# Natural Arabic Language Text Understanding



Al-Khonaizi, Mohammed Taqi

A thesis submitted in partial fulfillment of the requirements of the University of Greenwich for the award of the degree of Doctor of Philosophy

School of Computing and Mathematical Sciences
Faculty of Science and Engineering



March, 1999.

University of Greenwich

London

## Table of Contents

## List of Figures

## List of Tables

## Dedication

I would like to dedicate this humble work to my father and mother who put me on the first steps and supported me during the education path.

I would also like to dedicate this work to my wife and four children who accompanied me during this work's journey.

Finally I would dedicate this work to my nation which inspired the topic in me.

## Acknowledgements

Thank Providence and all those who participated in encouraging me to complete this work.

I would like to express my many thanks to the *Ministry of Finance and National Economy* of the State of Bahrain for their financial support for the early stages of this work. With their support I managed to overcome the major obstacles that eased my further steps.

Many thanks to the British Council and the British Airways, who provided some financial and logistic support during the late stages of this work. Their support contributed to making me to finish my work with fewer difficulties.

Special thanks to my first supervisor Dr. Ala Al-Zobaidie who devoted a lot of his precious time into providing the professional guidance and comments. His efforts greatly improved my knowledge and skills. I would also like to thank my second supervisors, Dr. Sati Mckanzi of Greenwich university and Dr. Mansoor Al Aali of the University of Bahrain who devoted a lot of their time to the discussions and reviews of my work. His support has enriched my knowledge and skills.

I extend my thanks to Mr. Nick wood, the Financial Management Information Systems Project Manager at the Ministry of Finance and National Economy in the State of Bahrain, who gave his valuable time and effort in reviewing the English language of this thesis.

I also have to express my appreciation to my colleagues and friends who boosted my morale during the progress of this work.

## Acronyms & Abbreviations

| Acronym | Description |
|---|---|
| Adj | Adjective |
| AI | Artificial Intelligence |
| ATEMT | Arabic-To-English Machine Translator |
| ATN | Augmented Transition Network |
| CBSUAL | Computer Based System for Understanding Arabic Language |
| CFG | Context Free Grammars |
| C-structure | Constituent structure |
| DB | Database |
| Det | Determinant |
| FST | Finite-State Transducers |
| F-structure | Functional structure |
| HPSG | Head-driven Phrase Structure Grammar |
| K-structure | Common sense knowledge structure |
| LFG | Lexical-Functional Grammar |
| MS | Microsoft |
| NAL | Natural Arabic Language |
| NLP | Natural Language Processing |
| NP | Noun Phrase |
| PP | Preposition Phrase |
| Pred | Predicate |
| RTN | Recursive Transition Network |
| S-structure | Semantic structure |
| STN | Simple Transition Network |
| TG | Transformational Grammar |
| VP | Verb Phrase |
| WH-ness | Questions start with Wh (e.g., What, Where, Who, When, etc) |
| XMA | Xerox Morphological Analyzer |

## Glossary of Arabic Words

| Arabic Word | Transliteral | Translation |
|---|---|---|
| ابتداء | Ibtida | Primacy |
| استثناء | Istithna | Exclusion |
| استدراك | Istidraj | Restriction |
| استفتاح | Istiftah | Inceptive |
| استفهام | Istiham | Interrogation |
| استقبال | Istiqbal | Future |
| اشارة | Ishareh | Pointer |
| اضراب | Idrab | Rectification |
| افعل التفضيل | Afaal Al Tafdeel | Preeminence |
| الآلة | Al Aaleh | Tool |
| الاسم | Ism | Noun |
| الاسماء الخمسة | Al Asmaa Al Khamsa | 5 Nouns |
| البدل | Badal | Substitute |
| الحال | Haal | Condition |
| الخبر | Khabar | Predicate |
| الزمان | Al Zaman | Time |
| الشرط | Al Shart | Condition |
| الصفة | Al Sifa | Adjective |
| الظرف | Al Zarf | Circumstance |
| العدد | Aladdad | Numeral |
| الفاعل | Al Fa el | Subject |
| الفعل | Al Fe el | Verbal |

| | | |
|---|---|---|
| الفعل المنفي | Al Fe el Al Manfi | Negated Verb |
| الكلمة | Al Kalimah | Word |
| الكناية | Al Kenayeh | Allusive |
| المبتدأ | Mubtada | Primate |
| المجرور | Majroor | With reduced ending [Genitive, |
| المضاف اليه | MudafElaih | Annexed |
| المعرّف | Moarraf | Determinee |
| المعطوف | Matoof | Attracted [Coupled] |
| المفعول | Al Mafool | Objectal |
| المفعول المطلق | MafoolMutlaq | Object |
| المفعول به | Al Mafool Behe | Object |
| المفعول لاجله | Al Mafool Le ajleh | WhyObject |
| المفعول له | Al Mafool Laho | ToObject |
| المفعول منه | Mafool Menh | FromObject |
| المكان | Al Makan | Location |
| الموصول | Al Mawsool | Conjunctive |
| امثلة المبالغة | Amthilat AL Mubalagah | Superlative |
| امر | Amr | Imperative |
| تام | Tamm | Complete |
| تحضيض | Tahdeed | Stimulation |
| تحقيق | Tahqiq | Authenticity |
| تخيير | Takhyeer | Selection |
| ترج | Tarajjy | Solicitation |
| تشبيه | Tashbeeh | Similitude |
| تصريف | Tasreef | Variability |
| تعجب | Taajjub | Astonishment |

| | | |
|---|---|---|
| تعريف | Taareef | Definition |
| تعليل | Taleel | Causality |
| تفسير | Tafseer | Interpretation |
| تفصيل | Tafseel | Separation |
| تقليل | Taqleel | Paucity |
| تكثير | Taktheer | Profusion |
| تمن | Tamanny | Wish |
| تنبيه | Tanbeeh | Premonition |
| تندم | Tandeem | Regret |
| توكيد | Tawkeed | Confirmation |
| جنس | Gens | Genus |
| جواب | Jawab | Answer |
| حال المفعول | Haal Al Mafool | HowObject |
| حرف | Harf | Particle |
| ردع | Rada | Rejection |
| زيادة | Zeyadeh | Augmentation |
| شرط | Shart | Condition |
| ضمير | Dameer | Pronoun |
| ظرف الزمان | Zarf Al Zaman | TemporalObject |
| ظرف المكان | Zarf Al Makan | LocationalObject |
| ظرفية | Zarfeyyah | Circumstance |
| عرض | Ard | Exposition |
| عطف | Atf | Conjunction |
| علم | Alam | Name |
| غاية | Gayah | Finality |
| فعل | Fe el | Verb |

| | | |
|---|---|---|
| قسم | Qasam | Oath |
| لازم | Lazim | Intransitive |
| متعد | Mutaady | Transitive |
| مجهول | Majhool | Unknown |
| مصدر | Masdar | Source |
| مصدرية | Masdareyyah | Originality |
| معلوم | Maloom | Known |
| مفاجأة | Mufajaa | Surprise |
| ناقص | Naqis | Incomplete |
| نداء | Neda | Call |
| ندبة | Nedbeh | Lamentation |
| نفي | Nafy | Negation |
| نهي | Nahy | Interdiction |

## Abstract

The most challenging part of natural language understanding is the representation of meaning. The current representation techniques are not sufficient to resolve the ambiguities, especially when the meaning is to be used for interrogation at a later stage. Arabic language represents a challenging field for Natural Language Processing (NLP) because of its rich eloquence and free word order, but at the same time it is a good platform to capture understanding because of its rich computational, morphological and grammar rules.

Among different representation techniques, Lexical Functional Grammar (LFG) theory is found to be best suited for this task because of its structural approach. LFG lays down a computational approach towards NLP, especially the constituent and the functional structures, and models the completeness of relationships among the contents of each structure internally, as well as among the structures externally. The introduction of Artificial Intelligence (AI) techniques, such as knowledge representation and inferencing, enhances the capture of meaning by utilising domain specific common sense knowledge embedded in the model of domain of discourse and the linguistic rules that have been captured from the Arabic language grammar.

