _, _, _Known, _, _,_,_,
LastUnitType, FirstUnitType<--(
       tarkeeb_wasfy(TreeTarkeeb, _, _, _Known1,_,_,LastUnitType,FirstUnitType).
```

```
tarkeeb_swaa_simple('تركيب مكمل لسواء'TreeMaarefa), _, _, _Known, _, _,_,_,
LastUnitType, FirstUnitType<--(
       maarefa(TreeMaarefa, _SEX , _NUM , Known, _, _,_,_ , _NounSpecial, AdVerb, _Type,
_Modaf),
              {LastUnitType = [Known,AdVerb,_X] , FirstUnitType = [Known,AdVerb,_X]}.

tarkeeb_swaa_connectd(' تراكيب مكملة
'لسواءTreeTarkeebSwaaSimple1,Connector,TreeTarkeebSwaaSimple2), _, _, _Known, _, _,_,_,
LastUnitType2, FirstUnitType1)-->
       tarkeeb_swaa_simple(TreeTarkeebSwaaSimple1, _, _, _Known, _, _,_,_,
_LastUnitType1, FirstUnitType1),
       or_connector(Connector),
       tarkeeb_swaa_simple(TreeTarkeebSwaaSimple2, _, _, _Known, _, _,_,_, LastUnitType2,
_FirstUnitType2).

tarkeeb_swaa_connectd(' تراكيب مكملة
'لسواءTreeTarkeebSwaaSimple1,Connector,TreeTarkeebSwaaSimple2), _, _, _Known, _, _,_,_,
LastUnitType2, FirstUnitType1)-->
       tarkeeb_swaa_simple(TreeTarkeebSwaaSimple1, _, _, _Known, _, _,_,_,
_LastUnitType1, FirstUnitType1),
       or_connector(Connector),
       tarkeeb_swaa_connectd(TreeTarkeebSwaaSimple2, _, _, _Known, _, _,_,_,
LastUnitType2, _FirstUnitType2).

tarkeeb_swaa(TreeTarkeebSwaa, _, _, _Known, _, _,_,_, LastUnitType, FirstUnitType)-->
       tarkeeb_swaa_simple(TreeTarkeebSwaa, _, _, _Known, _, _,_,_, LastUnitType,
FirstUnitType).
tarkeeb_swaa(TreeTarkeebSwaa, _, _, _Known, _, _,_,_, LastUnitType, FirstUnitType)-->
       tarkeeb_swaa_connectd(TreeTarkeebSwaa, _, _, _Known, _, _,_,_, LastUnitType,
FirstUnitType).


/*********************************************
 *
 *                            *SELAT_EL_MOSOL*
 *
 *********************************************/
selat_el_mosol('صلة الموصول'TreeEsmSela,TreeGomla), Sex , _ , _, _,
_,_,_,NextConnectionDegree<--(
       essm_sela(TreeEsmSela, Sex, _ , _, _, _,_,_),
       gomla_feealia(TreeGomla,NextConnectionDegree, 'no', 'no') ,!.

/*********************************************
 *
 *                            *MOBTADAA*
 *
 *********************************************/
mobtadaa_simple('مبتدأ'TreeTarkeebWasfy) ,SEX , NUM , _, _, _,_,_, 'no',
LastUnitType,FirstUnitType<--(
       tarkeeb_wasfy(TreeTarkeebWasfy, SEX,NUM, '_,_,'معرفةLastUnitType,FirstUnitType ,(
.!
mobtadaa_simple('مبتدأ'TreeTarkeebEddafy) ,SEX , NUM , _, _, _,_,_, Contain_Dameer,
LastUnitType,  FirstUnitType<--(
       tarkeeb_eddafy(TreeTarkeebEddafy, SEX, NUM, _, _, _,_,_, Contain_Dameer,
LastUnitType,  FirstUnitType,'all'),
              {reject_type(FirstUnitType,'ظرف'.!, {)
mobtadaa_simple('مبتدأ'TreeTameez) ,SEX , NUM , _, _, _,_,_, Contain_Dameer,
LastUnitType,  FirstUnitType<--(
       tarkeeb_tameezy(TreeTameez, LastUnitType,  FirstUnitType),
              {SEX = 'nutral', NUM = 'nutral' , Contain_Dameer = 'no'} ,!.
```

```
mobtadaa_simple('مبتدأ'TreeTarkeebBadal) ,SEX , NUM , _, _, _,_,_, 'no', LastUnitType,
FirstUnitType<--(
       tarkeeb_badal(TreeTarkeebBadal, SEX , NUM , _Known, _, _,_,_, LastUnitType,
FirstUnitType) ,!.
mobtadaa_simple('مبتدأ'TreeMaarefa) ,SEX , NUM , _, _, _,_,_, 'null', LastUnitType,
FirstUnitType<--(
       maarefa(TreeMaarefa, SEX , NUM , Known, _, _,_,_ , _NounSpecial, AdVerb, _Type,
_Modaf),
              {LastUnitType = [Known,AdVerb,_X] , FirstUnitType = [Known,AdVerb,_X]} ,!.
mobtadaa_simple('مبتدأ'TreeDameerMonfasel) ,SEX , NUM , _, _, _,_,_, 'yes',
LastUnitType,  FirstUnitType<--(
       dameer_monfasel(TreeDameerMonfasel, SEX, NUM, _, _, _,_,_),
              {LastUnitType = ['ضمير_منفصل', null, null], FirstUnitType = ['ضمير_منفصل',
null, null]  }.

mobtadaa_simple('مبتدأ'TreeEsharaNoun) ,SEX , NUM , _, _, _,_,_, 'no', LastUnitType,
FirstUnitType<--(
       eshara_noun(TreeEsharaNoun, SEX, NUM, _, _, _,_,_),
              {LastUnitType = ['اسم اشارة', null,null], FirstUnitType = ['اسم اشارة',
null,null].{ [


mobtadaa_mosool('مبتدأ'Tree, TreeSelatElMosol),SEX1 , NUM , _, _, _,_,_, ContainDameer,
LastUnitType, FirstUnitType<--(
       mobtadaa_simple(Tree,SEX1 , NUM , _, _, _,_,_, ContainDameer, LastUnitType,
FirstUnitType),
       selat_el_mosol(TreeSelatElMosol, Sex2 , _ , _, _, _,_,_,_NextConnectionDegree),
              {SEX1 = 'nutral' ; SEX1 = Sex2}.

mobtadaa(Tree, SEX , NUM, _, _, _,_,_, ContainDameer, LastUnitType, FirstUnitType)-->
       mobtadaa_mosool(Tree,SEX , NUM , _, _, _,_,_, ContainDameer, LastUnitType,
FirstUnitType) ,!.
mobtadaa(Tree, SEX , NUM, _, _, _,_,_, ContainDameer, LastUnitType, FirstUnitType)-->
       mobtadaa_simple(Tree,SEX , NUM , _, _, _,_,_, ContainDameer, LastUnitType,
FirstUnitType).

/*********************************************
 *
 *                            *khabar*
 *
 *********************************************/

       /*     ========================================    */
       /*     =========      khabar_simple      =========  */
       /*     ========================================    */
khabar_simple(TreeShepehGomla, 'nutral' , 'nutral' , Known, _, _, _,_, _KhabarFree,
LastUnitType, FirstUnitType)-->
       shepeh_gomla(TreeShepehGomla, _, _, Known, _, _,_,_, LastUnitType, FirstUnitType,
'all' ,_LastTermType) ,!.

khabar_simple(TreeTarkeebWasfy, SEX , NUM , _, _, _,_,_,
_KhabarFree,LastUnitType,FirstUnitType)-->
       tarkeeb_wasfy(TreeTarkeebWasfy, SEX,NUM, '_,_,نكرة',LastUnitType,FirstUnitType (
.!,

khabar_simple(TreeTarkeebEddafy, SEX, NUM, _, _, _,_,_, _KhabarFree, LastUnitType,
FirstUnitType)-->
       tarkeeb_eddafy(TreeTarkeebEddafy, SEX, NUM, _, _, _,_,_,_, LastUnitType,
FirstUnitType,'all') ,!.

khabar_simple(TreeTameez, 'nutral', 'nutral', _, _, _,_,_, _KhabarFree, LastUnitType,
FirstUnitType)-->
       tarkeeb_tameezy(TreeTameez, LastUnitType,  FirstUnitType) ,!.
```

```
khabar_simple('خبر و صلة الموصول')(TreeNakera,TreeSelatElMosol), SEX , NUM , _, _, _,_,_,
_KhabarFree,LastUnitType, FirstUnitType<--(
       nakera(TreeNakera , SEX , NUM , Known, _, _,_,_ , _NounSpecial, AdVerb,_Modaf),
              {LastUnitType = [Known, AdVerb, _X], FirstUnitType = [Known, AdVerb, _Y]},
       selat_el_mosol(TreeSelatElMosol, Sex2 , _ , _, _, _,_,_,_NextConnectionDegree),
              {SEX1 = 'nutral' ; SEX1 = Sex2} ,!.

khabar_simple(TreeNakera, SEX , NUM , _, _, _,_,_, _KhabarFree, LastUnitType,
FirstUnitType)-->
       nakera(TreeNakera , SEX , NUM , Known, _, _,_,_ , _NounSpecial, AdVerb,_Modaf),
              {LastUnitType = [Known, AdVerb, _X], FirstUnitType = [Known, AdVerb, _Y]}.

       /*     ========================================================================
       */
       /*     =========      Khabar Free (accepted only in special cases)    =========
       */
       /*     ========================================================================
       */

khabar_simple('خبر')(TreeTarkeebWasfy), SEX , NUM , _, _, _,_,_,
'yes',LastUnitType,FirstUnitType<--(
       tarkeeb_wasfy(TreeTarkeebWasfy, SEX,NUM, ',_ ,_ ,'معرفة)LastUnitType,FirstUnitType (
.!,

khabar_simple('خبر')(TreeMaarefa), SEX , NUM , _, _, _,_,_, 'yes',LastUnitType,
FirstUnitType<--(
       maarefa(TreeMaarefa, SEX , NUM ,Known,_,_,_,_ , _NounSpecial, AdVerb1,
_MaarefaType, _Modaf),
              {LastUnitType = [Known,AdVerb1,_X] , FirstUnitType = [Known,AdVerb1,_Y]}.

       /*     ========================================    */
       /*     =========      khabar_as_gomla           =========  */
       /*     ========================================    */
khabar_as_gomla(TreeGomlaFeealia, 'nutral', 'nutral', _, _, _,_,_, _KhabarFree,
LastUnitType,  FirstUnitType ,NextConnectionDegree)-->
       gomla_feealia(TreeGomlaFeealia,NextConnectionDegree, 'no', 'yes'),
              {LastUnitType = [null,null,null],  FirstUnitType = [null,null,null]} ,!.

khabar_as_gomla(TreeGomla, 'nutral', 'nutral', _, _, _,_,_, _KhabarFree, LastUnitType,
FirstUnitType,NextConnectionDegree)-->
       gomla_essmya(TreeGomla, 'yes', 'yes', 'no', LastUnitType,
FirstUnitType,NextConnectionDegree).

       /*     ======================================================      */
       /*     =========      Khabar and related unit to it    =========   */
       /*     ======================================================      */
khabar_and_related('خبر و متعلق به')(TreeKhabar,TreeComplemntry), SEX , NUM , Known, _,
_,_,_, KhabarFree, LastUnitType2, FirstUnitType1,NextConnectionDegree<--(
       khabar_simple(TreeKhabar, SEX,NUM, Known, _, _,_,_, KhabarFree, LastUnitType1,
FirstUnitType1),
       complentry(TreeComplemntry, _,_,_,_,_,_,_, LastUnitType2, FirstUnitType2,'yes' ,
_RefuseFaaelInFollowingParse,NextConnectionDegree),
              {refused_unit_decompos(LastUnitType1, FirstUnitType2)} ,!.

khabar_and_related('خبر و متعلق به')(TreeKhabar,TreeGomla, TreeComplemntry), SEX , NUM ,
Known, _, _,_,_, KhabarFree, LastUnitType3 , FirstUnitType1 ,NextConnectionDegree2)-->
       khabar_simple(TreeKhabar, SEX,NUM, Known, _, _,_,_, KhabarFree, LastUnitType1,
FirstUnitType1),
       khabar_as_gomla(TreeGomla, _SEX,_NUM, _Known, _, _,_,_, _KhabarFree2,
LastUnitType2, FirstUnitType2,_NextConnectionDegree1),
              {refused_unit_decompos(LastUnitType1, FirstUnitType2)} ,
       complentry(TreeComplemntry, _,_,_,_,_,_,_, LastUnitType3, FirstUnitType3,'yes' ,
_RefuseFaaelInFollowingParse,NextConnectionDegree2),
              {refused_unit_decompos(LastUnitType2, FirstUnitType3)} ,!.
```

```
/*
khabar_and_related('خبر و متعلق به'TreeKhabar,TreeGomlaRelated), SEX , NUM , Known, _,
_,_,_, KhabarFree, LastUnitType2, FirstUnitType1,NextConnectionDegree<--(
       khabar_simple(TreeKhabar, SEX,NUM, Known, _, _,_,_, KhabarFree, _LastUnitType1,
FirstUnitType1),
       gomla_related(TreeGomlaRelated,LastUnitType2,
_FirstUnitType2,NextConnectionDegree) ,!.
*/

       /*     ================================== */
       /*     =========      khabar_compund =========      */
       /*     ================================== */

khabar_compund('خبر يسبقه مكمل جملة'Tree1,TreeGomlaFeealia), SEX,NUM, Known, _, _,_,_,
_KhabarFree, LastUnitType, FirstUnitType,NextConnectionDegree2<--(
       complentry(Tree1, SEX , NUM , Known , _,_,_,_, _LastUnitType1, FirstUnitType,'yes'
, _RefuseFaaelInFollowingParse,_NextConnectionDegree),
       gomla_feealia(TreeGomlaFeealia,NextConnectionDegree2, 'no', 'no'),
              {LastUnitType = 'nutral'} ,!.
/*            {refused_unit_decompos(LastUnitType2, FirstUnitType3)} ,!.*/

khabar_compund('خبر يسبقه مكمل جملة'Tree1,TreeGomla), SEX,NUM, Known, _, _,_,_,
_KhabarFree, LastUnitType2, FirstUnitType1,NextConnectionDegree2<--(
       complentry(Tree1, SEX , NUM , Known , _,_,_,_, LastUnitType1, FirstUnitType1,'yes'
, _RefuseFaaelInFollowingParse,_NextConnectionDegree),
       gomla_essmya(TreeGomla, 'yes', 'yes', 'no', LastUnitType2,
FirstUnitType2,NextConnectionDegree2),
       {refused_unit_decompos(LastUnitType1, FirstUnitType2)}.


       /*     ============================  */
       /*     =========      Khabar =========      */
       /*     ============================  */

khabar('خبر'Tree1,Tree2), SEX , NUM , Known, _, _,_,_, KhabarFree, LastUnitType,
FirstUnitType,NextConnectionDegree<--(
       reduntant_tarkeeb(Tree1),
       khabar(Tree2, SEX , NUM , Known, _, _,_,_, KhabarFree, LastUnitType,
FirstUnitType,NextConnectionDegree) ,!.
khabar('خبر'Tree), SEX , NUM , Known, _, _,_,_, KhabarFree, LastUnitType,
FirstUnitType,NextConnectionDegree<--(
       khabar_compund(Tree, SEX,NUM, Known, _, _,_,_, KhabarFree, LastUnitType,
FirstUnitType,NextConnectionDegree),!.
khabar('خبر'Tree), SEX , NUM , Known, _, _,_,_, KhabarFree, LastUnitType,
FirstUnitType,NextConnectionDegree<--(
       khabar_and_related(Tree, SEX , NUM , Known, _, _,_,_, KhabarFree, LastUnitType,
FirstUnitType,NextConnectionDegree),!.
khabar('خبر'Tree), SEX , NUM , Known, _, _,_,_, KhabarFree, LastUnitType,
FirstUnitType,NextConnectionDegree<--(
       khabar_as_gomla(Tree, SEX,NUM, Known, _, _,_,_, KhabarFree, LastUnitType,
FirstUnitType,NextConnectionDegree),!.
khabar('خبر'Tree), SEX , NUM , Known, _, _,_,_, KhabarFree, LastUnitType,
FirstUnitType,NextConnectionDegree<--(
       khabar_simple(Tree, SEX,NUM, Known, _, _,_,_, KhabarFree, LastUnitType,
FirstUnitType),
       {NextConnectionDegree = 'all'}.