This work has achieved the following results:

(i) It is the first attempt to apply the LFG formalism on a full Arabic declarative text that consists of more than one paragraph.

(ii) It extends the semantic structure of the LFG theory by incorporating a representation based on the thematic-role frames theory.

(iii) It extends to the LFG theory to represent domain specific common sense knowledge.

(iv) It automates the production process of the functional and semantic structures.

(v) It automates the production process of domain specific common sense knowledge structure, which enhances the understanding ability of the system and resolves most ambiguities in subsequent question-answer sessions.

## Statement of Novelty

The novelty of this work is that it extends the framework of LFG theory to include the semantic and pragmatic structures representation to the framework of the Lexical-Functional Grammar theory, which was designed to represent the syntax through the constituent and functional structures.

Moreover, the full framework has been implemented successfully in a prototype system on a complete story of 29 sentences written in Arabic language.

# Chapter One

## Introduction

The study of Natural Language Processing (NLP) has been a major research topic over the last three decades. A number of techniques and approaches have been proposed in order to resolve the enormous complexities of natural language processing. By far the most important techniques have been the proposal of different grammar theories, each of which claims flexibility and richness in handling both structure and semantics of natural language.

## 1.1 Objectives

The main objectives of this research are to: (i) Evaluate the most popular grammar theories in order to find out the most suitable one for Arabic sentences, (ii) Adopt the most suitable theory to represent the structures and semantics of Arabic sentences and perform any necessary enhancements by utilising the rich computational morphological and grammar rules, (iii) Develop a prototype system that implements the adopted theory along with the enhancements (iv) Apply the prototype on a few natural Arabic text stories and store both the original and the deduced information so it can be used for future utilisation such as query answering.

## 1.2 Motivations

English natural language processing has received a lot of attention from researchers and funding agencies. Arabic language on the other hand has not received the proper attention that matches its importance. Arabic language is the official language of twenty-one countries, and spoken by more than 252 million of people [Alai-96]. The use of computers in the Arab world is increasing very rapidly, with a resulting demand for more Arabic software. In addition, there is a need for many applications such as building a sophisticated intelligent system for modern studies of Arabic heritage, e.g., make important books available in a special format to extract answers to possible queries. The structure of the Arabic language represents a challenging field for Artificial Intelligence (AI) and Database (DB) researchers. The adoption of AI techniques to help in understanding the Natural Arabic Language (NAL) would be a significant achievement.

## 1.3 Natural Language Processing

### 1.3.1 What is Natural Language?

A language is called "Natural" when it is commonly used by human beings for the purpose of communicating between themselves (French, English, Arabic, etc.); natural languages are distinguished from "formal languages" (such as musical, mathematical notations, or programming languages) which have normally been created by some explicit and systematic act of definition for the purpose of being used in specific domains [Thay-89].

Linguists have mostly considered languages as a phenomenon whose rules and internal mechanisms must be explained, while Artificial Intelligence mainly sees

natural languages as a communication tool between a human being and a computer [Thay-91].

Natural Language Processing can be used for applications such as Machine Translation, Database modeling, Query Answering Systems and Natural Language Text regeneration. These applications should simulate human understanding of the natural language and produce the output that reflects the human response based on his/her understanding. This is exactly what Gazdar [Gazd-90] has described by saying that "in order to understand the meaning of a sentence the intended response from the statement has to be generated. "

### 1.3.2 Phases of Natural Language Processing

Natural Language processing has four main phases: morphological analysis, syntax analysis, semantic analysis and the pragmatic analysis [Nara-94] [Fedd-93] [Covi-94].

***Morphological analysis***: This is the analysis of the word regardless of its position in the sentence [Nara-94]. Morphological rules are used to generate new words from the linguistic *Roots* or stems through the insertion of *Affixes*. These rules can similarly be used to analyse the derived words.

***Syntax analysis***: This is concerned with the relationships between linguistic expressions [Fedd-93]. Sometimes this phase is referred to as, or is included in, parsing or grammar analysis. The approaches to syntax analysis include phrase structure grammar, transformational grammar, case grammar, augmented transition networks, conceptual parsing, systemic grammar and semantic grammar [Alaa-94].

***Semantic analysis***: This is concerned with the relationships between expressions and the object to which they refer. This phase checks for semantic validity of the syntactic

structures, builds semantic relations and represents them in a scheme. Semantics, or meaning, is the level at which language makes first contact with the real world. For a long time it was unclear how to describe the meanings of natural-language utterances. Mathematical logic and set theory [Alaa-94] have now provided suitable tools.

***Pragmatic analysis***: This is the use of language in context. The boundary between semantics and pragmatics is not clearly defined, different authors use the terms somewhat differently. In general, pragmatics includes aspects of communication that go beyond the literal truth conditions of each sentence [Covi-94].

Ideal Natural Arabic Language Understanding systems must support morphological analysis, syntax analysis, semantic analysis, and pragmatic analysis. Morphological analysis includes vowelization, vocabulary coverage, Arabic morphological rules, Arabic computational lexicons, and analysis and generation of Arabic words. Syntax analysis includes: Arabic grammar rules coverage; all sentence types (nominal, verbal, or interrogative); compound sentence structure. Semantic analysis should provide non-ambiguity, completeness, and correctness, while Pragmatic analysis should provide inference.

### 1.3.3 Grammars for Natural Language

Grammars are mathematical systems which (i) are used to define a language (ii) serve as devices for giving the sentences in the language a useful structure [Alfr-72].

There are three factors that can be used to evaluate grammars: *Generality* which is the range of sentences the grammar analyses correctly; *Selectivity* which is the range of non-sentences that it can identify as problematic; and *Understandability* which is the

simplicity of the grammar itself [Jame-87]. Grammars vary in achieving these three factors and this explains why some grammars have been more successful than others.

### *1.3.3.1 Context Free Grammars (CFG)*

A natural sentence consists of a hierarchy of phrases establishing the Constituent Structure. The Sentence (S) can have a Noun Phrase (NP) and a Verb Phrase (VP). The NP can have a Determinant (Det) and a Noun (N). The VP can have a Verb (V), NP, and a Preposition Phrase (PP). The PP can have a Preposition and a NP [Covi-94]. CFG is understandable, as it is simple and can show in a number of rules the structure of phrases, sentences and paragraphs. CFG satisfies the generality as all the correct structures can be represented. It is also satisfying selectivity as all non-described structures are considered to be incorrect. As the grammar of a language is expressed in an extensive list of CFG rules, the visualisation of the language structure is a bit difficult.

### *1.3.3.2 Simple Transition Network (STN)*

This network is composed of *Nodes* and *Labeled Arcs* [Jame-87]. Figure 1 shows a STN using the Context Free Grammar (CFG) symbols such as *art* for article, *NP* for noun phrase, *adj* for adjective, etc. It starts with the network name (e.g., NP:), followed by a node (e.g., NP), followed by a labeled arc (e.g. art), and so on, and should end with the arc labeled with the termination label "pop". This grammar is limited in its generality to represent the simple phrases of a sentence. It is understandable as it is visualising the grammar and can show which sentence structure is correct and which is not. The correct structure is when a matching grammar rule can be derived from the network such as an existing path from the first node to the ending arc labeled with pop. For example the rules NP ← art NP1, NP1 ← adj NP1,

NP1 ← noun NP2 are valid, any other rules are invalid. The reversed arrow (←) shows that the left hand side of the rule consists of the right hand side symbols as in the CFG rules.

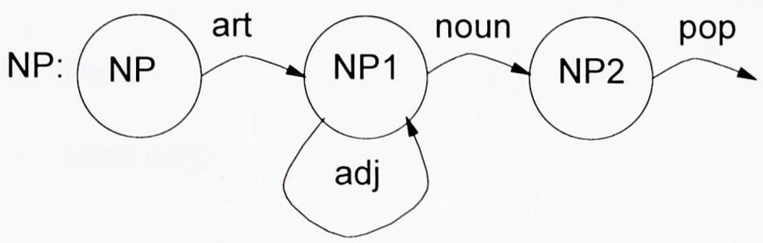


**Figure 1: Simple Transition Network**

### *1.3.3.3 Recursive Transition Network (RTN)*

The STN cannot represent the recursion of the natural language that is the compound sentence of a number of sentences. The Recursive Transition Network is introduced to overcome this problem by allowing the arc labels to have names in capital letters to refer to other networks (e.g., NP) along with word categories remaining in small letters (e.g., verb) [Jame-87], see figure 2. RTN is not as understandable as the STN because RTN is not as visual when arc labels refer to other networks.

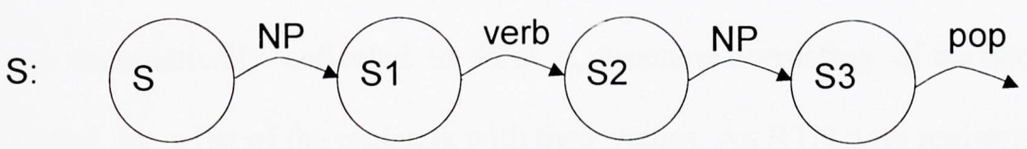


**Figure 2: Recursive Transition Network**

### *1.3.3.4 Augmented Transition Network (ATN)*

The previous grammars do not show the functional features of the language such as Subject, Object, etc. Adding such features to the RTN makes the Augmented Transition Network.

For the Sentence

*Ali found a cat*

the following ATN is produced

(S Subject (NP Name *Ali*)

Main-Verb *found*

Tense Past

Object (NP Det *a*

Head *cat*))

The RTN parser creates such a structure by allowing each network to have a set of *registers*. Registers are local to each network. Thus each time a new network is pushed, a new set of empty registers is created. When the network is popped, the registers disappear. In this case, the registers will have the names of the slots used for each of the preceding syntactic structures. Thus the NP network has registers named Det, Adjs, Head, and Num. Registers are set by *actions* that can be specified on the arcs. When an arc is followed, the actions associated with it are executed. The most common actions involve setting a register to a certain value. Other actions will be introduced as necessary. When a pop arc is followed, all registers set in the current network are automatically collected to form a structure consisting of the network name followed by a list of the registers with their values. An RTN with registers that are subject to tests and actions, is an augmented transition network [Jame-87].

### 1.3.4 Parsing the Natural Language

Parsing is the process of computing the structures assigned to a given phrase by a given grammar. As a declarative description of a language, a grammar does not specify how syntactic analysis are to be computed and there is a vast area of possible parsing algorithms [Gazd-90].

### *1.3.4.1 Top-Down Methods*

This parsing method starts from the grammar representation of a sentence and decomposes this representation into its sub constituents. Then it further decomposes the sub constituents until a specific word class is derived that can be checked against the actual input sentence. In this method we use the grammar rules (e.g., NP) to find the matching sequence (e.g., Art Noun, Adj Name, etc) [Jame-87].

### *1.3.4.2 Bottom-Up Methods*

This method is the process of matching a found sequence (e.g., Art Noun) to the right hand side of the grammar rules (e.g., NP ←Art Noun, VP ←Verb NP) in order to identify the correct rule, which is NP in this case. Matches are always considered from the point of view of one symbol, called the *key*. To find rules that match a string involving the key, look for rules that start with the key, or for rules that have already been started by earlier keys and require the present key either to complete the rule or to extend the rule [Jame-87].

### *1.3.4.3 Mixed-Mode Methods*

This method is the best parsing strategy as it combines the advantages of both Top-Down and Bottom-Up methods and avoids their disadvantages. The parser that uses this strategy is called the Chart Parser. The Top-Down method has the advantage that it will never consider word categories in positions where they could not occur in a legal sentence. This is because the parser works from a syntactic category and checks for the word that fits this syntactic category. Moreover, top-down parser may operate for quite some time, rewriting rules from complex grammar before the actual words in the sentence are ever considered. Even more important, the same piece of work may be repeated many times in searching for a solution.

The above problems are avoided with the bottom-up parser but, on the other hand, bottom-up parser must consider all categories of each word and construct a structure that could never lead to a legal sentence. For example the word "can" could be a verb to construct a VP or a noun to construct a NP. It is possible to design systems that use varying degrees of both top-down and bottom-up methods and gain the advantages of both approaches without the disadvantages. One such approach is to construct a top-down parser that adds each constituent to a chart as it is constructed. As the parse continues, before you rewrite a symbol to find a new constituent, you first check to see if that constituent is already on the chart. If so, you use it rather than applying the grammar to construct the constituent all over again [Jame-87].

There are a number of computational linguistics theories used as frameworks to represent natural languages and parse them using most of the techniques mentioned above such as the Transformational Grammar, Head-driven Phrase Structure Grammar, and the Lexical-Functional Grammar. These theories will be described in detail in chapter two. Moreover, those techniques are short of capturing the effect of actions between objects within the natural text. Hence the next section describes this concept which has been adopted in NLP to model the verbal interaction [Wins-92].

## 1.4 Thematic Role

Much of what happens in the world involves actions, and objects undergoing change. It is natural, therefore, that many of the sentences in human language specify actions, identify the object undergoing change, and indicate which other objects are involved in the change [Wins-92].

In linguistic terms, verbs often specify actions. Each noun phrase's thematic role specifies how the object participates in the action. For example the sentence "Ali hit a dog with a stick " carries information about how Ali, a dog, and a stick relate to the verb "Hit"[Wins-92]. See figure 3.

| | | | | | |
|---|---|---|---|---|---|
| Verb | Hit | Source | | Time | |
| Agent | Ali | Destination | | Location | |
| Co-agent | | Old surroundings | | Duration | |
| Beneficiary | | new surroundings | | | |
| Thematic Object | a dog | Conveyance | | | |
| Instrument | a stick | Trajectory | | | |

**Figure 3: Thematic role frame**

## 1.5 The Representation of lexical knowledge

The lexicon has permitted computational linguistics to adopt very simple and compact grammatical rule systems at the cost of pushing almost all of the syntactic facts about the language into the lexicon. This makes the organization of such a lexicon a very critical task. In a natural language understanding system, the lexicon would contain information such as: part of speech, sub categorization possibilities, case, finiteness, number, person, gender on noun class, aspect, mood, reflexiveness and WH-ness. The lexicon should also list word roots, sufficient morphological and syntactic information for the regular forms of words to be deduced. This information makes the lexicon a very important input to the parser, which also requires some semantic information to be included in the lexicon. [Gazd-90]

## 1.6 Arabic Language Structure

Arabic language is composed of words constructed from roots and affixes. Different combinations of these words form the sentence. The structure of the meaningful sentence should conform to the authenticated Arabic grammar. The rest of this section discusses the above terminology in detail. The discussion is derived mainly from [Alja-88], [Abus-85], [AlHa-80] and [Anto-94].

### 1.6.1 The Word

The word is any combination of letters that give a useful meaning. It could consist of one letter (ق care), two (كل every), three (شرب Drank) or more. The word is mainly of three types, *Noun* اسم, *Verb* فعل and *Particle* حرف. See figure 4.

The Noun independently means something but does not point to any tense. The noun is decomposed into a number of linguistic categories such as a human *Name* اسم علم e.g. (ALI علـــي). Any instance of Noun should belong to one linguistic category. The Verb independently means something and points to a tense. e.g. (past مـــاض,present حـــاضر,imperative امـــر). A past verb like (drank شـــرب), present like (drink يشرب) or imperative like (drink اشرب). Verbs are decomposed into *Complete* and *Incomplete*, the incomplete verb is further decomposed into *Transitive* and *Intransitive*, and the transitive verb is further decomposed into *Known* and *Unknown*. The complete verbs have at least one *Subject*. The intransitive verbs have at least one *Object*, while the Transitive verbs can have up to three Objects. The known form of a verb requires a Subject to exist in the sentence, while the Unknown form indicates that the Subject is omitted, but the Objects still exist. The Particle means something only in the company

of a Noun or a Verb. Particles are in the Arabic language to serve certain semantic purposes (e.g. confirmation التوكيد group that consists of many particles (e.g. Inna ان)).