/*********************************************
 *
 *                            *SHEPEH_GOMLA*
 *
 *********************************************/
```

```
        /*      ========================================    */
        /*      =========      shepeh_gomla_simple    =========      */
        /*      ========================================    */

shepeh_gomla_simple('شبه جملة'TreeHarfGar, TreeTarkeebWasfy) , Sex , Number , Known,
_,_,_,_, LastUnitType, FirstUnitType  , _ConnectinDegree, LastTermType<--(
        harf_gar(TreeHarfGar , _,_,_,_,_,_,_,_HarfGar_Class),
        tarkeeb_wasfy(TreeTarkeebWasfy, Sex ,Number
,Known,_,_,LastUnitType,_FirstUnitType),
                {FirstUnitType = ['حرف_جر','null','null',{['
                {LastTermType = '.!, {'تـركـيـب وصفـي

shepeh_gomla_simple('شبه جملة'TreeHarfGar, TreeTarkeebEddafy) , Sex , Number , Known,
_,_,_,_, LastUnitType, FirstUnitType , ConnectinDegree, LastTermType<--(
        harf_gar(TreeHarfGar , _,_,_,_,_,_,_,_HarfGar_Class),
        tarkeeb_eddafy(TreeTarkeebEddafy, Sex ,Number ,Known,_,_,_,_,_, LastUnitType,
_FirstUnitType, ConnectinDegree),
                {FirstUnitType = ['حرف_جر','null','null',{['
                {LastTermType = '.!, {'تـركـيـب إضـافي

shepeh_gomla_simple('شبه جملة'TreeHarfGar, TreeTarkeebBadal) , Sex , Number,'مـعـرفـة',
_,_,_,_LastUnitType, FirstUnitType  , _ConnectinDegree, LastTermType)-->
        harf_gar(TreeHarfGar , _,_,_,_,_,_,_,_HarfGar_Class),
        tarkeeb_badal(TreeTarkeebBadal, Sex ,Number , _Known, _, _,_,_, LastUnitType,
_FirstUnitType),
                {FirstUnitType = ['حرف_جر','null','null',{['
                {LastTermType = '.!, {'تـركـيـب بـدل

shepeh_gomla_simple('شبه جملة'TreeHarfGar, TreeDameerMotasel) , Sex , Number , 'dameer',
_,_,_,_, LastUnitType, FirstUnitType  , _ConnectinDegree, LastTermType<--(
        harf_gar(TreeHarfGar , _,_,_,_,_,_,_,_HarfGar_Class),
        dameer_motasel(TreeDameerMotasel, Sex , Number, _, _,
_,_,_,_RefuseFaaelInFollowingParse,_AcceptAdVerb , _IsMafool),
                {FirstUnitType = ['حرف_جر','null','null',{['
                {LastUnitType = ['ضمير_مـتصل','null','null',{['
                {LastTermType =  '.!, {'ضمير_مـتصل

shepeh_gomla_simple('شبه جملة'TreeHarfGar, TreeNoun1,TreeConnect,TreeNoun2) , Sex ,
Number , Known1, _,_,_,_, LastUnitType, FirstUnitType , ConnectinDegree, LastTermType<--(
        harf_gar(TreeHarfGar , _,_,_,_,_,_,_,_HarfGar_Class),
        noun(TreeNoun1, Sex , Number , Known1,_,_,_,_, _LastUnitType , _NounSpecial),
        connector(TreeConnect,ConnectinDegree),
        noun(TreeNoun2, _Sex , _Number , Known2,_,_,_,_, LastUnitType , _NounSpecial),
                {FirstUnitType = ['حرف_جر','null','null',{['
                {Known1 = Known2 , LastTermType = Known2} ,!.

shepeh_gomla_simple('شبه جملة'TreeHarfGar, TreeNoun) , Sex , Number , Known, _,_,_,_,
LastUnitType, FirstUnitType , _ConnectinDegree, LastTermType<--(
        harf_gar(TreeHarfGar , _,_,_,_,_,_,_,_HarfGar_Class),
        noun(TreeNoun, Sex , Number , Known,_,_,_,_, LastUnitType , _NounSpecial),
                {FirstUnitType = ['حرف_جر','null','null',{['
                {LastTermType = Known}.


        /*      ============================================        */
        /*      =========      shepeh_gomla_connected =========      */
        /*      ============================================        */
/*
```

```
shepeh_gomla_connected('أشباه جمل
'معطوفةTreeShepehGomla1,TreeConnector,TreeShepehGomla2), Sex , Number , Known, _,_,_,_,
LastUnitType2, FirtsUnitType1, ConnectinDegree,LastTermType)-->
       shepeh_gomla_simple(TreeShepehGomla1, Sex , Number , Known, _,_,_,_,
_LastUnitType1, FirtsUnitType1,ConnectinDegree,_LastTermType),
       connector(TreeConnector,ConnectinDegree),
       shepeh_gomla_simple(TreeShepehGomla2, _Sex , _Number , _Known, _,_,_,_,
LastUnitType2, _FirtsUnitType2,ConnectinDegree,LastTermType).

shepeh_gomla_connected('أشباه جمل
'معطوفةTreeShepehGomla1,TreeConnector,TreeShepehGomla2), Sex , Number , Known, _,_,_,_,
LastUnitType2, FirtsUnitType1, ConnectinDegree,LastTermType)-->
       shepeh_gomla_simple(TreeShepehGomla1, Sex , Number , Known, _,_,_,_,
_LastUnitType1, FirtsUnitType1,ConnectinDegree,_LastTermType),
       connector(TreeConnector,ConnectinDegree),
       shepeh_gomla_connected(TreeShepehGomla2, _Sex , _Number , _Known, _,_,_,_,
LastUnitType2, _FirtsUnitType2,ConnectinDegree,LastTermType).
*/

/*
shepeh_gomla(TreeShepehGomla , Sex , Number , Known, _,_,_,_, LastUnitType,
FirtsUnitType, ConnectinDegree,LastTermType)-->
       shepeh_gomla_connected(TreeShepehGomla, Sex , Number , Known, _,_,_,_,
LastUnitType, FirtsUnitType,ConnectinDegree,LastTermType).
*/

shepeh_gomla('شبه جملة و صلة الموصول'TreeShepehGomla,TreeSelatElMosol), Sex , Number ,
Known, _,_,_,_, LastUnitType, FirtsUnitType, ConnectinDegree2 , LastTermType)-->
       shepeh_gomla_simple(TreeShepehGomla, Sex , Number , Known, _,_,_,_, LastUnitType,
FirtsUnitType,_ConnectinDegree1,LastTermType),
       selat_el_mosol(TreeSelatElMosol, _Sex2 , _ , _, _, _,_,_,ConnectinDegree2) ,!.
       /*{Sex = 'nutral' ; Sex = Sex2},*/

shepeh_gomla(TreeShepehGomla , Sex , Number , Known, _,_,_,_, LastUnitType,
FirtsUnitType, ConnectinDegree,LastTermType)-->
       shepeh_gomla_simple(TreeShepehGomla, Sex , Number , Known, _,_,_,_, LastUnitType,
FirtsUnitType,ConnectinDegree,LastTermType).

/************************************************
 *
 *                            *SHEPEH_GOMLA_SINGLE_OR_COMPUND*
 *
 ************************************************/
shepeh_gomla_single_or_compund('أشباه جملة متعددة'TreeShepehGomla1,TreeShepehGomla2),
Sex , Number , Known, _,_,_,_, LastUnitType2, FirtsUnitType1,
ConnectinDegree,LastTermType2)-->
       shepeh_gomla_simple(TreeShepehGomla1, Sex , Number , Known, _,_,_,_,
_LastUnitType1, FirtsUnitType1, ConnectinDegree,_LastTermType1),
       shepeh_gomla_single_or_compund(TreeShepehGomla2, _Sex2 , _Number2 , _Known2,
_,_,_,_, LastUnitType2, _FirtsUnitType2, _ConnectinDegree2,LastTermType2),!.

shepeh_gomla_single_or_compund(TreeShepehGomla, Sex , Number , Known, _,_,_,_,
LastUnitType, FirtsUnitType, ConnectinDegree,LastTermType) -->
       shepeh_gomla_simple(TreeShepehGomla, Sex , Number , Known, _,_,_,_, LastUnitType,
FirtsUnitType, ConnectinDegree,LastTermType).

/*********************************************
 *
 *                            *SHEPEH_GOMLA_SPECIAL*
 *
 *********************************************/

shepeh_gomla_special(' )'أسلوب اختصاص - شبه جملةTreeHarfGar, TreeNoun) , Sex , Number
, Known, _,_,_,_, LastUnitType, FirstUnitType<--(
       harf_gar(TreeHarfGar , _,_,_,_,_,_,_,_HarfGar_Class),
       noun(TreeNoun, Sex , Number , Known,_,_,_,_, LastUnitType , ',اختصاص')
```

```
        {FirstUnitType = [' ,'حرف_جر'null,null.{[

/**********************************************
 *
 *                            *HARF_GAR*
 *
 **********************************************/
 harf_gar(')'حرف جر'Word) , _,_,_,_,_,_,_,HarfGar_Class<--(
        [Word, '_,_,_,_,_,_,_,_,'حرف_جر'NounSpecial, _, _AdVerb,_Modaf, _VerbPrefix,
_VerbSuffix,[
               { Word = '<-'لـل'
                      HarfGar_Class = 'double_lam'
                ;
                      HarfGar_Class = 'normal'
               }.

/**********************************************
 *
 *                            *ZARF*
 *
 **********************************************/
 zarf(')'زمـان ظرفWord<--(_,_,_,_,_,_,_  ,(
        [Word, '_ ,_,_,_,_,_,_,_,'ظرف_زمـنNounSpecial, _ ,_AdVerb, _Modaf, _VerbPrefix,
_VerbSuffix.[

zarf(')'مـكـان ظرفWord<--(_,_,_,_,_,_,_  ,(
        [Word, '_,_,_,_,_,_,_,_,'ظرف_مـكـنNounSpecial, _, _AdVerb, _Modaf, _VerbPrefix,
_VerbSuffix.[


/**********************************************
 *
 *                            *CONNECTOR*
 *
 **********************************************/
 connector(')'حرف وصل'Word), ConnectorType<--(
               {ConnectorType == 'all'},
        [Word, '_,_,_,_,_,_,_,_,'حرف_وصلNounSpecial, _,_AdVerb, _Modaf, _VerbPrefix,
_VerbSuffix.[

connector(')'حرف وصل'Word), ConnectorType<--(
               {ConnectorType == 'and'},
        [Word, '_,_,_,_,_,_,_,_,'حرف_وصلNounSpecial, _, _AdVerb, _Modaf, _VerbPrefix,
_VerbSuffix,[
               {Word = '.{'و

connector(')'حرف وصل'Word), ConnectorType<--(
               {ConnectorType == 'not or'},
        [Word, '_,_,_,_,_,_,_,_,'حرف_وصلNounSpecial, _, _AdVerb, _Modaf, _VerbPrefix,
_VerbSuffix,[
               {Word \== '.{'أو


/**********************************************
 *
 *                            *OR_CONNECTOR*
 *
 **********************************************/
```

```
 or_connector('حرف وصل'Word<--((
       [Word, '_,_,_,_,_,_,_,_'حرف_وصلNounSpecial, _, _AdVerb, _Modaf, _VerbPrefix,
_VerbSuffix,[
               {Word = 'أو'}.


/*********************************************
 *
 *                              *RABET_GOML*
 *
 *********************************************/
 rabet_goml1('رابط جمل'Word) ,_ConnectionDegree<--(
       [Word, '_,_,_,_,_,_,_,_'ربط_جملNounSpecial, _, _AdVerb,_Modaf, _VerbPrefix,
_VerbSuffix.[

 rabet_goml1('رابط جمل'TreeConnect) ,ConnectionDegree<--(
       connector(TreeConnect,ConnectionDegree).

 rabet_goml(Tree ,ConnectionDegree)-->
       rabet_goml1(Tree,ConnectionDegree).
 rabet_goml('رابط جمل'Tree ,Tree2),ConnectionDegree<--(
       rabet_goml1(Tree,ConnectionDegree), rabet_goml(Tree2,ConnectionDegree).

/*********************************************
 *
 *                              *NOUN*
 *
 *********************************************/
 noun(TreeNoun, Sex,Num ,Known,_,_,_,_, UnitType, NounSpecial)-->
       maarefa(TreeNoun, Sex,Num ,Known,_,_,_,_ , NounSpecial, AdVerb1, _MaarefaType,
_Modaf1),
               {UnitType = [Known , AdVerb1, _X]} ,!.

noun(TreeNoun, Sex,Num ,Known,_,_,_,_, UnitType, NounSpecial)-->
       nakera(TreeNoun, Sex,Num ,Known,_,_,_,_, NounSpecial, AdVerb2,Modaf2),
               {UnitType = [Known,AdVerb2, Modaf2]}.

/*********************************************
 *
 *                              *VERB*
 *
 *********************************************/
verb_simple('فعل يسبقه حرف'Word1,Word2,Trans,TreeDameerMotasel), Sex, Number, _ , _,
Trans, VerbSepcial,_ ,VerbPrefix, VerbSuffix,RefuseFaaelInFollowingParse,
IsContainMafool)-->
       [Word1,Type, _Sex, _Number, _,_,_Trans,_VerbSepcial,_NounSpecial, _, _AdVerb1,
_Modaf, _VerbPrefix, _VerbSuffix],
               {Type = 'ح_نفي'={'ح_تحقيق'Type ; '
       [Word2,'فعل',Sex, Number, _,_,Trans,VerbSepcial,_NounSpecial2, _, _AdVerb2,
_Modaf, VerbPrefix, VerbSuffix,[
       dameer_motasel(TreeDameerMotasel, _,_, _, _,
_,_,_,RefuseFaaelInFollowingParse,_AcceptAdVerb , IsMafool),
       {IsContainMafool = IsMafool}.

verb_simple('فعل يسبقه حرف'Word1,Word2,Trans), Sex, Number, _ , _, Trans,
VerbSepcial,_ ,VerbPrefix, VerbSuffix,RefuseFaaelInFollowingParse,IsContainMafool<--(
       [Word1,Type, _Sex, _Number, _Known1,_Function1,_Trans,_VerbSepcial,_NounSpecial,
_Wazn1, _AdVerb1, _Modaf, _VerbPrefix, _VerbSuffix],
               {Type = 'ح_نفي'={'ح_تحقيق'Type ; '
       [Word2,'فعل',Sex, Number, _Known2,_Function2,Trans,VerbSepcial,_NounSpecial2,
_Wazn, _AdVerb, _Modaf2, VerbPrefix, VerbSuffix,[
```

```
                {RefuseFaaelInFollowingParse = 'no'} ,
                {IsContainMafool = 'no'}.

verb_simple('فعل و ضمير متصل'Word,Trans,TreeDameerMotasel), Sex, Number, _ , _, Trans,
VerbSepcial,_ ,VerbPrefix, VerbSuffix,RefuseFaaelInFollowingParse , IsContainMafool)-->
        [Word,'فعل',Sex, Number, _,_,Trans,VerbSepcial,_NounSpecial, _, _AdVerb, _Modaf,
VerbPrefix, VerbSuffix,[
        dameer_motasel(TreeDameerMotasel, _,_, _, _,
_,_,_,RefuseFaaelInFollowingParse,_AcceptAdVerb , IsMafool),
        {IsContainMafool = IsMafool}.

verb_simple('فعل'Word,Trans), Sex, Number, _ , _, Trans, VerbSepcial,_ ,VerbPrefix,
VerbSuffix,RefuseFaaelInFollowingParse , IsContainMafool<--(
        [Word,'فعل',Sex, Number, _,_,Trans,VerbSepcial,_NounSpecial, _, _AdVerb, _Modaf,
VerbPrefix, VerbSuffix,[
                {RefuseFaaelInFollowingParse = 'no'} ,
                {IsContainMafool = 'no'}.

verb_connected('أفعال معطوفة'Tree1,TreeConnector,Tree2), Sex, Number, _ , _, Trans,
VerbSepcial,_ ,VerbPrefix, VerbSuffix,RefuseFaaelInFollowingParse , IsContainMafool1)-->
        verb_simple(Tree1,Sex, Number, _ , _, Trans, VerbSepcial,_ ,VerbPrefix,
VerbSuffix,RefuseFaaelInFollowingParse1 , IsContainMafool1),
        connector(TreeConnector,'all'),
        verb_simple(Tree2,Sex, Number, _ , _, Trans, VerbSepcial,_ ,VerbPrefix,
VerbSuffix,RefuseFaaelInFollowingParse2,_IsContainMafool2),
                {(RefuseFaaelInFollowingParse1 = 'yes' ; RefuseFaaelInFollowingParse2 =
'yes') ->
                        RefuseFaaelInFollowingParse = 'yes'
                  ;
                        RefuseFaaelInFollowingParse = 'no'
                }.

verb(Tree, Sex, Number, _ , _, Trans, VerbSepcial,_ ,VerbPrefix,
VerbSuffix,RefuseFaaelInFollowingParse , IsContainMafool)-->
        verb_connected(Tree, Sex, Number, _ , _, Trans, VerbSepcial,_ ,VerbPrefix,
VerbSuffix,RefuseFaaelInFollowingParse , IsContainMafool),!.
verb(Tree, Sex, Number, _ , _, Trans, VerbSepcial,_ ,VerbPrefix,
VerbSuffix,RefuseFaaelInFollowingParse , IsContainMafool)-->
        verb_simple(Tree, Sex, Number, _ , _, Trans, VerbSepcial,_ ,VerbPrefix,
VerbSuffix,RefuseFaaelInFollowingParse , IsContainMafool).

/*********************************************
 *
 *                            *FAAEL*
 *
 *********************************************/
faael_simple('فاعل'TreeTarkeebEddafy), Sex, _ , Known , _Function ,_,_,_, LastUnitType,
FirstUnitType<--(
        tarkeeb_eddafy(TreeTarkeebEddafy, Sex, _, Known, _Function, _,_,_,_, LastUnitType,
FirstUnitType,'all'),
                {reject_type(FirstUnitType,'.!, {('ظرف

faael_simple('فاعل'TreeTarkeebBadal), Sex, _ , Known , _Function ,_,_,_, LastUnitType,
FirstUnitType<--(
        tarkeeb_badal(TreeTarkeebBadal, Sex,_, Known, _, _,_,_, LastUnitType,
FirstUnitType) ,!.

faael_simple('فاعل'TreeTarkeebWasfy), Sex, _ , Known , Function ,_,_,_, LastUnitType,
FirstUnitType<--(
        tarkeeb_wasfy(TreeTarkeebWasfy, Sex
,_,Known,Function,_,LastUnitType,FirstUnitType) ,!.
```

```
faael_simple('فاعل'TreeMaarefa), Sex, _ , Known , Function ,_,_,_, LastUnitType,
FirstUnitType<--(
       maarefa(TreeMaarefa, Sex ,_,Known,Function,_,_,_ , _NounSpecial, AdVerb,
_Type,_Modaf),
              {LastUnitType = [Known,AdVerb,_X],  FirstUnitType = [Known,AdVerb,_Y]} ,!.

faael_simple('فاعل'TreeNakera), Sex, _ , Known , Function ,_,_,_, LastUnitType,
FirstUnitType<--(
       nakera(TreeNakera, Sex ,_,Known,Function,_,_,_, _NounSpecial, AdVerb,_Modaf),
              {LastUnitType = [Known,AdVerb,_X],  FirstUnitType = [Known,AdVerb,_Y]}.



faael_connected('فواعل معطوفة'Tree1,TreeConnector,Tree2), Sex, Num , Known , Function
,_,_,_, LastUnitType,  FirstUnitType<--(
       maarefa(Tree1, Sex ,Num,Known,Function,_,_,_ , _NounSpecial, AdVerb,
Type,_Modaf1),
       connector(TreeConnector,'all'),
       maarefa(Tree2, Sex ,Num,Known,Function,_,_,_ , _NounSpecial2, AdVerb2, Type,
_Modaf2),
              {LastUnitType = [Known,AdVerb,_X],  FirstUnitType = [Known,AdVerb2,_Y]}.

faael_connected('فواعل معطوفة'Tree1,TreeConnector,Tree2), Sex, Num , Known , Function
,_,_,_, LastUnitType2,  FirstUnitType1<--(
       maarefa(Tree1, Sex ,Num,Known,Function,_,_,_ , _NounSpecial, AdVerb, _Type,
_Modaf),
              {FirstUnitType1 = [Known,AdVerb,_X]},
       connector(TreeConnector,'all'),
       faael_connected(Tree2, _Sex, _Num , _Known , _Function ,_,_,_, LastUnitType2,
_FirstUnitType2).

faael(Tree, Sex, Num , Known , Function ,_,_,_, LastUnitType,  FirstUnitType)-->
       faael_simple(Tree, Sex, Num , Known , Function ,_,_,_, LastUnitType,
FirstUnitType),
              {validate_faael_function(Function)}.

faael(Tree, Sex, Num , Known , Function ,_,_,_, LastUnitType,  FirstUnitType)-->
       faael_connected(Tree, Sex, Num , Known , Function ,_,_,_, LastUnitType,
FirstUnitType),
              {validate_faael_function(Function)}.


/*******************************************
 *
 *                              *MAFOOL_BEHE*
 *
 *******************************************/
mafool_behe('مفعول به'Tree1,TreeMafool), Sex ,Number,Known,_,_,_,_, LastUnitType,
FirstUnitType<--(
       reduntant_tarkeeb(Tree1),
       mafool_behe(TreeMafool,Sex ,Number,Known,_,_,_,_, LastUnitType,  FirstUnitType)
,!.

mafool_behe('مفعول به'Tree), Sex ,Number,_Known,_,_,_,_, LastUnitType,  FirstUnitType--(
<
       gomla_enn(Tree,_NextConnectionDegree),
              {LastUnitType = [null,null,null],  FirstUnitType = ['فعل',null,null,{[
              {Sex  = 'حيا د',Number = 'nutral.!, {'

mafool_behe('مفعول به'Tree), Sex , Number ,Known,_,_,_,_, LastUnitType,  FirstUnitType-(
<-
```

```
       tarkeeb_eddafy((Tree), Sex , Number , Known,_,_,_,_, _Contain_Dameer,
LastUnitType,  FirstUnitType,'all') ,!.

mafool_behe('مفعول به'TreeTarkeebWasfy), Sex ,Number ,Known,_,_,_,_, LastUnitType,
FirstUnitType<--(
       tarkeeb_wasfy(TreeTarkeebWasfy, Sex ,Number,Known,_,_,LastUnitType,FirstUnitType)
,!.

mafool_behe('مفعول به'TreeTameez), _Sex ,_Number ,_Known,_,_,_,_, LastUnitType,
FirstUnitType<--(
       tarkeeb_tameezy(TreeTameez, LastUnitType,  FirstUnitType), !.

mafool_behe('مفعول به'TreeTarkeebBadal), Sex ,Number,Known,_,_,_,_, LastUnitType,
FirstUnitType<--(
       tarkeeb_badal(TreeTarkeebBadal, Sex ,Number, Known, _, _,_,_, LastUnitType,
FirstUnitType) ,!.

mafool_behe('مفعول به'TreeMaarefa), Sex,Num ,Known,_,_,_,_, LastUnitType,
FirstUnitType<--(
       maarefa_connected(TreeMaarefa, Sex,Num, Known, _Function , _,_,_, _MaarefaType,
_AllUnitsAdVerbed,LastUnitType, FirstUnitType) ,!.

mafool_behe('مفعول به'TreeMaarefa), SEX , NUM ,Known,_,_,_,_, LastUnitType,
FirstUnitType<--(
       maarefa(TreeMaarefa, SEX , NUM , Known, _, _,_,_ , _NounSpecial, AdVerb, _Type,
_Modaf),
              {LastUnitType = [Known,AdVerb,_X],  FirstUnitType = [Known,AdVerb,_Y]} ,!.

mafool_behe('مفعول به'TreeNakera), Sex ,Num , Known , _ ,_,_,_, LastUnitType,
FirstUnitType<--(
       nakera(TreeNakera, Sex ,Num ,Known,_,_,_,_, _NounSpecial, AdVerb,_Modaf),
              {LastUnitType = [Known,AdVerb,_X],  FirstUnitType = [Known,AdVerb,_Y]}.

/*********************************************
 *
 *                            *COMPLENTRY*
 *
 *********************************************/

       /*     ========================================    */
       /*     =========      complentry_simple     =========      */
       /*     ========================================    */

complentry_simple_A('مكمل جملة'TreeHarfGar ,Tree2), _,_,_Known,_,_,_,_,
_LastUnitType,FirstUnitType, TermType, LastTermType,
FirstTermType,RefuseFaaelInFollowingParse<--(
       harf_gar(TreeHarfGar , _,_,_,_,_,_,_,_HarfGar_Class),
       gomla_enn(Tree2,_ConnectinDegree),
              {FirstUnitType = [ ' ,'حرف_جرnull,_X,{[
              {TermType = ' ,'شبه جملة أنLastTermType ,'شبه جملة أن' = FirstTermType =
,'شبه جملة أن'}
              {RefuseFaaelInFollowingParse = 'yes'} ,!.

complentry_simple_A('مكمل جملة'TreeShepehGomla), _,_,Known,_,_,_,_, LastUnitType,
FirstUnitType, TermType , LastTermType, FirstTermType,RefuseFaaelInFollowingParse<--(
       shepeh_gomla(TreeShepehGomla, _, _, Known, _,_,_,_, LastUnitType, FirstUnitType,
'all' , LastTermType),
              {TermType = ' ,'شبه جملةFirstTermType ,'حرف_جر' =
RefuseFaaelInFollowingParse = 'no.{'
```

```
complentry_simple_B('مكمل جملة - ضمير شأن للتوكيد'(Tree1), _,_,Known,_,_,_,_,
LastUnitType, FirstUnitType, TermType , LastTermType,
FirstTermType,RefuseFaaelInFollowingParse)-->
       dameer_monfasel(Tree1, _SEX,  _NUM, _, _, _,_,_),
              {Known = ',{'ضمير_منفصل'},
              {LastUnitType = [',ضمير_منفصل'null,_X,{ ]
              {FirstUnitType = [',ضمير_منفصل'null,_X,{ ]
              {TermType = ' ,'ضمير_منفصلFirstTermType ,'ضمير_منفصل' = LastTermType =
,{'ضمير_منفصل'},
              {RefuseFaaelInFollowingParse = 'yes'}.

complentry_simple_C('مكمل جملة'(TreeTarkeebEddafy), _, _, Known, _, _,_,_, LastUnitType,
FirstUnitType,TermType, LastTermType, FirstTermType, RefuseFaaelInFollowingParse<--(
       tarkeeb_eddafy(TreeTarkeebEddafy, _Sex ,_Number ,Known,_,_,_,_,_, LastUnitType,
FirstUnitType, 'all'),
              {is_unit_type(FirstUnitType,',{(ظرف'},
              {RefuseFaaelInFollowingParse = 'no'},
              {TermType = ' ,'تـركيب إضافي'LastTermType ,'تـركيب إضافي' = FirstTermType =
'تـركيب إضافي'} ,!.

complentry_simple_C('مكمل جملة'(TreeTarkeebWasfy), _, _, Known, _, _,_,_, LastUnitType,
FirstUnitType, TermType, LastTermType, FirstTermType,RefuseFaaelInFollowingParse<--(
       tarkeeb_wasfy(TreeTarkeebWasfy, _SEX,_NUM, Known, Function,
_,LastUnitType,FirstUnitType),
              {Function = ',{'نصب أو جر'},
              {RefuseFaaelInFollowingParse = 'no'},
              {TermType = ' ,'تـركيب وصفي'LastTermType ,'تـركيب وصفي' = FirstTermType =
'تـركيب وصفي'} ,!.

complentry_simple_C('مكمل جملة'(TreeNakera), _, _, Known, _, _,_,_,LastUnitType,
FirstUnitType,TermType, LastTermType, FirstTermType,RefuseFaaelInFollowingParse<--(
       nakera(TreeNakera, _Sex ,_,Known,Function,_,_,_, _NounSpecial, AdVerb,Modaf),
              {LastUnitType = [Known,AdVerb,Modaf] ,  FirstUnitType =
[Known,AdVerb,Modaf]},
              {Function = ' ;'نصبFunction,{'نصب أو جر' =
              {RefuseFaaelInFollowingParse = 'no'},
              {TermType = ' ,'نكرةLastTermType ,'نكرة' = FirstTermType.{'نكرة' =

complentry_simple(Tree, _, _, Known, _, _,_,_, LastUnitType, FirstUnitType,TermType,
LastTermType, FirstTermType,RefuseFaaelInFollowingParse)-->
       complentry_simple_A(Tree, _, _, Known, _, _,_,_, LastUnitType,
FirstUnitType,TermType, LastTermType, FirstTermType,RefuseFaaelInFollowingParse).
complentry_simple(Tree, _, _, Known, _, _,_,_, LastUnitType, FirstUnitType,TermType,
LastTermType, FirstTermType,RefuseFaaelInFollowingParse)-->
       complentry_simple_B(Tree, _, _, Known, _, _,_,_, LastUnitType,
FirstUnitType,TermType, LastTermType, FirstTermType,RefuseFaaelInFollowingParse).
complentry_simple(Tree, _, _, Known, _, _,_,_, LastUnitType, FirstUnitType,TermType,
LastTermType, FirstTermType,RefuseFaaelInFollowingParse)-->
       complentry_simple_C(Tree, _, _, Known, _, _,_,_, LastUnitType,
FirstUnitType,TermType, LastTermType, FirstTermType,RefuseFaaelInFollowingParse).