### 1.6.2 The Affixes

Certain letters are used to change the meaning or the state of a word when they are added to the beginning, middle or to the end of that word. For example the letter (Y ي) is considered to be a *Prefix* when it is added to the beginning of the past tense verb (Drank شـــرب) converts it to the "present tense" verb (Drinks يشرب). The *Infix* is a letter like (A ا) when inserted after the first letter of the "past tense" verb (Drank شرب) converts it to the *Subjectal Noun* (Drinker شارب). The *Suffix* is added to the end of a word. For example, the letters (Woon ون) when it joins the singular verb (He Drinks يشـــرب) at the end converting it to the plural form (They (male) Drink يشربون). Another example is (N ن) for the female which gives (They (female) Drink يشربن) indicating that this verb is being performed by females.

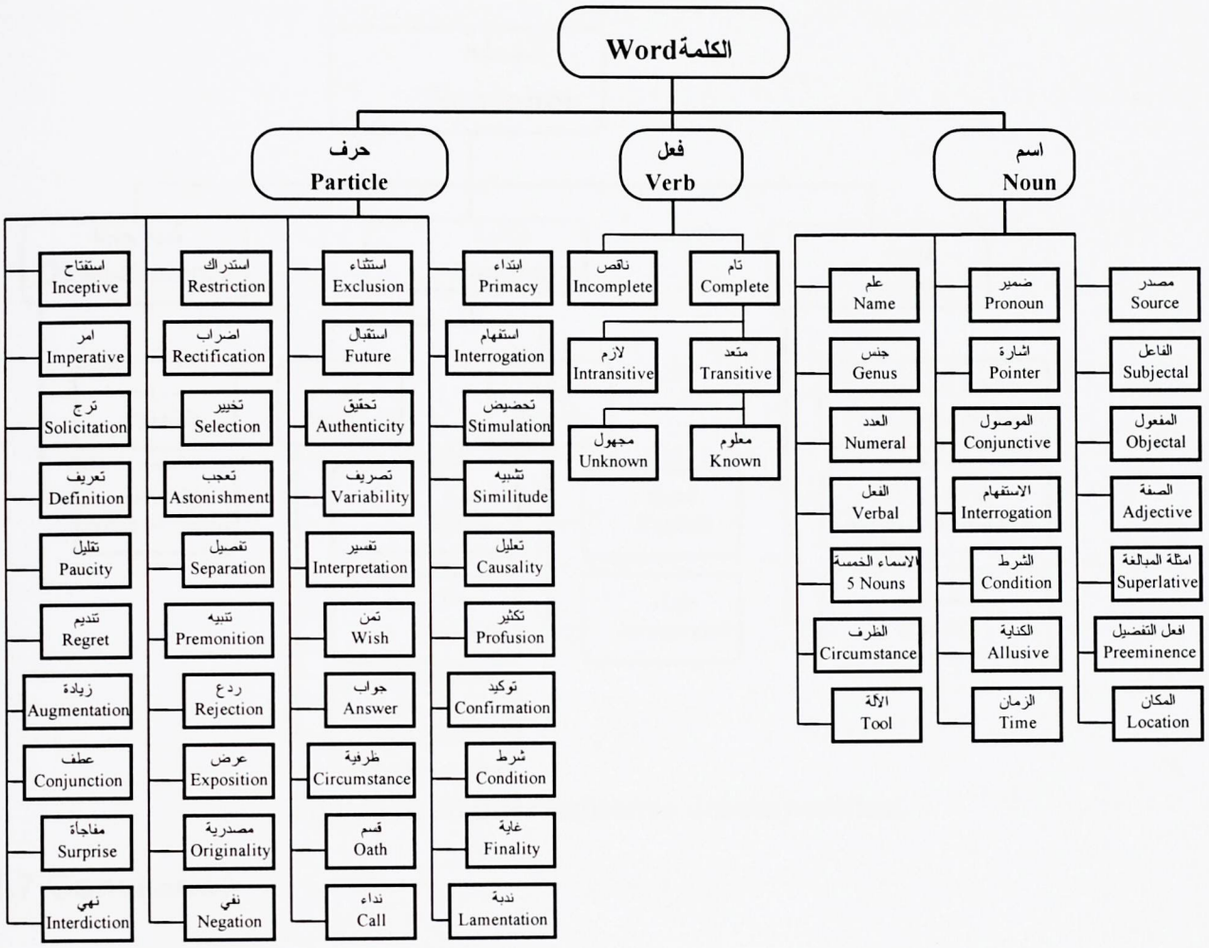


**Figure 4: Arabic Word decomposition**

### 1.6.3 The Sentence

One or more words could form a sentence or semi-sentence. The sentence that conveys a meaning and satisfies a linguistic syntax is called a meaningful sentence. The meaningful sentence is of two types, *Verbal* that starts with a verb such as (Ali drank the water شرب علي الماء) or *Nominal* that starts with a *Primate* مبتدأ and is completed by a *Predicate* خبر as in (Ali is brave علي شجاع). The Quasi-sentence is that which starts with a Preposition (e.g. "In the bag في الشنطة" or a circumstance e.g., " Over the table فوق الطاولة". See figure 5.

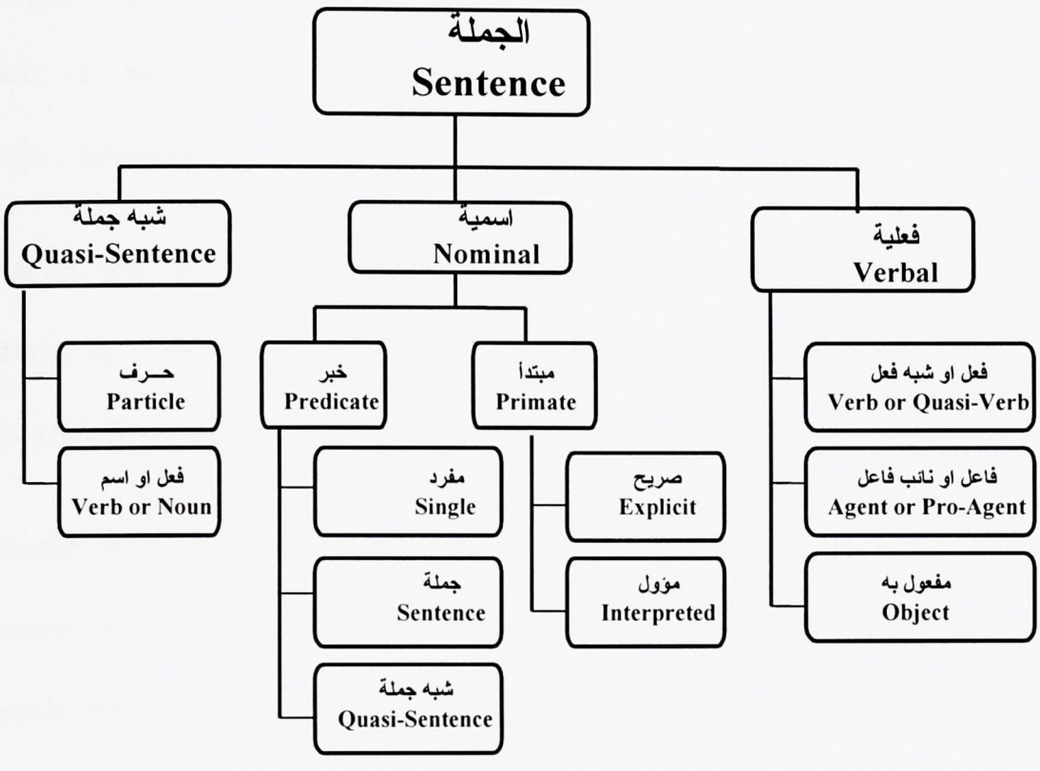


**Figure 5: Arabic Sentence decomposition.**

## 1.7 Conclusion

In order to have a comprehensive natural language understanding system, it should process three related phases: Syntax, Semantics, and Pragmatics.

The most challenging part of natural language understanding is the representation of meaning. The current representation techniques are not sufficient to resolve the ambiguities, especially when the meaning is to be interrogated at a later stage.