       /*     ========================================    */
       /*     =========      complentry_simple2    =========      */
       /*     ========================================    */

complentry_simple2('مكمل جملة'(TreeTarkeebEddafy), _, _, Known, _, _,_,_, LastUnitType,
FirstUnitType,TermType, LastTermType, FirstTermType, RefuseFaaelInFollowingParse<--(
       tarkeeb_eddafy(TreeTarkeebEddafy, _Sex ,_Number ,Known,_,_,_,_,_, LastUnitType,
FirstUnitType, 'all'),
              {RefuseFaaelInFollowingParse = 'no'},
```

```
                {TermType = ' ,'تـركيب إضافي'LastTermTypeتـركيب إضافي' = FirstTermType =
'تـركيب إضافي'} ,!.

complentry_simple2('مكمل جملة'TreeTarkeebWasfy), _, _, Known, _, _,_,_, LastUnitType,
FirstUnitType,TermType, LastTermType, FirstTermType, RefuseFaaelInFollowingParse<--(
        tarkeeb_wasfy(TreeTarkeebWasfy, _SEX,_NUM, Known, Function,
_,LastUnitType,FirstUnitType),
                {RefuseFaaelInFollowingParse = 'no'},
                {Function \== ',نصب أو جر'},
                {TermType = ' ,'تـركيب وصفي'LastTermTypeتـركيب وصفي' = FirstTermType =
'تـركيب وصفي'} ,!.

complentry_simple2('مكمل جملة'TreeTameez), _, _, _Known, _, _,_,_, LastUnitType,
FirstUnitType,TermType, LastTermType, FirstTermType, RefuseFaaelInFollowingParse<--(
        tarkeeb_tameezy(TreeTameez, LastUnitType,FirstUnitType),
                {RefuseFaaelInFollowingParse = 'no'},
                {TermType = ' ,'تـركيب تمييزي'LastTermTypeتـركيب تمييزي' = FirstTermType =
'تـركيب تمييزي'} ,!.'

complentry_simple2('مكمل جملة'TreeNakera), _, _, Known, _, _,_,_, LastUnitType,
FirstUnitType,TermType,LastTermType, FirstTermType, RefuseFaaelInFollowingParse<--(
        nakera(TreeNakera, _Sex ,_,Known,_,_,_,_, _NounSpecial, AdVerb,_Modaf),
                {LastUnitType = [Known,AdVerb,_X] ,  FirstUnitType = [Known,AdVerb,_Y]},
                {RefuseFaaelInFollowingParse = 'no'},
                {TermType = ' ,'نـكرة'LastTermTypeنـكرة' = 'نـكرة' = FirstTermType.!, {

complentry_simple2('مكمل جملة'TreeMaarefa), _, _, Known, _, _,_,_, LastUnitType,
FirstUnitType,TermType, LastTermType, FirstTermType, RefuseFaaelInFollowingParse<--(
        maarefa(TreeMaarefa, _Sex, _Number,Known, _Function, _,_,_ , _NounSpecial, AdVerb,
_Type, _Modaf),
                {LastUnitType = [Known,AdVerb,_X] ,  FirstUnitType = [Known,AdVerb,_Y]},
                {RefuseFaaelInFollowingParse = 'no'},
                {TermType = ' ,'معرفة'LastTermTypeمعرفة' = 'معرفة' = FirstTermType.{'معرفة'

        /*      ========================================== */
        /*      =========      complentry_simple_all  =========      */
        /*      ========================================== */
complentry_simple_all(Tree, _, _, Known, _, _,_,_, LastUnitType, FirstUnitType,TermType,
LastTermType, FirstTermType,RefuseFaaelInFollowingParse)-->
        complentry_simple(Tree, _, _, Known, _, _,_,_, LastUnitType,
FirstUnitType,TermType, LastTermType, FirstTermType,RefuseFaaelInFollowingParse).
complentry_simple_all(Tree, _, _, Known, _, _,_,_, LastUnitType, FirstUnitType,TermType,
LastTermType, FirstTermType,RefuseFaaelInFollowingParse)-->
        complentry_simple2(Tree, _, _, Known, _, _,_,_, LastUnitType,
FirstUnitType,TermType, LastTermType, FirstTermType,RefuseFaaelInFollowingParse).

        /*      ==========================================       */
        /*      =========      complentry_Compund_all =========      */
        /*      ==========================================       */

complentry_Compund_all('مكمل جملة مركب'TreeSimple,TreeFollowing), _,_,Known,_,_,_,_,
LastUnitType2, FirstUnitType1,TermType, LastTermType,
FirstTermType,RefuseFaaelInFollowingParse)-->
        complentry_simple_all(TreeSimple, _, _, Known, _, _,_,_, _LastUnitType1,
FirstUnitType1,_TermType1, _LastTermType1, FirstTermType1,RefuseFaaelInFollowingParse1),
                {FirstTermType = FirstTermType1},
                {FirstTermType \== ',شبه جملة أن'},
        complentry_Compund_all(TreeFollowing,_, _, _, _, _,_,_, LastUnitType2,
_FirstUnitType2,_TermType2, LastTermType2, _FirstTermType2,RefuseFaaelInFollowingParse2),
                {(RefuseFaaelInFollowingParse1 = 'yes' ; RefuseFaaelInFollowingParse2 =
'yes') ->
```

```
                        RefuseFaaelInFollowingParse = 'yes'
                 ;
                        RefuseFaaelInFollowingParse = 'no'
                },
                {LastTermType = LastTermType2},
/*              {refused_unit_decompos(LastUnitType1, FirstUnitType2)},*/
                {TermType = '.!, {'مـركب

complentry_Compund_all('مكمل جملة مـركب'TreeSimple,TreeFollowing), _,_,Known,_,_,_,_,
LastUnitType2, FirstUnitType1,TermType, LastTermType,
FirstTermType,RefuseFaaelInFollowingParse)-->
        complentry_simple_all(TreeSimple, _, _, Known, _, _,_,_, _LastUnitType1,
FirstUnitType1, _TermType1, _LastTermType1, FirstTermType1,RefuseFaaelInFollowingParse1),
                {FirstTermType = FirstTermType1},
                {FirstTermType \== ',{'أن شبـه جملة
        complentry_simple_all(TreeFollowing,_, _, _, _, _,_,_, LastUnitType2,
_FirstUnitType2, _TermType2, LastTermType2,
_FirstTermType2,RefuseFaaelInFollowingParse2),
                {(RefuseFaaelInFollowingParse1 = 'yes' ; RefuseFaaelInFollowingParse2 =
'yes') ->
                        RefuseFaaelInFollowingParse = 'yes'
                 ;
                        RefuseFaaelInFollowingParse = 'no'
                },
                {LastTermType = LastTermType2},
/*              {FirstTermType2 \== '/*,{'أن شبـه جملة
/*              {refused_unit_decompos(LastUnitType1, FirstUnitType2)},*/
                {TermType  = '.{'مـركب

        /*      ========================================    */
        /*      =========      complentry_Compund    =========      */
        /*      ========================================    */

complentry_Compund('مكمل جملة مـركب'TreeSimple,TreeFollowing), _,_,Known,_,_,_,_,
LastUnitType2, FirstUnitType1,TermType, LastTermType,
FirstTermType,RefuseFaaelInFollowingParse)-->
        complentry_simple(TreeSimple, _, _, Known, _, _,_,_, _LastUnitType1,
FirstUnitType1,_TermType1, _LastTermType1, FirstTermType1,RefuseFaaelInFollowingParse1),
                {FirstTermType = FirstTermType1},
                {FirstTermType \== ',{'أن شبـه جملة
        complentry_Compund(TreeFollowing,_, _, _, _, _,_,_, LastUnitType2,
_FirstUnitType2,_TermType2, LastTermType2, _FirstTermType2,RefuseFaaelInFollowingParse2),

                {(RefuseFaaelInFollowingParse1 = 'yes' ; RefuseFaaelInFollowingParse2 =
'yes') ->
                        RefuseFaaelInFollowingParse = 'yes'
                 ;
                        RefuseFaaelInFollowingParse = 'no'
                },
                {LastTermType = LastTermType2},
/*              {refused_unit_decompos(LastUnitType1, FirstUnitType2)},*/
                {TermType = '.!, {'مـركب

complentry_Compund('مكمل جملة مـركب'TreeSimple,TreeFollowing), _,_,Known,_,_,_,_,
LastUnitType2, FirstUnitType1,TermType, LastTermType,
FirstTermType,RefuseFaaelInFollowingParse)-->
        complentry_simple(TreeSimple, _, _, Known, _, _,_,_, _LastUnitType1,
FirstUnitType1,_TermType1 , _LastTermType1, FirstTermType1,RefuseFaaelInFollowingParse1),
                {FirstTermType = FirstTermType1},
                {FirstTermType \== ',{'أن شبـه جملة
        complentry_simple(TreeFollowing,_, _, _, _, _,_,_, LastUnitType2,
_FirstUnitType2,_TermType2, LastTermType2, _FirstTermType2,RefuseFaaelInFollowingParse2),
                {(RefuseFaaelInFollowingParse1 = 'yes' ; RefuseFaaelInFollowingParse2 =
'yes') ->
                        RefuseFaaelInFollowingParse = 'yes'
                 ;
```

```
                RefuseFaaelInFollowingParse = 'no'
            },
            {LastTermType = LastTermType2},
        /*  {LastTermType \== '/*,{'أن شبه جملة
/*          {refused_unit_decompos(LastUnitType1, FirstUnitType2)},*/
            {TermType = '.{'مركب

        /*  ============================================        */
        /*  =========      complentry_Connected  =========      */
        /*  ============================================        */

complentry_Connected('(')'مكمل جملة معطوفة'Tree1,TreeConnector,Tree2),_, _, Known, _,
_,_,_, LastUnitType2, FirstUnitType1,AcceptAllTypes,TermType, LastTermType ,
FirstTermType,RefuseFaaelInFollowingParse<--(
        complentry_unit(Tree1, _, _, Known, _, _,_,_, _LastUnitType1,
FirstUnitType1,AcceptAllTypes, TermType1 , LastTermType1,
FirstTermType1,RefuseFaaelInFollowingParse1),
            {FirstTermType = FirstTermType1},
        connector(TreeConnector,'all'),
        complentry_Connected(Tree2, _, _, _Known, _, _,_,_, LastUnitType2,
_FirstUnitType2,'yes', _TermType2, LastTermType2,
FirstTermType2,RefuseFaaelInFollowingParse2),
            {(RefuseFaaelInFollowingParse1 = 'yes' ; RefuseFaaelInFollowingParse2 =
'yes') ->
                RefuseFaaelInFollowingParse = 'yes'
             ;
                RefuseFaaelInFollowingParse = 'no'
            },
            {LastTermType = LastTermType2},
            {TermType1 \== ',{'أن شبه جملة
            {LastTermType1 = '<-'في إضافي تركيب
                FirstTermType2 \== 'في إضافي تركيب
            ;
                true
            },
            {TermType = '.{'معطوف

complentry_Connected('(')'مكمل جملة معطوفة'Tree1,TreeConnector,Tree2),_, _, Known, _,
_,_,_, LastUnitType2, FirstUnitType1,AcceptAllTypes,TermType, LastTermType ,
FirstTermType,RefuseFaaelInFollowingParse<--(
        complentry_unit(Tree1, _, _, Known, _, _,_,_, _LastUnitType1,
FirstUnitType1,AcceptAllTypes, TermType1, LastTermType1,
FirstTermType1,RefuseFaaelInFollowingParse1),
            {FirstTermType = FirstTermType1},
        connector(TreeConnector,'all'),
        complentry_unit(Tree2, _, _, _Known, _, _,_,_, LastUnitType2,
_FirstUnitType2,'yes', TermType2 , LastTermType2,
FirstTermType2,RefuseFaaelInFollowingParse2),
            {(RefuseFaaelInFollowingParse1 = 'yes' ; RefuseFaaelInFollowingParse2 =
'yes') ->
                RefuseFaaelInFollowingParse = 'yes'
             ;
                RefuseFaaelInFollowingParse = 'no'
            },
            {LastTermType = LastTermType2},
            {TermType1 \== ' ,'أن شبه جملة TermType2,{'أن شبه جملة'==\
            {LastTermType1 = '<-'في إضافي تركيب
                (FirstTermType2 \== ' ,'في إضافي تركيب FirstTermType2('معرفة'==\
            ;
                true
            },
            {TermType = '.{'معطوف
```

```
        /*      ====================================*/
        /*      =========      complentry_unit       =========      */
        /*      ====================================*/
complentry_unit(Tree, _,_,Known,_,_,_,_, LastUnitType, FirstUnitType, 'yes', TermType ,
LastTermType, FirstTermType,RefuseFaaelInFollowingParse)-->
        complentry_Compund_all(Tree,_, _, Known, _, _,_,_, LastUnitType,
FirstUnitType,TermType, LastTermType, FirstTermType,RefuseFaaelInFollowingParse).
complentry_unit(Tree, _,_,Known,_,_,_,_, LastUnitType, FirstUnitType, 'yes', TermType ,
LastTermType, FirstTermType,RefuseFaaelInFollowingParse)-->
        complentry_simple_all(Tree, _, _, Known, _, _,_,_, LastUnitType,
FirstUnitType,TermType, LastTermType, FirstTermType,RefuseFaaelInFollowingParse).

complentry_unit(Tree, _,_,Known,_,_,_,_, LastUnitType, FirstUnitType, 'no',TermType,
LastTermType, FirstTermType,RefuseFaaelInFollowingParse)-->
        complentry_Compund(Tree,_, _, Known, _, _,_,_, LastUnitType,
FirstUnitType,TermType, LastTermType, FirstTermType,RefuseFaaelInFollowingParse).
complentry_unit(Tree, _,_,Known,_,_,_,_, LastUnitType, FirstUnitType, 'no',TermType,
LastTermType, FirstTermType,RefuseFaaelInFollowingParse)-->
        complentry_simple(Tree, _, _, Known, _, _,_,_, LastUnitType,
FirstUnitType,TermType, LastTermType, FirstTermType,RefuseFaaelInFollowingParse).

        /*      ================================    */
        /*      =========      complentry     =========      */
        /*      ================================    */
complentry('مكمل جملة'Tree1,Tree2), _,_,Known,_,_,_,_, LastUnitType, FirstUnitType,
AcceptAllTypes, RefuseFaaelInFollowingParse,NextConnectionDegree<--(
        reduntant_tarkeeb(Tree1),
        complentry(Tree2, _,_,Known,_,_,_,_, LastUnitType, FirstUnitType, AcceptAllTypes,
RefuseFaaelInFollowingParse,NextConnectionDegree) ,!.

complentry(Tree, _,_,Known,_,_,_,_, LastUnitType, FirstUnitType, AcceptAllTypes,
RefuseFaaelInFollowingParse,NextConnectionDegree)-->
        complentry_Connected(Tree,_, _, Known, _, _,_,_, LastUnitType,
FirstUnitType,AcceptAllTypes,_TermType, _LastTermType,
_FirstTermType,RefuseFaaelInFollowingParse),
                {NextConnectionDegree = 'all'}.

complentry(Tree, _,_,Known,_,_,_,_, LastUnitType, FirstUnitType, AcceptAllTypes,
RefuseFaaelInFollowingParse,NextConnectionDegree)-->
        complentry_unit(Tree, _,_,Known,_,_,_,_, LastUnitType, FirstUnitType,
AcceptAllTypes,_TermType, _LastTermType, _FirstTermType, RefuseFaaelInFollowingParse),
                {NextConnectionDegree = 'all'}.

/*********************************************
 *
 *                              *ESSM_SELA*
 *
 *********************************************/
essm_sela('اسم صلة'Word), Sex<--(_,_,_,_,_ ,
        [Word, ' ,'اسم_صلةSex, _ , _, _, _,_,_NounSpecial, _,_AdVerb, _Modaf, _VerbPrefix,
_VerbSuffix.[

/*********************************************
 *
 *                              *ESSM_SWAA*
 *
 *********************************************/
essm_swaa('كلمة سواء'Word<--((
        [Word, '_ ,'اسمSex, _ , _, _, _,_,_NounSpecial, _,_AdVerb, _Modaf, _VerbPrefix,
_VerbSuffix,[
                {Word = 'سواء'}.