Arabic language represents a challenging field for Natural Language Processing (NLP) because of its rich eloquence and free word order, but at the same time it is a good platform to capture understanding because of its rich computational morphological and grammar rules.

## 1.8 Summary of Chapters

Chapter one gives an introduction to the field of natural language processing and states the objectives of this specific research topic, and describes the structure of the Arabic language.

Chapter two reviews the literature on this subject and starts with the achievements pertaining to Natural Language Processing in general mainly to English. This chapter also includes a survey of progress in Natural Language Processing for Arabic.

Chapter three describes the data and knowledge representation in addition to the process model. The chapter also describes the input to the system, which is a parsed constituent structure that is used to generate domain specific common sense knowledge structure (k-structure) through the functional structure (f- structure ) and the semantic structure (s- structure) with the assistance of the Lexicon.

Chapter four describes the basic architecture of the proposed prototype in terms of input/output diagrams, production rules, algorithms and inference trees. This chapter also describes the architecture of the Lexicon in terms of categorised words within Nouns, Adjectives, Verbs and Particles tables.

Chapter five describes the implementation platform and the results of implementing the various structures. This chapter also describes the implemented object's database with three class hierarchies. The first accommodates the C and F structures together, the second accommodates both the S and K structures and the third is for the Lexicon.

Chapter six describes the testing and evaluation of the input and output of each module. Some statistics are presented in this chapter for each structure including the

lexicon. Such statistics reflect the percentage of success for each structure compared to the design objective.

Chapter seven presents the conclusions reached from the research and implementation activities. A number of industrial applications are mentioned. This chapter also describes future work that could arise from this research and the design and implementation requirements.

# Chapter Two

# Literature Review

## 2.1 Introduction

This chapter reviews the popular computational linguistics frameworks. The review of such frameworks is described in terms of their structure, contribution, limitations, and some natural language processing systems based on them. Natural language processing systems that are not based on a theoretical computational linguistic framework such as [Robe-98] [John-76] are excluded from this survey.

This chapter also reviews progress in natural Arabic language processing research and the attempts towards developing natural Arabic language understanding systems. In this part of the review, the system will be described in terms of the computational linguistic framework it uses, the modules it processes and its limitations.

Finally this chapter identifies the research scope and the approach adopted in producing the intended results.

## 2.2 Computational Linguistics Frameworks for Natural Language

When undertaking the development of a Natural Language understanding system, it is advisable that this system should be based on a solid theoretical framework. A

number of popular natural language representation grammars have been reviewed. Their structures, contributions, and limitations are described. Computerised applications are described where available.

### 2.2.1 Chomsky's Transformational Grammar (TG)

Chomsky's Transformational Grammar is a theory of how the components of linguistic competence work together [Step-98]. TG has Transformational rules for transforming a sentence into a closely related sentence. For example the sentence "The boy hit the ball (NP1 + Verb + NP2)" becomes "The ball was hit by the boy (NP2 + was + Verb + by + NP1)" [Noam-98].

**TG Structure**

TG consists of two structures, the Deep Structure and the Surface Structure. The Deep Structure is the structure of the sentence resulting from the application of the phrase structure rules. It conveys the meaning of the sentence, but may be ungrammatical. The Surface Structure is the final description of the sentence after application of the transformational rules to the deep structure [Step-98]. TG rules define the way in which deep and surface structures are related. Transformations turn one tree into another by adding, deleting or moving constituents. An example for applying the relativization transformation is given below:

Ali is a good boy.
Ali does not go to school.
➔ is transformed into the following surface structure:
Ali who does not go to school is a good boy.

**TG Contribution**

TG provides an explanation for the syntactic system, semantic system, and the phonological system. These linguistic universals were thought to derive from an

embedded mechanism that provides humans with the structures needed to acquire and use natural languages. TG also demonstrates the inadequacies of the behaviourist attempt to explain human language [Step-98].

**TG Limitations**

Meaning and surface structure are only indirectly connected [Step-98]. This is described in three levels of ambiguities: the lexical ambiguity, surface structure ambiguity and the deep structure ambiguity. The lexical ambiguity is the cause of the surface structure and deep structure ambiguities as in the word *Fly* which can be a verb lexical entry or a noun. This in effect generates two surface structures and two deep structures, resulting in the surface structure ambiguity and the deep structure ambiguity.

**TG Systems**

Friedman [Frie-69] described a comprehensive system for transformational grammar, which has been designed and implemented on an IBM 360/67 platform. The system deals with the transformational model of syntax, along the lines of Chomsky's Aspects of the Theory of Syntax. The major innovations include a full, (i) formal description of the syntax of a transformational grammar, (ii) a directed random phrase structure generator, (iii) a lexical insertion algorithm, (iv) an extended definition of analysis, (v) and a simple problem-oriented programming language in which the algorithm for the application of transformations can be expressed.

### 2.2.2 Head-driven Phrase Structure Grammar (HPSG)

HPSG is a linguistic theory based on signs that are structured phonology, syntax, semantic, discourse and other phrase structural information. Signs include sentences, clauses, phrases and lexical items [Carl-94].

**HPSG Structure**

The signs in HPSG have the phonological information features PHON and the syntax/semantics information features SYNSEM. The SYNSEM features are defined in terms of information about the long distance dependencies NONLOCAL and other syntactic and semantic information in LOCAL. LOCAL includes CATEGORY for categorical and sub categorization information, and CONTENT whose value contains semantic information [Davis -96]. Figure 6 shows the HPSG abstract frame.

$$\begin{bmatrix} \textit{sign} & \begin{bmatrix} \text{PHON} & \textit{phonology} \\ \text{SYNSEM} & \textit{synsem} \begin{bmatrix} \text{LOCAL} & \textit{local} \begin{bmatrix} \text{CATEGORY} & \textit{category} \\ \text{CONTENT} & \textit{content} \end{bmatrix} \\ \text{NONLOCAL} & \textit{nonlocal} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

**Figure 6: HPSG abstract frame**

In HPSG, the constituent structure is represented by the various child attributes of phrasal signs, and trees are used as a convenient graphic representation of the immediate constituents and linear order properties of phrasal signs [Green-98]. Figure 7 shows the HPSG Flow of Linguistic Information for the sentence: Kim likes Pat.

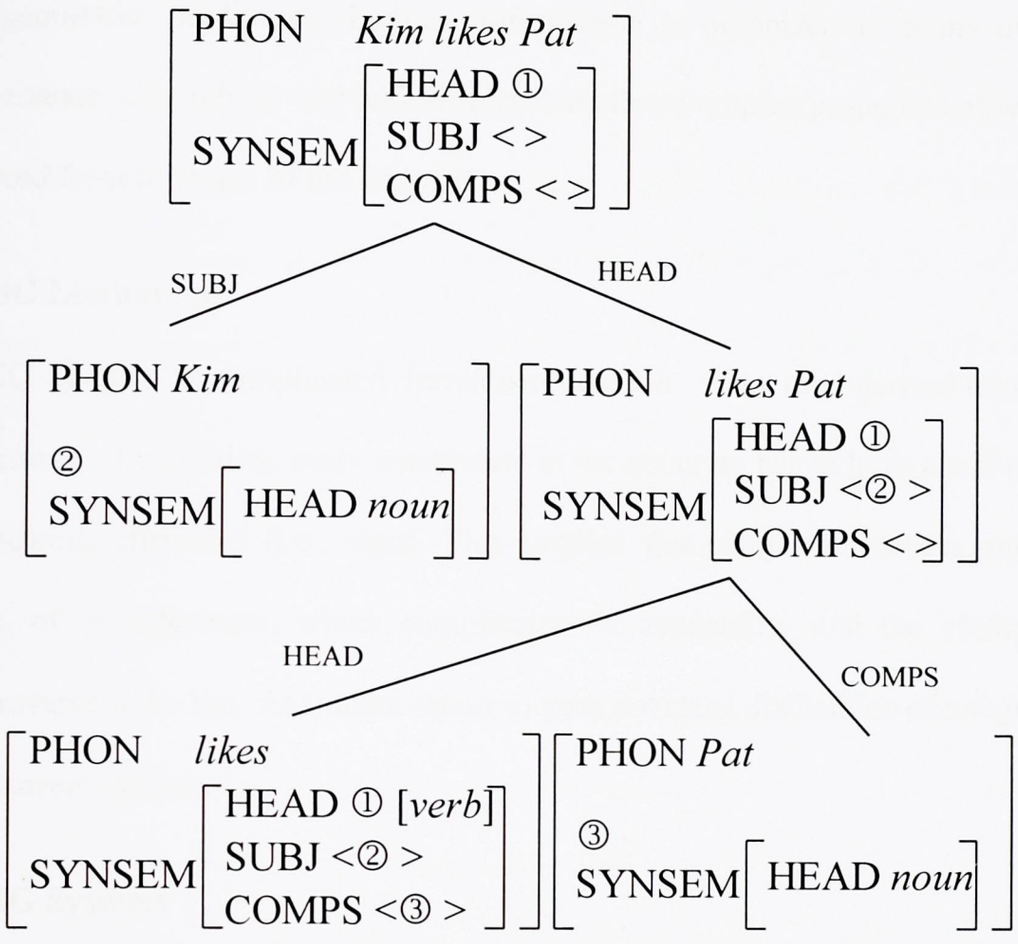


**Figure 7: HPSG Flow of Linguistic Information**

**HPSG Contribution**

Susanne [Susa-96] describes some leading ideas on HPSG. Strict Lexical word and phrase structures are defined such that they are governed by independent principles. Concrete, surface-oriented structures are maintained such as empty categories. Functional projections are avoided wherever possible, using relatively conservative constituent structures instead. The Geometric prediction is achieved through hierarchically organizing Linguistic information in such a way as to predict the impossibility of certain kinds of linguistic phenomena. Locality of head selection is an idea that is implemented through the selection of lexical heads only for the SYNSEM objects of their complements, subjects, or specifiers. It follows that category selection, role assignment, case assignment, head agreement and semantic selection all obey a particular kind of locality determined by equivalence selection features, this is a kind

of geometric prediction. Lexical information is organized in terms of multiple inheritance hierarchies and lexical rules that allow complex properties of words to be derived from the logic of the lexicon.

**HPSG Limitations**

HPSG is quite a complicated formalism since it is not modularised into linguistic structures. In addition, every constituent in the structure has to have a complete set of framework attributes (i.e., sign). This implies that many slots remain empty for the sake of completeness, which complicates the readability and the clarity. Further improvement to the formalism seems to pose potential difficulties of integration into the current system.

**HPSG Systems**

Many systems have been developed to implement the HPSG formalism [Bolc-96], among which is the Type Description Language (TDL) [Jong-98]. TDL is a typed feature-based representation language and inference system, specifically designed to support highly lexicalized grammar theories like HPSG. Type definitions in TDL consist of type and feature constraints over the Boolean connectivity. TDL supports open-world and closed-world reasoning over types and allows for partitions and incompatible types. Working with partially as well as with fully expanded types is possible. Control knowledge is specified on a separate layer. Efficient reasoning in TDL is accomplished through several specialized modules. TDL is part of a larger system that provides further components: a parser, an explanation-based learning component, morphology (2-level + classification-based), feature editor, type grapher, chart display, a large German HPSG grammar (approx. 1500 type definitions), etc.

### 2.2.3 Lexical-Functional Grammar (LFG)

Lexical Functional Grammar (LFG) is a theory that was first introduced by Kaplan and Bresnan in 1982 [Kapl-82]. The LFG formalism has evolved from computational, linguistic, and psychological research, which provides a simple set of devices for describing the common properties of all human languages and the particular properties of individual languages [Kapl-89].

**LFG Structure**

In Lexical-Functional Grammar (LFG) [Kapl-89], there are two parallel levels of syntactic representation: constituent structure (c-structure) and functional structure (f-structure). C-structures have the form of context-free phrase structure trees. F-structures are sets of pairs of attributes and values; attributes may be features, such as tense and gender, or functions, such as subject and object. The name of the theory emphasizes an important difference between LFG and the Chomskyan tradition from which it is developed. Many phenomena are thought to be more naturally analysed in terms of grammatical functions as represented in the lexicon or in the f-structure, rather than on the level of a phrase structure. An example is the alternation between active and passive, which rather than being treated as a transformation, is handled by the lexicon. Grammatical functions are not derived from phrase structure configurations, but are represented at the parallel level of functional structure. Figure 8 shows a matching between the c-structure and the f-structure in LFG for the sentence: Seeing me surprised Mary.

**LFG Contribution**

LFG places great importance on the words in the lexicon so that much of the work of syntactic description is done by an elaborated theory of the lexicon. The natural

language is completely described by LFG in a modular way by considering a grammatical system that make use of multiple parallel levels of linguistic representation (surface phrase structure, grammatical relations, argument structure, semantics and information structure) with corresponding relations between levels. Moreover, many grammatical processes are completely described by LFG in terms of grammatical functions or in terms of the primitives of other levels rather than in terms of phrase structure configurations [Ling-98].

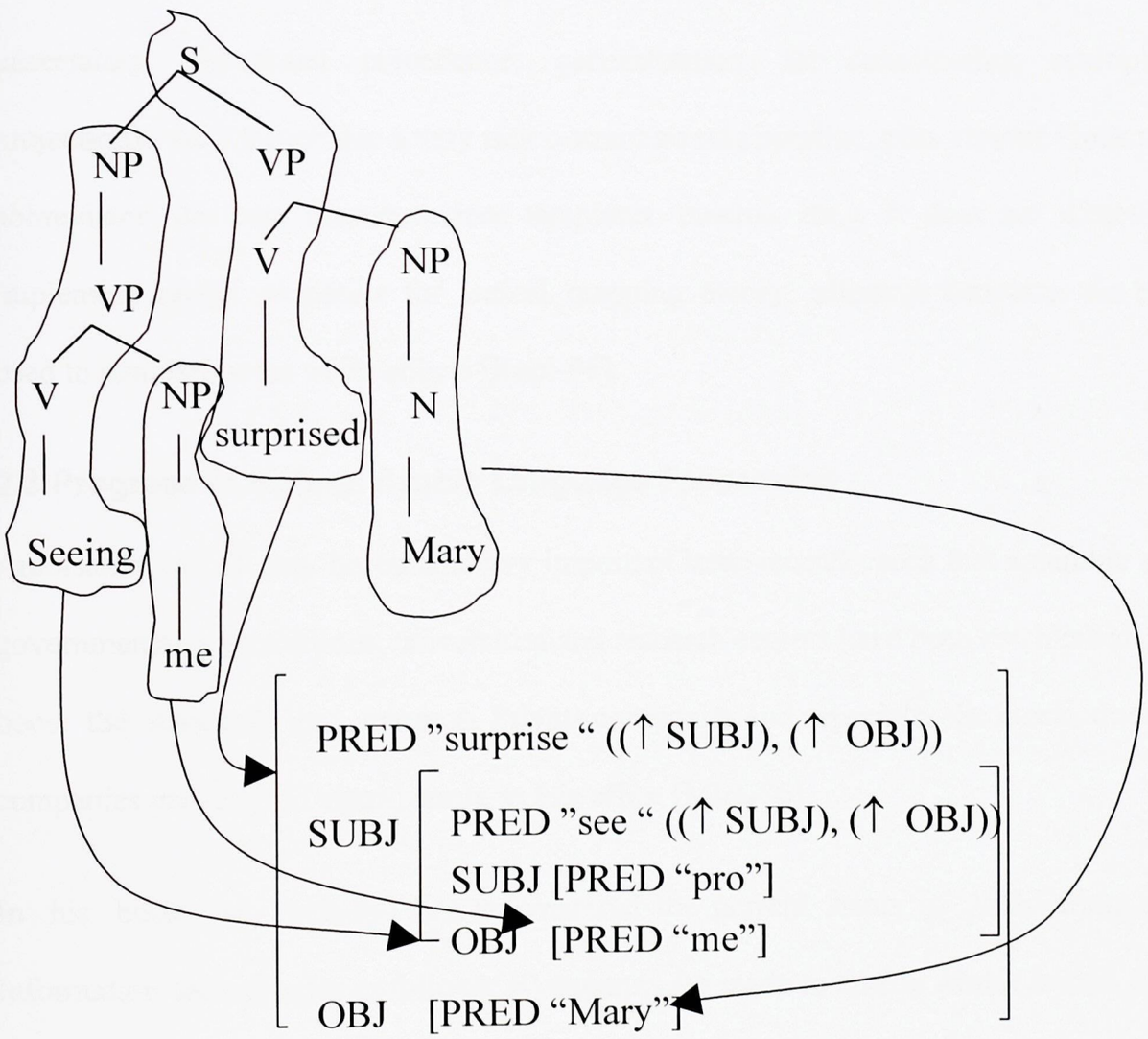


**Figure 8: Matching between the c-structure and the f-structure in LFG**

**LFG Limitations**

The LFG framework is designed chiefly to process the syntactic level of the natural language. The semantic structure is described in abstract form as a set of predicate-

arguments. For a complete natural language understanding system at the syntactic and semantic levels, semantic and pragmatic structures are required.