/*********************************************
 *
 *                              *ESHARA_NOUN*
```

```
 *
 *********************************************/
eshara_noun('اسم اشارة'Word), Sex,Number<--(_,_,_,_,
       [Word, ' ,'اشارةح_Sex, Number , _, _, _,_,_NounSpecial, _,_AdVerb, _Modaf,
_VerbPrefix, _VerbSuffix.[


/*********************************************
 *
 *                            *TARKEEM_NOTATION*
 *
 *********************************************/
tarkeem_notation('علامة ترقيم'Word<--((
       [Word, '_ ,'ترقيمSex, _Number , _, _, _,_,_NounSpecial, _,_AdVerb, _Modaf,
_VerbPrefix, _VerbSuffix.[

/*********************************************
 *
 *                            *ANY_WORDS*
 *
 *********************************************/

any_words('كلمة عارضة'Word<--((
       [Word, _, _Sex, _Number , _, _, _,_,_NounSpecial, _, _AdVerb, _Modaf, _VerbPrefix,
_VerbSuffix].

any_words('كلمة عارضة'Word,Tree<--((
       [Word, _, _Sex, _Number , _, _, _,_,_NounSpecial, _, _AdVerb, _Modaf, _VerbPrefix,
_VerbSuffix],
       any_words(Tree).

/*********************************************
 *
 *                            *REDUNTANT_WORDS*
 *
 *********************************************/
reduntant_words('كلمة عارضة'Word<--((
       [Word, '_ ,'عارضSex, _Number , _, _, _,_,_NounSpecial, _, _AdVerb, _Modaf,
_VerbPrefix, _VerbSuffix.[

reduntant_words('كلمة عارضة'Word,Tree<--((
       [Word, '_ ,'عارضSex, _Number , _, _, _,_,_NounSpecial, _, _AdVerb, _Modaf,
_VerbPrefix, _VerbSuffix,[
       reduntant_words(Tree).


/*********************************************
 *
 *                            *REDUNTANT_TARKEEB*
 *
 *********************************************/
reduntant_tarkeeb('جملة اعتراضية'TreeTarkeemNotation1,TreeWord,TreeTarkeemNotation2<--((
       tarkeem_notation(TreeTarkeemNotation1),
       any_words(TreeWord),
       tarkeem_notation(TreeTarkeemNotation2).

reduntant_tarkeeb('جملة اعتراضية'TreeWord<--((
       reduntant_words(TreeWord).
```

```
/*********************************************
 *
 *                            *DIGIT_PERCENTAGE_UNIT*
 *
 *********************************************/
digit_percentage_unit('نسبة مئوية'Word),   Sex, Number, Known, Function, _,_,_ ,
NounSpecial , AdVerb<--(
       [Word,'نسبة رقم',Sex  ,Number,Known,Function,_,_,NounSpecial, _, AdVerb, _Modaf,
_VerbPrefix, _VerbSuffix.[

digit_percentage_unit('نسبة مئوية'Word1,TreeHarfGar,Word2),   Sex, Number, Known,
Function, _,_,_ , NounSpecial , AdVerb<--(
       [Word1,'نسبة رقم',Sex  ,Number,Known,Function,_,_,NounSpecial, _, AdVerb, _Modaf,
_VerbPrefix, _VerbSuffix,[
       harf_gar(TreeHarfGar , _,_,_,_,_,_,_,_HarfGar_Class),
       [Word2,'نسبة رقم'_,Sex2  ,_Number2,_Known2,_Function2,_,_,_NounSpecial2, _,
_AdVerb2, _Modaf2, _VerbPrefix2, _VerbSuffix2.[


/*********************************************
 *
 *                            *DIGIT_UNIT_SIMPLE*
 *
 *********************************************/
digit_unit_simple('وحدة رقمية'Word),   Sex, Number, Known, Function, _,_,_ ,
NounSpecial , AdVerb<--(
       [Word,'وحدة رقم',Sex  ,Number,Known,Function,_,_,NounSpecial, _, AdVerb, _Modaf,
_VerbPrefix, _VerbSuffix.[

digit_unit_simple('وحدة رقمية'Word1,TreeHarfGar,Word2),   Sex, Number, Known, Function,
_,_,_ , NounSpecial , AdVerb<--(
       [Word1,'وحدة رقم',Sex  ,Number,Known,Function,_,_,NounSpecial, _, AdVerb, _Modaf,
_VerbPrefix, _VerbSuffix,[
       harf_gar(TreeHarfGar , _,_,_,_,_,_,_,_HarfGar_Class),
       [Word2,'وحدة رقم'_,Sex2  ,_Number2,_Known2,_Function2,_,_,_NounSpecial2, _,
_AdVerb2, _Modaf2, _VerbPrefix2, _VerbSuffix2.[

/*********************************************
 *
 *                            *DIGIT_UNIT*
 *
 *********************************************/

digit_unit(Tree, Sex, Number, Known, Function, _,_,_ , NounSpecial , AdVerb)-->
 digit_percentage_unit(Tree, Sex, Number, Known, Function, _,_,_ , NounSpecial , AdVerb)
,!.

digit_unit(Tree, Sex, Number, Known, Function, _,_,_ , NounSpecial , AdVerb)-->
 digit_unit_simple(Tree, Sex, Number, Known, Function, _,_,_ , NounSpecial , AdVerb).


/*********************************************
 *
 *                            *MAAREFA*
 *
 *********************************************/
maarefa_by_alf_lam('معرفة'Word,Sex) , Sex , Number ,'معرفة' , Function,_,_,_, NounSpecial,
AdVerb, Type , Modaf<--(
```

```
       [Word,',''اسم'|Sex,Number,'معرفة',Function,_,_,NounSpecial, _, AdVerb, Modaf,
_VerbPrefix, _VerbSuffix,[
              {Type = 'معرفة'}.

maarefa_alm('معرفة_علم - أعلام متتابعة'Word1,Tree,Sex) , Sex , Number,'معرفة',
Function,_,_,_, NounSpecial, AdVerb, Type ,Modaf<--(
       [Word1,',''اسم'|Sex,Number,'علم',Function,_,_,NounSpecial, _, AdVerb,  Modaf,
_VerbPrefix, _VerbSuffix,[
       maarefa_alm(Tree , Sex , Number , _Known , Function,_,_,_, NounSpecial, AdVerb,
Type , _Modaf2) ,!.

maarefa_alm('معرفة_علم'Word,Sex) , Sex , Number,'معرفة' , Function,_,_,_, NounSpecial,
AdVerb, Type ,Modaf<--(
       [Word,',''اسم'|Sex,Number,'علم',Function,_,_,NounSpecial, _, AdVerb,  Modaf,
_VerbPrefix, _VerbSuffix,[
              {Type = 'علم'}, !.

maarefa_alm('معرفة_علم_لاتيني'Word,Sex) , Sex , Number,'معرفة' , Function,_,_,_,
NounSpecial, AdVerb, Type ,Modaf)-->
       [Word,',''لاتيني'Sex,Number,'علم',Function,_,_,NounSpecial, _, AdVerb,  Modaf,
_VerbPrefix, _VerbSuffix,[
              {Type = 'علم'}.


maarefa('معرفة'Word1,Tree,Sex) , Sex , Number,'معرفة' , Function,_,_,_, NounSpecial,
AdVerb, Type, Modaf<--(
       [Word1,',''اسم'|Sex,Number,'معرفة',Function,_,_,NounSpecial, _, AdVerb, Modaf,
_VerbPrefix, _VerbSuffix,[
       maarefa_alm(Tree , _Sex , _Number , _Known, _Function,_,_,_, _NounSpecial,
_AdVerb, Type ,_Modaf2) ,!.
maarefa(Tree , Sex , Number , Known, Function,_,_,_, NounSpecial, AdVerb, Type, Modaf)-->
       maarefa_by_alf_lam(Tree , Sex , Number , Known, Function,_,_,_, NounSpecial,
AdVerb, Type, Modaf) ,!.
maarefa(Tree , Sex , Number , Known, Function,_,_,_, NounSpecial, AdVerb, Type, Modaf)-->
       maarefa_alm(Tree , Sex , Number , Known, Function,_,_,_, NounSpecial, AdVerb,
Type, Modaf).

/*********************************************
 *
 *                            *MAAREFA_CONNECTED*
 *
 *********************************************/
maarefa_connected('معارف معطوفة'TreeFirst,TreeConnector,TreeSecnd), Sex,Number, Known,
Function1, _,_,_, Type, AllUnitsAdVerbed, LastUnitType, FirstUnitType<--(
       maarefa(TreeFirst,Sex,Number,Known, Function1, _,_,_, _NounSpecial, AdVerb1,
Type1, _Modaf1),
              {FirstUnitType = [Known,AdVerb1,_X]},
              {AllUnitsAdVerbed = 'yes' ->
                     AdVerb1 = 'نعت'
               ;
                     true
              },
       connector(TreeConnector,'all'),
       maarefa(TreeSecnd,Sex,Number,Known, _Function2, _,_,_, _NounSpecial,AdVerb2,
Type2, _Modaf2),
              {LastUnitType = [Known,AdVerb2,_Y]},
              {AllUnitsAdVerbed = 'yes' ->
                     AdVerb2 = 'نعت'
```

```
                ;
                        true
                },
                {Type1 = Type2}, {Type = Type2}.


maarefa_connected('معارف معطوفة'TreeFirst,TreeConnector,TreeSecnd), Sex,Number, Known,
Function1, _,_,_, Type, AllUnitsAdVerbed , LastUnitType2, FirstUnitType<--(
        maarefa(TreeFirst,Sex,Number,Known, Function1, _,_,_ , _NounSpecial, AdVerb1,
Type1, _Modaf1),
                {FirstUnitType = [Known,AdVerb1,_X]},
                {AllUnitsAdVerbed = 'yes' ->
                        AdVerb1 = 'نعت'
                ;
                        true
                },
        connector(TreeConnector,'all'),
        maarefa_connected(TreeSecnd,Sex,Number,Known, _Function2, _,_,_, Type2,
AllUnitsAdVerbed, LastUnitType2, _FirstUnitType2),
                {Type1 = Type2}, {Type = Type2}.

/*********************************************
 *
 *                              *NAKERA*
 *
 *********************************************/
nakera('نكرة'Word,Sex) , Sex , Number ,'نكرة',Function,_,_,Wazn , NounSpecial,
AdVerb,Modaf<--(
        [Word,',اسم'Sex,Number,'نكرة',Function,_,_,NounSpecial, Wazn, AdVerb, Modaf,
_VerbPrefix, _VerbSuffix.[

/*********************************************
 *
 *                              *NAKERA_CONTIGUSE*
 *
 *********************************************/
nakera_contiguse('نكرات متعددة'TreeNakera1,Tree2), Sex1, Number1, Known, Function1,
_,_,Wazn1, LastUnitType2, FirstUnitType<--(
        nakera(TreeNakera1, Sex1  ,Number1, Known,Function1 ,_,_,Wazn1, _NounSpecial1,
AdVerb1,Modaf1),
                {FirstUnitType = [Known,AdVerb1,Modaf1]},
        nakera_contiguse(Tree2, _Sex2 ,_Number2,Known,_Function2, _ , _ , _Wazn2 ,
LastUnitType2, FirstUnitType2),
                {unite_modaf(FirstUnitType2)},
                /*{Modaf0 = ' ,'مضاف له'FirstUnitType2 = [_Known0,_AdVerb0,Modaf0/*,{[
                {Wazn1 \== '.!, {'أفعل

nakera_contiguse('نكرات متعددة'TreeNakera1,TreeNakera2), Sex1, Number1, Known,
Function1, _,_,Wazn1, LastUnitType, FirstUnitType<--(
        nakera(TreeNakera1, Sex1  ,Number1, Known,Function1 ,_,_,Wazn1, _NounSpecial1,
AdVerb1,Modaf1),
                {FirstUnitType = [Known,AdVerb1,Modaf1]},
        nakera(TreeNakera2, _Sex2 ,_Number2,Known,Function2,_,_,_ ,    _NounSpecial2,
AdVerb2,Modaf2),
                {LastUnitType = [Known,AdVerb2,Modaf2]},
                {Modaf2 = ',{'مضاف له
                {AdVerb2 \== ',{'نعت
                {Function2 \== ',{'نصب
                {Wazn1 \== '.{'أفعل
```

```
/*********************************************
 *
 *                              *NAKERA_CONNECTED*
 *
 *********************************************/
nakera_connected('نكرات معطوفة'TreeNakera1,TreeConnector,TreeNakera2), Sex1, Number1,
Known, Function1, _,_,Wazn ,LastUnitType, FirstUnitType<--(
       nakera(TreeNakera1, Sex1 ,Number1,Known,Function1,_,_,Wazn, _NounSpecial
,AdVerb1,Modaf1),
              {FirstUnitType = [Known,AdVerb1,Modaf1]},
       connector(TreeConnector,'all'),
       nakera(TreeNakera2, _Sex2 ,_Number2,Known,_Function2,_,_,_ , _NounSpecial,
AdVerb2,Modaf2),
              {LastUnitType = [Known,AdVerb2,Modaf2]}.

nakera_connected('نكرات معطوفة'TreeNakera1,TreeConnector,TreeNakera2), Sex1, Number1,
Known, Function1, _,_,Wazn ,LastUnitType2, FirstUnitType<--(
       nakera(TreeNakera1, Sex1 ,Number1,Known,Function1,_,_,Wazn, _NounSpecial
,AdVerb1,Modaf1),
              {FirstUnitType = [Known,AdVerb1,Modaf1]},
       connector(TreeConnector,'all'),
       nakera_connected(TreeNakera2, _Sex2 ,_Number2,Known,_Function2, _,_,_Wazn2
,LastUnitType2, _FirstUnitType2).


/*********************************************
 *
 *                              *NAKERA_COMPUND*
 *
 *********************************************/
nakera_compund('نكرات'TreeNakera1,TreeNakera2),   Sex1, Number1, Known, Function1,
_,_,Wazn, LastUnitType2, FirstUnitType<--(
       nakera(TreeNakera1, Sex1 ,Number1,Known,Function1,_,_,Wazn , _NounSpecial,
AdVerb1,Modaf1),
              {FirstUnitType = [Known,AdVerb1,Modaf1]},
       nakera_connected(TreeNakera2, _Sex2 ,_Number2,Known,_Function2,_,_,_,
LastUnitType2, FirstUnitType2),
       {FirstUnitType2 = [_Known2,_AdVerb2,Modaf2] , Modaf2 = '.'مضاف له}
nakera_compund('نكرات'TreeNakera),   Sex1, Number1, Known, Function1, _,_,Wazn,
LastUnitType, FirstUnitType<--(
       nakera_connected(TreeNakera, Sex1 ,Number1,Known,Function1,_,_,Wazn, LastUnitType,
FirstUnitType).


/*********************************************
 *
 *                              *DAMEER_MOTASEL*
 *
 *********************************************/
dameer_motasel('ضمير_متصل'Word), Sex, Num, _, _, _,_,_, RefuseFaaelInFollowingParse,
AcceptAdVerb , IsMafool<--(
       [Word,','ضمير_متصلSex, Num,_,_,_,_,_NounSpecial, _, _AdVerb,_Modaf,  _VerbPrefix,
_VerbSuffix,[
              {       Word = '<-' و ا
                      RefuseFaaelInFollowingParse = 'yes'
               ;
                      RefuseFaaelInFollowingParse = 'no'
              },
              {       Word = '<-' ي
                      AcceptAdVerb = 'yes'
               ;
                      AcceptAdVerb = 'no'
              },
```

```
                {       Word = 'هـا' <-'
                        IsMafool = 'yes'
                 ;
                        IsMafool = 'no'
                }.