**LFG Systems**

Among the implemented systems, Xerox LFG Grammar Writer's Workbench is the most important as it is being developed by those who introduced LFG. It is a complete parsing implementation of the LFG syntactic formalism, including various features introduced since the original Kaplan and Bresnan [Kapl-82] paper (functional uncertainty, functional precedence, generalization for coordination, multiple projections, etc.) It includes a very rich c-structure rule notation, plus various kinds of abbreviator devices (parameterized templates, macros, etc.). It does not directly implement recent proposals for lexical mapping theory, although templates can be used to simulate some of its effects [Kapl-96].

## 2.3 Progress in Natural Arabic Language Processing

Arabisation of IT has become a very important issue recently such that a number of governmental organisations, universities and research centers have been established to boost the standards and research. Private companies and especially the international companies participated significantly in this effort [Mira-96].

In his book Ali [Ali-94] has summarized the current status of Arabisation of Information technologies as follows; (i) attempts are made to enable Arabic within the English application; (ii) interfaces are developed for Arabizing the Data Entry; (iii) progress is limited due to the Arabization process being undertaken by non-Arabs; and (iv) the absence of the essential research in Arabic computational linguistics is a serious limitation.

### 2.3.1 Computer Based System for Understanding Arabic Language (CBSUAL)

A Computer Based System for Understanding Arabic Language [Ghei-89] was developed to understand Arabic text written in the form of exercises in Mechanics for school students through translating it into French and then solving it.

**CBSUAL Framework**

In [Ghei-89] the author mentioned that the augmented transition network ATN was used for the morphological analyser. The semantic network together with a set of rules were used to describe the transformation of a sentence into its internal representation.

**CBSUAL Modules**

*Dictionary*: it mainly contains, a) All used vocabulary divided into several classes according to their semantic value and their grammatical categories excluding the inflections; b) Translation of words into French; c) Conditions to be fulfilled and/or the actions to be executed such as Add, Delete, or Replace.

*Lexical Analyzer*: this is a program which performs three routines: a) Accepts input in normal Arabic orthography and punctuation, looks up words in the dictionary and performs morphological analysis while recognizing the words; b) Reads the grammar network and builds up a data structure representing the ATN; c) Traverses the ATN so that it attempts the arc that is leaving a state in the order in which they are listed in the grammar.

*Semantic analysis* is the process of: a) Translating the main ideas of the Arabic Mechanics exercise in this case into French; b) Producing numerical results as a solution to the exercise.

**CBSUAL Limitations**

The input is not purely natural Arabic language text. It is specifically designed to describe mechanics exercises for High School students.

A weak computational linguistic framework is used. It lacks modularity, integration, and clarity.

### 2.3.2 Xerox Morphological Analyzer (XMA)

The Xerox Morphological Analyzer [Bees-98] is a finite-state morphological analyzer of written modern standard Arabic. The system consists of the analyzer, running on a network server, and Java applets that run on the user's machine and render words in standard Arabic orthography both for input and output.

**XMA Framework**

The Arabic morphological analyzer is built using Finite-State compilers and algorithms, and the results are stored and run as Finite-State Transducers (FST). FST is the corresponding machine that accepts all and only the ordered pairs in the Finite-State relation, and if given a string from the lower language, it returns all the related strings in the upper language, and vice versa. The Finite-State relation is thought of as having an upper-side language and a lower-side language; and each string in one language is related to one or more strings in the other language.

**XMA Modules**

*User Interface:* It is a Web Browser Java applet that runs on the user's machine and accepts input and displays output in standard Arabic orthography through an internal buffer.

*CGI Script (Perl)*: This script runs on the server and transfers the information between the user's machine and the other three modules.

*Morphological Analyzer*: It subjects each input word to an upward direction analysis. Typically there are several output strings, each representing a possible analysis of the input word into an upper-level language.

*Morphological Generator*: It takes all the possibilities produced by the morphological analyzer and applies a downward direction generation of the lower-level language that is restricted to fully-vowelized strings.

*English Glossary Buffer*: The various solutions are also tokenized into morphemes, which are looked up in an English glossary.

**XMA Limitations**

The system processes the word analysis and generation phases only. Sentence syntax and semantics phases are not covered.

Finite-state framework poses some complications in modeling the Infixes of the Arabic morphology.

The system needs to devise a way to handle multi-word expressions before the work expands into part-of-speech disambiguation and parsing.

### 2.3.3 Arabic-To-English Machine Translator (ATEMT)

Apptek is researching an Arabic-To-English Machine Translator [Shih-98] that accepts natural Arabic as the source language and translates it into English as the target language.

**ATEMT Framework**

The ATEMT is based on the Lexical-Functional framework, which consists of three structures: the constituent; the functional; the lexical. The constituent structure is the external representation of the language that consists of Noun Phrases, Verb Phrases, and Prepositional Phrases. The functional structure is the internal representation of the language that includes Subjects, Objects, and predicates. The lexical structure is the representation of the words of the language in terms of its attributes such as singular, plural, feminine, masculine, etc.

**ATEMT Modules**

*Parsing*: It is an active chart parsing process with bottom-up first. It is either left-to-right or right-to-left, and in breadth-first manner. The parser takes the natural Arabic sentence and produces the constituent structure and consequently produces the functional structure.

*Transfer*: This is the process of transforming the Arabic source functional structure into the English target functional structure. This involves selecting the most suitable English target word that corresponds to the given Arabic source word.

*Generation*: The process here is converting the English structure into the target natural English sentence directly without producing the constituent structure.

**ATEMT Limitations**

This system produces the constituent structure and functional structure only without producing the semantic structure, which could resolve the possible semantic ambiguities.

The constituent structure is not produced for the target English language, which could resolve the possible word order ambiguities.

The author mentioned that in some cases the system faces difficulties in matching nouns from the source language to adjectives in the target. For example, the source noun حق should be the adjective *rightful*. He also mentioned that the preposition phrase is not handled properly in some cases such as the source preposition من in the source sentence حقه من العناية it should be excluded during the transformation process to have the target sentence *his rightful attention* without the proposition *from*.

## 2.4 The Work of this Thesis

The scope of this research is to develop a system to simulate the natural Arabic language understanding. The system will deduce the meaning of a given text and have it available for future interrogations, machine translation, etc.

LFG formalism will be applied to a full Arabic declarative text. It will be extended to accommodate the semantic structure that would be designed according to the thematic role theory. LFG will also be extended to accommodate the sort of knowledge structure required to use domain specific common sense knowledge in refining the semantic structure.

The input of the proposed system is a constituent structure produced manually from Arabic natural language sentences. The output of the system is the f-structure, s-structure, and the k-structure.

The implementation of this system will be evaluated and future work will be suggested to expand the system functionality.

## 2.5 Conclusion

Present research and development in the area of natural Arabic language understanding does not go beyond the syntactic phase. Both the semantic phase and the pragmatic phase still need more investigation, which triggered this work.

A natural Arabic language understanding system based on the LFG formalism is proposed and will be developed. The system should pass through four processing phases: the c-structure, f-structure, s-structure, and the k-structure.