/*********************************************
 *
 *                          *DAMEER_MONFASEL*
 *
 *********************************************/
 dameer_monfasel('ضمير_منفصل'(Word), SEX,  NUM<--(_,_, _, _, _ ,
       [Word,'ضمير_من',SEX, NUM,_,_,_,_,_NounSpecial, _, _AdVerb,_Modaf,  _VerbPrefix,
_VerbSuffix.[


/*********************************************
 *
 *                          *TARKEEB_EDDAFY*
 *
 *********************************************/

       /*     ===================================================       */
       /*     =========      tarkeeb_eddafy_simple_dameer  =========      */
       /*     ===================================================       */
tarkeeb_eddafy_simple_dameer('إضافي'(تركيب TreeNakera,TreeDameerMotasel) , Sex , Number ,
'dameer',_,_,_,_, 'yes', LastUnitType, FirstUnitType<--(
       nakera_compund(TreeNakera,   Sex, Number, _Known, _, _,_,_, _LastUnitType,
FirstUnitType),
       dameer_motasel(TreeDameerMotasel, _,_, _, _,
_,_,_,_RefuseFaaelInFollowingParse,_AcceptAdVerb , _IsMafool),
               {LastUnitType = ['ضمير_متصل',null,null.!, {[

tarkeeb_eddafy_simple_dameer('إضافي'(تركيب TreeNakera,TreeDameerMotasel) , Sex , Number ,
'dameer',_,_,_,_, 'yes', LastUnitType, FirstUnitType<--(
       nakera_contiguse(TreeNakera, Sex, Number, _Known, _, _,_,_,_LastUnitType,
FirstUnitType),
       dameer_motasel(TreeDameerMotasel, _,_, _, _,
_,_,_,_RefuseFaaelInFollowingParse,_AcceptAdVerb , _IsMafool),
               {LastUnitType = ['ضمير_متصل',null,null.!, {[

tarkeeb_eddafy_simple_dameer('إضافي'(تركيب TreeNakera,TreeDameerMotasel) , Sex , Number ,
'dameer',_,_,_,_, 'yes',LastUnitType, FirstUnitType<--(
       nakera(TreeNakera,   Sex, Number, Known, _, _,_,_ , _NounSpecial ,AdVerb,_Modaf),
               {FirstUnitType = [Known,AdVerb,_X]},
       dameer_motasel(TreeDameerMotasel, _,_, _, _,
_,_,_,_RefuseFaaelInFollowingParse,_AcceptAdVerb , _IsMafool),
               {LastUnitType = ['ضمير_متصل',null,null.{[

       /*     ================================================    */
       /*     =========      tarkeeb_eddafy_simple_digit   =========      */
       /*     ================================================    */
tarkeeb_eddafy_simple_digit('إضافي'(تركيب TreeNakera,TreeDigit ) , Sex , Number ,
Known,_,_,_,_, 'null'  ,  LastUnitType2, FirstUnitType<--(
       nakera(TreeNakera,   Sex, Number, Known, _Function1, _,_,_ , _NounSpecial1 ,
AdVerb1, Modaf1),
               {FirstUnitType = [Known,AdVerb1,Modaf1]},
       tarkeeb_eddafy_simple_digit(TreeDigit , _Sex , _Number , _Known,_,_,_,_, _,
LastUnitType2, _FirstUnitType2) ,!.

tarkeeb_eddafy_simple_digit('إضافي'(تركيب TreeDigit,TreeNakera) , Sex , Number ,
Known,_,_,_,_, 'null'  ,  LastUnitType, FirstUnitType<--(
```

```
       digit_unit(TreeDigit,   _Sex, _Number, _Known, _Function, _,_,_ , _NounSpecial1 ,
_AdVerb1),
       nakera_contiguse(TreeNakera, Sex, Number, _Known2, _Function2, _,_,
_Wazn,LastUnitType, _FirstUnitType2),
               {Known  =    ',{ 'معرفة
               {FirstUnitType  =  [','وحدة رقمية'null,null.!, {[

tarkeeb_eddafy_simple_digit(')'إضافي تركيبTreeDigit,TreeNakera) , Sex , Number ,
Known,_,_,_,_, 'null'  ,  LastUnitType, FirstUnitType<--(
       digit_unit(TreeDigit,   _Sex, _Number, _Known, _Function, _,_,_ , _NounSpecial1 ,
_AdVerb1),
       nakera(TreeNakera,   Sex, Number, Known2, _Function2, _,_,_ , _NounSpecial2 ,
AdVerb2,Modaf2),
               {Known  =    ',{ 'معرفة
               {LastUnitType  =  [Known2,AdVerb2,Modaf2]},
               {FirstUnitType  =  [','وحدة رقمية'null,null.!, {[

tarkeeb_eddafy_simple_digit(')'إضافي تركيبTreeDigit,TreeMaarefa) , Sex , Number ,
Known,_,_,_,_, 'null'  ,  LastUnitType, FirstUnitType<--(
       digit_unit(TreeDigit,   _Sex, _Number, _Known, _Function, _,_,_ , _NounSpecial1 ,
_AdVerb1),
       maarefa(TreeMaarefa,  Sex ,Number , Known2, _, _,_,_ , _NounSpecial2 , AdVerb2,
_Type, Modaf2),
               {Known  =    ',{ 'معرفة
               {LastUnitType  =  [Known2,AdVerb2,Modaf2]},
               {FirstUnitType =  [','وحدة رقمية'null,null.!, {[

tarkeeb_eddafy_simple_digit(')'إضافي تركيبTreeDigit) , Sex , Number , Known,_,_,_,_,
'null'  ,  LastUnitType, FirstUnitType<--(
       digit_percentage_unit(TreeDigit,   Sex, Number, _Known, _Function, _,_,_ ,
_NounSpecial1 , _AdVerb1),
               {Known  =    ',{ 'معرفة
               {FirstUnitType  =  [','نسبة مئوية'null,null] , LastUnitType =
FirstUnitType}.

       /*     ================================================    */
       /*     =========      tarkeeb_eddafy_simple_zarf    =========      */
       /*     ================================================    */

tarkeeb_eddafy_simple_zarf(')'إضافي تركيبTreeZarf,Tree2) , 'nutral' , 'nutral' ,
'nutral',_,_,_,_, ContainDameer, LastUnitType,  FirstUnitType<--(
       zarf(TreeZarf,   _, _, _, _, _,_,_),
              {FirstUnitType  = [' ,'ظرفnull,null,{[
       gomla_enn(Tree2,_NextConnectionDegree),
              {ContainDameer = 'no' },
              {LastUnitType = ['','جملة إنnull,null.!, {[

tarkeeb_eddafy_simple_zarf(')'إضافي تركيبTreeZarf,TreeDameerMotasel) , 'nutral' ,
'nutral' , 'nutral','_,_,_,_, 'yes', LastUnitType,  FirstUnitType<--(
       zarf(TreeZarf,   _, _, _, _, _,_,_),
              {FirstUnitType  = [' ,'ظرفnull,null,{[
       dameer_motasel(TreeDameerMotasel, _,_, _, _,
_,_,_,_RefuseFaaelInFollowingParse,_AcceptAdVerb , _IsMafool),
       {LastUnitType = ['','ضمير_متصلnull,null.!, {[

tarkeeb_eddafy_simple_zarf(')'إضافي تركيبTreeZarf,TreeTarkeeb) , 'nutral' , 'nutral' ,
'nutral',_,_,_,_, ContainDameer, LastUnitType,  FirstUnitType<--(
       zarf(TreeZarf,   _, _, _, _, _,_,_),
```

```
                {FirstUnitType  = [' ,'ظرفnull,null,{[
        tarkeeb_eddafy_simple(TreeTarkeeb, _,_, _, _, _,_,_, ContainDameer, LastUnitType,
_FirstUnitType),!.

tarkeeb_eddafy_simple_zarf(')إضافي تـركـيب'TreeZarf,TreeTarkeebWasfy) , 'nutral' , 'nutral'
, _,_,_,_,_,_, 'null', LastUnitType, FirstUnitType<--(
        zarf(TreeZarf,   _, _, _, _, _,_,_),
                {FirstUnitType  = [' ,'ظرفnull,null,{[
        tarkeeb_wasfy(TreeTarkeebWasfy, _,_, ',_ ,_ ,'معرفـةLastUnitType,_FirstUnitType.!, (

tarkeeb_eddafy_simple_zarf(')إضافي تـركـيب'TreeZarf,TreeEsharaNoun) , 'nutral' , 'nutral' ,
_,_,_,_,_,_, 'null',LastUnitType, FirstUnitType<--(
        zarf(TreeZarf,   _, _, _, _, _,_,_),
                {FirstUnitType  = [' ,'ظرفnull,null,{[
        eshara_noun(TreeEsharaNoun, _Sex,_Number, _, _, _,_,_),
                {LastUnitType = [' ,'اسم اشارةnull,null.!, {[

tarkeeb_eddafy_simple_zarf(')إضافي تـركـيب'TreeZarf,TreeMaarefa) , 'nutral' , 'nutral' ,
_,_,_,_,_,_, 'null',LastUnitType, FirstUnitType<--(
        zarf(TreeZarf,   _, _, _, _, _,_,_),
                {FirstUnitType  = [' ,'ظرفnull,null,{[
        maarefa(TreeMaarefa, _,_, Known2, _, _,_,_ , _NounSpecial, AdVerb, _Type, Modaf2),
                {LastUnitType = [Known2,AdVerb,Modaf2]} ,!.

tarkeeb_eddafy_simple_zarf(')إضافي تـركـيب'TreeZarf,TreeNakera) , 'nutral' , 'nutral' ,
_,_,_,_,_,_, 'null', LastUnitType, FirstUnitType<--(
        zarf(TreeZarf,   _, _, _, _, _,_,_),
                {FirstUnitType  = [' ,'ظرفnull,null,{[
        nakera(TreeNakera, _,_, Known2, _, _,_,_ , _NounSpecial, AdVerb2,Modaf2),
                {LastUnitType = [Known2,AdVerb2,Modaf2]}.

        /*      ======================================================      */
        /*      =========      tarkeeb_eddafy_simple_normal  =========      */
        /*      ======================================================      */

tarkeeb_eddafy_simple_normal(')إضافي تـركـيب'TreeNakera,TreeMaarefa) , Sex , Number ,
' ,_,_,_,'مـعـرفـة'null', LastUnitType, FirstUnitType)-->
        nakera_compund(TreeNakera,   Sex, Number, _Known, _, _,_,_, _LastUnitType,
FirstUnitType),
        maarefa(TreeMaarefa, _,_, Known, _, _,_,_ , _NounSpecial, AdVerb, _Type, Modaf2),
                {LastUnitType = [Known,AdVerb,Modaf2]} ,!.

tarkeeb_eddafy_simple_normal(')إضافي تـركـيب'TreeNakera,TreeMaarefa) , Sex , Number ,
' ,_,_,_,'مـعـرفـة'null', LastUnitType2, FirstUnitType1)-->
        nakera_compund(TreeNakera,   Sex, Number, _, _, _,_,_, _LastUnitType1,
FirstUnitType1),
        maarefa_connected(TreeMaarefa, _,_, _, _, _,_,_, _MaarefaType,
_AllUnitsAdVerbed,LastUnitType2, _FirstUnitType2) ,!.

tarkeeb_eddafy_simple_normal(')إضافي تـركـيب'TreeNakera,TreeTarkeebWasfy) , Sex , Number ,
Known,_,_,_,_, 'null', LastUnitType2, FirstUnitType1<--(
        nakera_contiguse(TreeNakera, Sex, Number, Known, _Function, _,_, _Wazn,
_LastUnitType1, FirstUnitType1),
        tarkeeb_wasfy(TreeTarkeebWasfy, _,_, ',_ ,_ ,'معرفـةLastUnitType2,_FirstUnitType2 (
.!,

tarkeeb_eddafy_simple_normal(')إضافي تـركـيب'TreeNakera,TreeMaarefa) , Sex , Number ,
Known,_,_,_,_, 'null', LastUnitType, FirstUnitType1<--(
```

```
        nakera_contiguse(TreeNakera, Sex, Number, Known, _Function, _,_, _Wazn,
_LastUnitType1, FirstUnitType1),
        maarefa(TreeMaarefa, _,_, _Known2, _, _,_,_ , _NounSpecial, AdVerb, _Type,
Modaf2),
                {LastUnitType = [Known,AdVerb,Modaf2]} ,!.

tarkeeb_eddafy_simple_normal('إضافي تركيب')(TreeNakera,TreeTarkeebWasfy) , Sex , Number ,
', _,_,_,'معرفة','null', LastUnitType,  FirstUnitType<--(
        nakera(TreeNakera,   Sex, Number, Known, _, _,_,_ , _NounSpecial, AdVerb,Modaf),
                {FirstUnitType = [Known,AdVerb,Modaf]},
        tarkeeb_wasfy(TreeTarkeebWasfy, _,_, ', _,_','معرفة'LastUnitType,_FirstUnitType.!, (

tarkeeb_eddafy_simple_normal('إضافي تركيب')(TreeNakera,TreeTarkeebBadal) , Sex , Number ,
', _,_,_,'معرفة','null', LastUnitType,   FirstUnitType<--(
        nakera(TreeNakera,   Sex, Number, Known, _, _,_,_ , _NounSpecial, AdVerb,Modaf),
                {FirstUnitType = [Known,AdVerb,Modaf]},
        tarkeeb_badal(TreeTarkeebBadal, _,_, _, _, _,_,_, LastUnitType, _FirstUnitType)
,!.

tarkeeb_eddafy_simple_normal('إضافي تركيب')(TreeNakera,TreeTarkeebWasfy) , Sex , Number ,
', _,_,_,'معرفة','null', LastUnitType ,  FirstUnitType<--(
        nakera(TreeNakera,   Sex, Number, Known, _, _,_,_ , _NounSpecial, AdVerb,Modaf),
                {FirstUnitType = [Known,AdVerb,Modaf]},
        tarkeeb_wasfy(TreeTarkeebWasfy, _,_, ', _,_','نكرة'LastUnitType,_FirstUnitType.!, (

tarkeeb_eddafy_simple_normal('إضافي تركيب')(TreeNakerat) , Sex , Number , Known,_,_,_,_,
'null', LastUnitType, FirstUnitType<--(
        nakera_contiguse(TreeNakerat, Sex, Number, Known, _Function, _,_, _Wazn,
LastUnitType, FirstUnitType) ,!.

tarkeeb_eddafy_simple_normal('إضافي تركيب')(TreeNakera,TreeMaarefa) , Sex , Number ,
', _,_,_,'معرفة','null', LastUnitType, FirstUnitType)-->
        nakera(TreeNakera,   Sex, Number, Known1, _Function, _,_,_ , _NounSpecial1 ,
AdVerb1,Modaf1),
                {FirstUnitType = [Known1,AdVerb1,Modaf1]},
        maarefa(TreeMaarefa, _,_, Known2, _, _,_,_ , _NounSpecial2 , AdVerb2, _Type,
Modaf2),
                {LastUnitType = [Known2,AdVerb2,Modaf2]}.


        /*      ============================================        */
        /*      =========      tarkeeb_eddafy_simple  =========      */
        /*      ============================================        */

/* remove this rule to "tarkeeb_eddafy_simple_dameer" section */
tarkeeb_eddafy_simple( 'إضافي تركيب')(Tree,TreeTarkeebWasfy), Sex , Number ,
Known,_,_,_,_, ContainDameer, LastUnitType2,  FirstUnitType1<--(
        tarkeeb_eddafy_simple_dameer(Tree,   Sex, Number, Known, _, _,_,_, ContainDameer,
_LastUnitType1,  FirstUnitType1),
        tarkeeb_wasfy(TreeTarkeebWasfy, _,_, ', _,_','معرفة'LastUnitType2,_FirstUnitType2 (
.!,

/* remove this rule to "tarkeeb_eddafy_simple_dameer" section */
tarkeeb_eddafy_simple( 'إضافي تركيب')(Tree,TreeMaarefa), Sex , Number , Known,_,_,_,_,
ContainDameer, LastUnitType,  FirstUnitType<--(
        tarkeeb_eddafy_simple_dameer(Tree,   Sex, Number, Known, _, _,_,_, ContainDameer,
_LastUnitType,  FirstUnitType),
        maarefa(TreeMaarefa, _,_, Known2, _, _,_,_ , _NounSpecial, AdVerb2, _Type,
Modaf2),
                {AdVerb2 = ',{'نعت
```

```
                {LastUnitType = [Known2,AdVerb2,Modaf2]} ,!.

tarkeeb_eddafy_simple(Tree, Sex , Number , Known,_,_,_,_, ContainDameer, LastUnitType,
FirstUnitType)-->
       tarkeeb_eddafy_simple_dameer(Tree,   Sex, Number, Known, _, _,_,_, ContainDameer,
LastUnitType,  FirstUnitType) ,!.
tarkeeb_eddafy_simple(Tree, Sex , Number , Known,_,_,_,_, ContainDameer, LastUnitType,
FirstUnitType)-->
       tarkeeb_eddafy_simple_digit(Tree,   Sex, Number, Known, _, _,_,_, ContainDameer,
LastUnitType,  FirstUnitType) ,!.
tarkeeb_eddafy_simple(Tree, Sex , Number , Known,_,_,_,_, ContainDameer, LastUnitType,
FirstUnitType)-->
       tarkeeb_eddafy_simple_zarf(Tree, Sex , Number , Known,_,_,_,_, ContainDameer,
LastUnitType,  FirstUnitType) ,!.
tarkeeb_eddafy_simple(Tree, Sex , Number , Known,_,_,_,_, ContainDameer, LastUnitType,
FirstUnitType)-->
       tarkeeb_eddafy_simple_normal(Tree, Sex , Number , Known,_,_,_,_, ContainDameer,
LastUnitType,  FirstUnitType),!.

/*
tarkeeb_eddafy_simple('تركيب إضافي',TreeZarf,Tree) , 'nutral' , 'nutral' ,
'nutral',_,_,_,_, ContainDameer, LastUnitType,  FirstUnitType<--(
       zarf(TreeZarf,   _, _, _, _, _,_,_),
              {FirstUnitType  = ['ظرف',null,null]},
       gomla_enn(Tree,_NextConnectionDegree),
              {ContainDameer = 'no' },
              {LastUnitType = ['جملة إن',null,null]}.
*/


       /*     ================================================    */
       /*     =========      tarkeeb_eddafy_connected      =========      */
       /*     ================================================    */

tarkeeb_eddafy_connected('تراكيب إضافية معطوفة',Tree, TreeConnector,TreeMaarefa),
Sex, Number, Known,_,_,_,_, Contain_Dameer, LastUnitType2,
FirstUnitType1,ConnectionType)-->
       tarkeeb_eddafy_simple(Tree ,Sex, Number, Known,_,_,_,_, Contain_Dameer,
_LastUnitType1,  FirstUnitType1),
       connector(TreeConnector,ConnectionType),
       maarefa_connected(TreeMaarefa, _,_, _, _, _,_,_, _MaarefaType, _AllUnitsAdVerbed,
LastUnitType2, _FirstUnitType2).

tarkeeb_eddafy_connected('تراكيب إضافية معطوفة',Tree, TreeConnector ,Tree2), Sex,
Number, Known,_,_,_,_, Contain_Dameer, LastUnitType2,  FirstUnitType1,ConnectionType)-->
       tarkeeb_eddafy_simple(Tree ,Sex, Number, Known,_,_,_,_, Contain_Dameer,
_LastUnitType1,  FirstUnitType1),
       connector(TreeConnector,ConnectionType),
       tarkeeb_eddafy_connected(Tree2, _, _, _,_,_,_,_, _, LastUnitType2,
_FirstUnitType2,ConnectionType).

tarkeeb_eddafy_connected('تراكيب إضافية معطوفة',Tree, TreeConnector,Tree2), Sex,
Number, Known,_,_,_,_, Contain_Dameer, LastUnitType2,  FirstUnitType1,ConnectionType)-->
       tarkeeb_eddafy_simple(Tree ,Sex, Number, Known,_,_,_,_, Contain_Dameer,
_LastUnitType1,  FirstUnitType1),
       connector(TreeConnector, ConnectionType),
       tarkeeb_eddafy_simple(Tree2, _, _, _,_,_,_,_, _, LastUnitType2,  _FirstUnitType2).
/*             {FirstUnitType2 \== LastUnitType1}.*/

tarkeeb_eddafy_connected('تراكيب إضافية معطوفة',Tree, TreeConnector,TreeMaarefa,
TreeConnector2, Tree3), Sex, Number, Known,_,_,_,_, Contain_Dameer, LastUnitType3,
FirstUnitType1,ConnectionType)-->
       tarkeeb_eddafy_simple(Tree ,Sex, Number, Known,_,_,_,_, Contain_Dameer,
_LastUnitType1,  FirstUnitType1),
       connector(TreeConnector,ConnectionType),
       maarefa(TreeMaarefa, _,_, _Known2, _, _,_,_, _NounSpecial, _AdVerb2,
_Type,_Modaf2),
       connector(TreeConnector2,ConnectionType),
```

```
       tarkeeb_eddafy_simple(Tree3 ,_Sex3, _Number3, _Known3,_,_,_,_, _Contain_Dameer,
LastUnitType3,  _FirstUnitType3).

tarkeeb_eddafy_connected('تراكيب إضافية معطوفة'Tree, TreeConnector,TreeMaarefa),
Sex, Number, Known,_,_,_,_, Contain_Dameer, LastUnitType2,
FirstUnitType1,ConnectionType)-->
       tarkeeb_eddafy_simple(Tree ,Sex, Number, Known,_,_,_,_, Contain_Dameer,
_LastUnitType1,  FirstUnitType1),
       connector(TreeConnector,ConnectionType),
       maarefa(TreeMaarefa, _,_, Known2, _, _,_,_, _NounSpecial, AdVerb2, _Type,Modaf2),
              {LastUnitType2 = [Known2,AdVerb2,Modaf2]}.

tarkeeb_eddafy_connected('تراكيب إضافية معطوفة'Tree,
TreeConnector,TreeTarkeebWasfy), Sex, Number, Known,_,_,_,_, Contain_Dameer,
LastUnitType2, FirstUnitType1,ConnectionType<--(
       tarkeeb_eddafy_simple(Tree ,Sex, Number, Known,_,_,_,_, Contain_Dameer,
_LastUnitType1,  FirstUnitType1),
       connector(TreeConnector,ConnectionType),
       tarkeeb_wasfy(TreeTarkeebWasfy, _,_, ',_ ,_','معرفة'LastUnitType2,_FirstUnitType2.(

tarkeeb_eddafy_connected('تراكيب إضافية معطوفة'Tree,
TreeConnector,TreeTarkeebWasfy), Sex, Number, Known,_,_,_,_, Contain_Dameer,
LastUnitType2, FirstUnitType1,ConnectionType<--(
       tarkeeb_eddafy_simple(Tree ,Sex, Number, Known,_,_,_,_, Contain_Dameer,
_LastUnitType1,  FirstUnitType1),
       connector(TreeConnector,ConnectionType),
       tarkeeb_wasfy(TreeTarkeebWasfy, _,_, ',_ ,_','نكرة'LastUnitType2,_FirstUnitType2.(

       /*     ===================================*/
       /*     =========      tarkeeb_eddafy =========      */
       /*     ===================================*/

tarkeeb_eddafy((Tree), Sex , Number , Known,_,_,_,_, Contain_Dameer, LastUnitType,
FirstUnitType,ConnectionType)-->
       tarkeeb_eddafy_unit((Tree), Sex , Number , Known,_,_,_,_, Contain_Dameer,
LastUnitType,  FirstUnitType,ConnectionType).


tarkeeb_eddafy_unit((Tree), Sex , Number , Known,_,_,_,_, Contain_Dameer, LastUnitType,
FirstUnitType,ConnectionType)-->
       {ConnectionType \== 'no'},
       tarkeeb_eddafy_connected(Tree, Sex , Number , Known,_,_,_,_, Contain_Dameer,
LastUnitType,  FirstUnitType, ConnectionType) ,!.
tarkeeb_eddafy_unit((Tree), Sex , Number , Known,_,_,_,_, Contain_Dameer, LastUnitType,
FirstUnitType,_ConnectionType)-->
       tarkeeb_eddafy_simple(Tree, Sex , Number , Known,_,_,_,_, Contain_Dameer,
LastUnitType,  FirstUnitType).


/*********************************************
 *
 *                          *TARKEEB_WASFY*
 *
 *********************************************/

       /*     ============================================       */
       /*     =========      tarkeeb_wasfy_simple   =========      */
       /*     ============================================       */
tarkeeb_wasfy_simple('تركيب وصفي'TreeFirst, TreeOntherWasfy) , Sex1 , Number ,
,'معرفة'Function,_,LastUnitType,FirstUnitType)-->
       maarefa(TreeFirst, Sex1,Number, Known1, Function, _,_,_ , _NounSpecial, AdVerb1,
_Type,Modaf1),
       tarkeeb_wasfy_simple(TreeOntherWasfy, Sex2,_, '_ ,'معرفة'Function2,
_,LastUnitType,_FirstUnitType2,(
              {FirstUnitType = [Known1,AdVerb1,Modaf1]},
```

```
                {Sex1 = Sex2; Sex1 = ' ;'حيـاد'Sex2.!, {'حيـاد' =

tarkeeb_wasfy_simple('تـركيـب وصفـي'TreeFirst, TreeSecond) , Sex1 , Number ,
,'مـعرفـة'Function,_,LastUnitType,FirstUnitType)-->
        maarefa(TreeFirst,  Sex1, Number ,Known1, Function,   _,_,_ , _NounSpecial,
AdVerb1, _Type, Modaf1),
        maarefa(TreeSecond, Sex2,_Number2,Known2, _Function2, _,_,_,  _NounSpecial2,
AdVerb2, _Type2,Modaf2),
                {FirstUnitType = [Known1,AdVerb1,Modaf1]},
                {LastUnitType = [Known2,AdVerb2,Modaf2]},
                {AdVerb2 = ',{'نـعت
                {Sex1 = Sex2; Sex1 = ' ;'حيـاد'Sex2.! , {'حيـاد' =

tarkeeb_wasfy_simple('تـركيـب وصفـي'TreeFirst, TreeDameerMotasel,TreeSecond) , Sex1 ,
Number,'مـعرفـة',Function,_,LastUnitType,FirstUnitType)-->
        maarefa(TreeFirst, Sex1, Number,Known1, Function, _,_,_ , _NounSpecial, AdVerb1,
_Type,_Modaf1),
                {FirstUnitType = [Known1,AdVerb1,_X]},
        dameer_motasel(TreeDameerMotasel, _,_, _, _,
_,_,_,_RefuseFaaelInFollowingParse,_AcceptAdVerb , _IsMafool),
        maarefa(TreeSecond, Sex2,_Num,Known2, _Function2, _,_,_, _NounSpecial, AdVerb2,
_Type,_Modaf2),
                {LastUnitType = [Known2,AdVerb2,_X2]},
                {AdVerb2 = ',{'نـعت
                {Sex1 = Sex2; Sex1 = ' ;'حيـاد'Sex2.!, {'حيـاد' =

tarkeeb_wasfy_simple('تـركيـب وصفـي'TreeFirst, TreeSecond) , Sex1 , Number ,
,'نـكرة'Function,_,LastUnitType,FirstUnitType)-->
        nakera_contiguse(TreeFirst, Sex1,Number,  _Known1 , Function, _,_,Wazn1 ,
_LastUnitType, FirstUnitType),
        nakera(TreeSecond, Sex2,_Number, Known2, _Function2, _,_,Wazn2 , _NounSpecial,
AdVerb2,_Modaf2),
                {LastUnitType = [Known2,AdVerb2,_X2]},
                {AdVerb2 = ',{'نـعت
                {Wazn1 \== ' ,'أفـعل'Wazn2,{'أفـعل' ==\
                {Sex1 = Sex2; Sex1 = ' ;'حيـاد'Sex2.!, {'حيـاد' =


tarkeeb_wasfy_simple('تـركيـب وصفـي'TreeFirst, TreeSecond) , Sex1 , Number ,
,'نـكرة'Function,_,LastUnitType,FirstUnitType)-->
        nakera(TreeFirst, Sex1,Number,  Known1 , Function, _,_,Wazn1 , _NounSpecial,
AdVerb1,_Modaf1),
                {FirstUnitType = [Known1,AdVerb1,_X]},
        nakera(TreeSecond, Sex2,_Number, Known2, _Function2, _,_,Wazn2 , _NounSpecial,
AdVerb2,_Modaf2),
                {LastUnitType = [Known2,AdVerb2,_X2]},
                {AdVerb2 = ',{'نـعت
                {Wazn1 \== ' ,'أفـعل'Wazn2,{'أفـعل' ==\
                {Sex1 = Sex2; Sex1 = ' ;'حيـاد'Sex2.!, {'حيـاد' =

tarkeeb_wasfy_simple('تـركيـب وصفـي'TreeFirst, TreeOntherWasfy) , Sex1 , Number ,
,'نـكرة'Function,_,LastUnitType,FirstUnitType<--(
        nakera(TreeFirst, Sex1,Number, Known1, Function, _,_,Wazn , _NounSpecial,
AdVerb1,_Modaf1),
                {FirstUnitType = [Known1,AdVerb1,_X]},
                {Wazn \== ',{'أفـعل
```

```
       tarkeeb_wasfy_simple(TreeOntherWasfy, Sex2,_Number, '_',نـكرةFunction2,
_,LastUnitType,_FirstUnitType2,(
              {Sex1 = Sex2; Sex1 = ';حيـاد'}.Sex2حيـاد'='

       /*     ============================================       */
       /*     =========      tarkeeb_wasfy_connected      =========      */
       /*     ============================================       */
tarkeeb_wasfy_connected('تـراكـيب وصفـيـة مـعطوفـة'Tree1,TreeConnector,Tree2), Sex1 ,
Number, Known ,Function,_,LastUnitType2,FirstUnitType1<--(
       tarkeeb_wasfy_simple(Tree1, Sex1 , Number, Known
,Function,_,_LastUnitType1,FirstUnitType1),
              {Known = ',{'مـعرفـة
       connector(TreeConnector,'all'),
       tarkeeb_wasfy_connected(Tree2, Sex2 , _Number, _Known
,_Function,_,LastUnitType2,_FirstUnitType2),
              {Sex1 = Sex2; Sex1 = ';حيـاد'}.Sex2حيـاد'='

tarkeeb_wasfy_connected('تـراكـيب وصفـيـة مـعطوفـة'Tree1,TreeConnector,Tree2), Sex1 ,
Number, Known ,Function,_,LastUnitType,FirstUnitType<--(
       tarkeeb_wasfy_simple(Tree1, Sex1 , Number, Known
,Function,_,_LastUnitType1,FirstUnitType),
              {Known = ',{'مـعرفـة
       connector(TreeConnector,'all'),
       maarefa(Tree2, Sex2,_,     Known2, _Function2, _,_,_, _NounSpecial, AdVerb2 ,
_Type, _Modaf2),
              {LastUnitType = [Known2,AdVerb2,_X2]},
              {AdVerb2 = ',{'نـعت
              {Sex1 = Sex2; Sex1 = ';حيـاد'}.Sex2حيـاد'='

tarkeeb_wasfy_connected('تـراكـيب وصفـيـة مـعطوفـة'Tree1,TreeConnector,Tree2), Sex1 ,
Number, Known ,Function,_,LastUnitType,FirstUnitType<--(
       tarkeeb_wasfy_simple(Tree1, Sex1 , Number, Known
,Function,_,_LastUnitType1,FirstUnitType),
              {Known = ',{'مـعرفـة
       connector(TreeConnector,'all'),
              {AllUnitsAdVerbed = 'yes'},
       maarefa_connected(Tree2, Sex2,_Number, _Known2, _Function2, _,_,_, _Type,
AllUnitsAdVerbed,LastUnitType, _FirstUnitType2),
              {Sex1 = Sex2; Sex1 = ';حيـاد'}.Sex2حيـاد'='

tarkeeb_wasfy_connected('تـراكـيب وصفـيـة مـعطوفـة'Tree1,TreeConnector,Tree2), Sex1 ,
Number, Known ,Function,_,LastUnitType,FirstUnitType<--(
       tarkeeb_wasfy_simple(Tree1, Sex1 , Number, Known
,Function,_,_LastUnitType1,FirstUnitType),
              {Known = ',{'مـعرفـة
       connector(TreeConnector,'all'),
       tarkeeb_wasfy_simple(Tree2, Sex2, _Number, Known2
,_Function,_,LastUnitType,_FirstUnitType2),
              {Known2 = ',{'مـعرفـة
              {Sex1 = Sex2; Sex1 = ';حيـاد'}.Sex2حيـاد'='

tarkeeb_wasfy_connected('تـراكـيب وصفـيـة مـعطوفـة'Tree1,TreeConnector,Tree2), Sex1 ,
Number, Known ,Function,_,LastUnitType,FirstUnitType<--(
       maarefa(Tree1, Sex1,_,     Known1, _Function1, _,_,_, _NounSpecial, AdVerb1 ,
_Type, _Modaf1),
              {FirstUnitType = [Known1,AdVerb1,_X1]},
       connector(TreeConnector,'all'),
```

```
       tarkeeb_wasfy_simple(Tree2, Sex2 , Number, Known
,Function,_,LastUnitType,_FirstUnitType2),
               {Known = ',{'معرفة
               {Sex1 = Sex2; Sex1 = ' ;'حياد'Sex2.{'حياد' =

       /*      ===================================*/
       /*      =========      tarkeeb_wasfy  =========      */
       /*      ===================================*/
tarkeeb_wasfy(Tree , Sex, Number, Known,Function,_,LastUnitType,FirstUnitType)-->
       tarkeeb_wasfy_connected(Tree, Sex , Number, Known
,Function,_,LastUnitType,FirstUnitType) ,!.
tarkeeb_wasfy(Tree , Sex, Number, Known,Function,_,LastUnitType,FirstUnitType)-->
       tarkeeb_wasfy_simple(Tree, Sex , Number, Known
,Function,_,LastUnitType,FirstUnitType).

/*********************************************
 *
 *                            *TARKEEB_BADAL*
 *
 *********************************************/
tarkeeb_badal(')'تركيب بدلTreeEsharaNoun, TreeTarkeebWasfy) , Sex , Number, _,_,_,_,_,
LastUnitType,FirstUnitType<--(
       eshara_noun(TreeEsharaNoun, Sex,Number, _, _, _,_,_),
       {FirstUnitType = ['  ,'اشارة اسمnull,null,{[
       tarkeeb_wasfy(TreeTarkeebWasfy, _SEX,_NUM, ' ,_ ,'معرفة
,_LastUnitType,_FirstUnitType2.!, (

tarkeeb_badal(')'تركيب بدلTreeEsharaNoun, TreeSecond) , Sex , Number, _,_,_,_,_ ,
LastUnitType,FirstUnitType<--(
       eshara_noun(TreeEsharaNoun, Sex,Number, _, _, _,_,_),
       {FirstUnitType = ['  ,'اشارة اسمnull,null,{[
       maarefa(TreeSecond, _Sex1,_Num1, Known2, _, _,_,_ , _NounSpecial, AdVerb, Type,
_Modaf2),
       {LastUnitType = [Known2,AdVerb,_Y]},
       {Type = '.{'معرفة

/*********************************************
 *
 *                            *TARKEEB_TAMEEZY*
 *
 *********************************************/
tarkeeb_tameezy(')'تركيب تمييزيTreeNakera1 , Tree2) ,LastUnitType2,  FirstUnitType<--(
       nakera(TreeNakera1 , _SEX1 , _NUM1 , Known, _, _,_,'_ ,'أفعلNounSpecial1,
AdVerb1,Modaf1,(
       tarkeeb_eddafy(Tree2, _SEX2 , _NUM2 , _Known , _, _,_,_,_, LastUnitType2,
_FirstUnitType2,'all'),
       {FirstUnitType = [Known,AdVerb1,Modaf1]} ,!.

tarkeeb_tameezy(')'تركيب تمييزيTreeNakera1 , TreeNakera2) ,LastUnitType,  FirstUnitType-(
<-
       nakera(TreeNakera1 , _SEX1 , _NUM1 , Known, _, _,_,'_ ,'أفعلNounSpecial1,
AdVerb1,Modaf1,(
       nakera(TreeNakera2 , _SEX2 , _NUM2 , Known, _, _,_,_ , _NounSpecial2,
AdVerb2,Modaf2),
       {FirstUnitType = [Known,AdVerb1,Modaf1]},
       {LastUnitType = [Known,AdVerb2,Modaf2]}.

/*****************************************
*****************************************
*****************************************/
test2:-
```

```
        read(String),
        ( String = end_of_file ->
          write('end of file'),
          true
        ;
          (     setof(  (Tree),
                                s(Tree , String,[]),
                                Tree
                        ),
           write(Tree)
          )
        ;
          format("ERROR While parsing ~w",[String])
        ).

/*////////////////////////////////////////////
//  prolog predicates.
////////////////////////////////////////////*/


/**************************************************
*       *refused_unit_decompos
**************************************************/
refused_unit_decompos(LastUnitType, FirstUnitType) :-
        (
                is_unit_type(LastUnitType, '<- (' نـكرة')
                        reject_type(FirstUnitType,'(' مـعرفـة')
                ;
                        true
        ),
        (
                is_unit_type(LastUnitType, '<-(' نـكرة')
                        reject_type(FirstUnitType,'(' اسم  اشـارة')
                ;
                        true
        ),
        (
                is_unit_type(LastUnitType, '<-(' ضمير_مـتصل')
                        reject_type(FirstUnitType,'(' مـعرفـة')
                ;
                        true
        ),

        (       is_unit_type(LastUnitType, '<-(' نـكرة')
                        (
                                (unite_adverbed(FirstUnitType) ;
unite_modaf(FirstUnitType))->
                                        reject_type(FirstUnitType,'(' نـكرة')
                                ;
                                        true
                        )
                ;
                true
        ),

        (       ((is_unit_type(LastUnitType, ') , ((' مـعرفـة')unite_adverbed(FirstUnitType<-(((
                        reject_type(FirstUnitType,'(' مـعرفـة')
                ;
                        true
        ).

/**************************************************
*       validate_KhabarFree
**************************************************/
validate_KhabarFree(KhabarFree,FirstUnitType) :-
```

```
        (
                KhabarFree = 'no' ->
                        true
                ;
                        FirstUnitType = ['_|'ضمير_منفصلTail]
        ).


/**************************************************
*       validate_NumAgreement
**************************************************/
validate_NumAgreement(Agreement,NUM1,NUM2) :-
        (Agreement = 'no' ->
                true
          ;
                (NUM2 = 'nutral'; NUM1 = ' ;'مثنىNUM2 = NUM1(
        ).

/**************************************************
*       validate_faael_function
**************************************************/
validate_faael_function(Function) :-
        Function \== '.نصب أو جر'

/**************************************************
*       validate_faael_sex_agreement
**************************************************/
validate_faael_sex_agreement(VerbPrefix, SexFaael) :-
        (       VerbPrefix = '<-'ي
                 (SexFaael = ' ;'مذكرSexFaael('حيا د'=
                ;
                 true
        ),
        (       VerbPrefix = '<-'ت
                 (SexFaael = ' ;'مؤنثSexFaael('حيا د'=
                ;
                 true
        ).

/**************************************************
*       reject_type
**************************************************/
reject_type(UnitInfo,RejectedType) :-
        UnitInfo = [Type | _Tail],
        Type \== RejectedType.

/**************************************************
*       *is_unit_type
**************************************************/
is_unit_type(UnitInfo, TargetType) :-
        UnitInfo = [TargetType | _Tail].

/**************************************************
*       *unite_adverbed
**************************************************/
unite_adverbed(UnitInfo) :-
        UnitInfo = [_Type, '_','نعتModaf.[

/**************************************************
*       *unite_modaf
**************************************************/
unite_modaf(UnitInfo) :-
        UnitInfo = [_Type, _AdVerb, '.['مضاف لـه
```

REFERENCES

Abdel Aziz M., 1993 'كتاب مدخل إلي اللغة' - دار الفكر العربي

Ali N., 1988 'اللغة العربية و الحاسوب' - تعريب

Allen, J. Natural Language Understanding, second edition, The Benjamin/Cummings Publishing Company, 1995.

Bouma G., Koeling R., Nederhof M., and Van Noord G., Grammatical Analysis in a Spoken Dialogue System, Language and Cognition 5, Groningen, 1996.

Brame M., Base Generated Syntax, Noit Amrofer, Seattle, WA, 1978.

Bratko I., 'Prolog programming for Artificial Intelligence', Addison-Wesley publishing company, 1990.

Bresnan J., The Mental Representation of grammatical Relations, MIT Press, Cambridge, MA, 1983.

Chomsky N., Aspects of the Theory of Syntax, MIT Press, Cambridge, MA, 1965. Summarizes transformational theory as of 1965.

Colmerauer, A. Metamorphosis grammars. In L. Bloc (ed.) Natural Commmunication with Computers. Berlin: Springer-Verlag, 1978.

Convington M., Natural Language Processing for PROLOG Programmers, Prentice Hall 1994.

Dougherty R. Natural Language Computing: An English Generative Grammar in Prolog, Lawrence Erlbaum Associates, Inc., NJ, 1994.

Elsamahy A.M., Al-Hasanin Al-Barhamtoshy and Mohmed A. Madkour. An Arabic Morphological Analyser, Al-Azhar Engineering third international conference , December 18-21, 1993.

Feddag A., 'Arabic Morpho-syntax and semantic parsing', Department of computer science, university of Manchester, 3rd international conference on Multilingual, 10-12 Dec., 1992, Univ. of Durham.

Fillmore, C.J. The case for case. In E. Bach and R. Harms (eds.) Universals in Linguistic Theory. New York: Holt, Rinehart, and Winstone, 1-90, 1968.

Fillmore, C.J. The case for case reopened. In P. Cole and J. Sadock (eds.), 59-81, Syntax and Semantics 8: Grammatical Relations. New York: Academic Press, 1977.

Gazdar, G. Phrase Strcture Grammar. In P. Jacobson and G.K. Pullum (eds.) the nature of Syntactic Representation. Dordrecht: D. Reidel, 131-186, 1982.

Gazdar G. and Mellish C., Natural Language Processing in Prolog: An Introduction to Computational Linguistics, Addison-Wesley, 1990.

HARRIS M. 'Introduction to natural language processing', Reston publishing company, Reston, Virginia , 1985.

HLAL Y. 'Information system and Arabic : the use of Arabic in information system', Linguistics and Signal & information processing, A subsidiary of Haarper & Row publishing, Inc. 191-197, 1987.

Ismail M., 1995 دار إحياء الكتب العربية - 'قواعد النحو بأسلوب العصر'
Jespersen O., The Philosophy of grammar, Norton, New York, 1965.
Kaplan, R.M. Augmented transition networks as psychological models of sentence comprehension. Artif. Intell. 77-100 (1972).
Khayat M. Understanding Natural Arabic, in proceeding of the First KFUM workshop on information and computer science, Dhahran, Saudi Arabic, pp. k1-k4, 1996.
Kowalski R. Logic for Problem Solving, Elsevier North-Holland, New York, 1979.
LEECH G., 'semantic the study of meaning', Penguin Book Ltd., England, 1985.
Neidle, C. Lexical-Functional Grammar, Encyclopedia of Language and Linguistics, New York: Pergamon Press., 1994.
Newmeyer F., Linguistic Theory in America, Academic Press, New York, 1980. Provides an intellectual history of transformational grammar.
Pereira, F.C.N. Extraposition Grammars, AJCL 7, 4, 243-256, 1981.
Pereira F., and Warren D.H.D., Definite Clause Grammars for language analysis–A Survey of the Formalism and a Comparison with Augmented Transition Networks, In Readings in Natural Language Processing, Grosz, B., Jones K., Webber B. (Eds.), pp. 24-101, Morgan Kuffmann, 1986.
Pollard C. and Sag I., Head-Drive Phrase Structure Grammar. University of Chicago/CSLI, 1994.
Radford B., Transformational Syntax, Cambridge University Press, Cambridge, MA, 1981. Gives a textbook introduction to the extended standard theory.
Rafea A. and Shaalan K., Lexical Analysis of Inflected Arabic words using Exhaustive Search of an Augmented Transition Network, Software Practice and Experience, Vol. 23(6), pp. 567-588, John Wiley & sons, U.K., June 1993.
Sabatier P., Puzzle grammars, in Natural Language Understanding and logic Programming, V. Dahl and P. Saint-Dizier (eds.) Elsevier Science Publishers B.V. (North-Holland), 1985.
Sadler, L., New developments in LFG, In Keith Brown and Jim Miller, editors, Concise Encyclopedia of Syntactic Theories. Elsevier Science, Oxford. 1996.
SHAALAN K., "A Knowledge base system for understanding an inflected Arabic word", M.Sc. thesis 1989, Cairo university ISSR.
Shaalan K., Farouk A., Rafea A. Towards An Arabic Parser for Modern Scientific Text, In Proceeding of the 2nd Conference on Language Engineering, Egyptian Society of Language Engineering (ELSE), Egypt, April 18, 1999.
Shieber, S. M., Introduction to Unification-Based Approaches to Grammar, Center for the study of language and information, Stanford, 1986.
Van Riemsdijk H. and Williams E., introduction to the Theory of Grammar, MIT Press, Cambridge, MA, 1986.

Winograd, T. Language as a cognitive Process. Vol.1 : Syntax, Reading, MA: Addison-Wesley, 1983.

Woods, W.A. Transition network grammars for natural language analysis. Communication of the ACM 13, 591-606, 1970.

Zamora A., Zamora E., 'Development of natural language processing systems from a manager's prespective', chapter 10, 1994.

Zarri G., 'Natural language processing, associated with expert system', chapter 19, the handbook of Applied Expert systems , Edited by Liebwitz, CRC PCJI 1998.

قسم علوم الحاسب والمعلومات

معهد الدراسات و البحوث الإحصائية

جامعة القاهرة

تطوير محلل نحوي للغة العربية في نظام ترجمة آلية متعدد اللغات

إعـــداد

أحمـــــد فـــاروق أحـــــــمد

إشراف

أ.د. أحمـــــد رافـــع

وكيل كلية الحاسبات و المعلومات

جامعة القاهرة

د. خـــــالد شعـــلان

قسم علوم الحاسب-كلية الحاسبات و المعلومات

جامعة القاهرة

قدمت هذه الرسالة استكمالا لمتطلبات درجة الماجستير في علوم الحاسب

قسم علوم الحاسب و المعلومات معهد الدراسات و البحوث الإحصائية جامعة القاهرة

1999