Lexical Functional Grammar (LFG) theory is found to be best suited for natural Arabic language understanding among the popular computational linguistics frameworks because of its structural approach, which is excellent for implementation and future expansion. LFG lays down a computational approach towards NLP, especially the constituent and the functional structures, and models the completeness of relationships among the contents of each structure internally as well as among the structures externally. LFG gives due consideration to the functional structure. This is good for Arabic language because the meaning in Arabic language is heavily dependent on the functional description of the sentence.

LFG still needs to be extended to accommodate the semantic structure and domain specific common sense knowledge structure and this is the focus of this work.

# Chapter Three

# The System Analysis

## 3.1 Introduction

The data and knowledge representation and the process model are described in this chapter. At present, the input to the system is a parsed constituent structure that is used to generate domain specific common sense knowledge structure (k-structure) through the functional structure (f-structure) and the semantic structure (s-structure) with the assistance of the Lexicon.

## 3.2 Data and Knowledge Representation

The adopted representation covers mainly the theoretical, linguistic and semantic structures.

### 3.2.1 The Theoretical Structure

The Lexical-Functional Grammar (LFG) theory is the selected framework to build the proposed prototype. The current constituent and functional structures of the LFG are used for this purpose. The semantic structure needs to be modified and an additional domain specific common sense knowledge structure has to be introduced.

**An Arabic Natural sentence:** عصف وقوع حادثين كبيرين بسكون شارع سكني هادئ في غضون ســـاعتين مـــن الاسبوع الماضي

**The transliteration:** asifa woqooa hadithain kabeerain be sokoon sharea sakani hadi fi ghodoon saatain min alosbooa almadhi

**The translation:** The occurrence of two big accidents have stormed a residential quiet road within two hours of the last week.

```
VP
"    |___ V----عصف"
"    |___ NP"
"    |    |___ N----وقوع"
"    |    |___NP"
"    |         |___N----حادثين"
"    |         |___Adj----كبيرين"
"    |___PP"
"    |    |___P----ب"
"    |    |___NP"
"    |         |___N----سكون"
"    |         |___NP"
"    |              |___N----شارع"
"    |              |___Adj----سكني"
"    |              |___ConjP"
"    |                   |___Conj----و"
"    |                   |___Adj----هادئ"
"    |___PP"
"         |___P----في"
"         |___NP"
"              |___N----غضون"
"              |___NP"
"                   |___N----ساعتين"
"                   |___PP"
"                        |___P----من"
"                        |___NP"
"                             |___Det----ال"
"                             |___N----اسبوع"
"                             |___AdjP"
"                                  |___Det----ال"
"                                  |___Adj----ماضي"
```

**Figure 9: Constituent structure**

The constituent structure is the first level of the framework, which is designed based on the context free grammar (CFG). The natural language sentence is broken into tokens based on the linguistic categories such as nouns, verbs, and articles. The resultant tokens are in turn grouped into phrases such as Noun Phrases (NP), Verb Phrases (VP), or Prepositional Phrases (PP). The logical collection of the CFG phrases constitutes the parsed sentence. Figure 9 shows an Arabic natural sentence, its transliteration, translation and constituent structure. Note that a conjunction phrase is required to represent a series of adjectives.

The constituent structure is a solved problem and available as on the shelf product, therefore it will not be automated in this work. Despite this fact, figure 9 shows that the Conjunction (و) is added to normalise the relationship between the noun (Road شارع) and its two objectives (Residencial سكني, Quiet هادئ)

The functional structure is the second level of the theoretical framework in which the functional role (e.g., Subject, Object, etc.) of each constituent should be identified in the sentence. Moreover, the functional relationships among all constituents have also to be identified. Figure 10 shows the theoretical functional structure of the above sentence.

The semantic structure is the third level of the theoretical framework in which the meaning of the text is represented as a frame hierarchy. Figure 11 shows the thematic role representation of the above sentence having four thematic roles, which are Actions, Themes, Timings, and Locations.

The domain specific common sense knowledge structure is the fourth level of the framework in which the semantic structure has to be refined such that objects/attributes/relationships are modified, inserted or deleted.

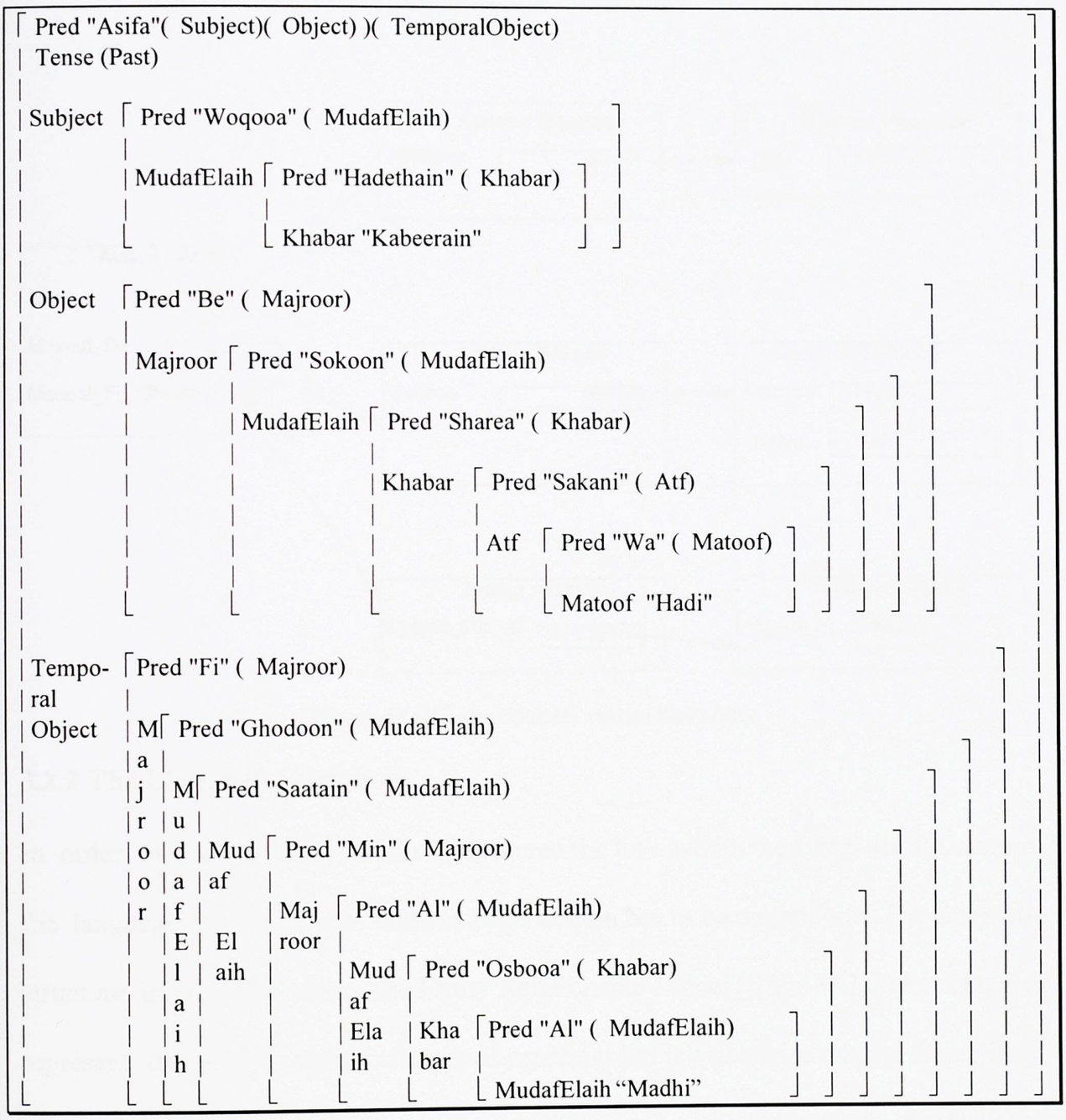


**Figure 10: Functional structure frames**

The lexicon is a collection of natural word entries categorized according to their linguistic properties. These specific categories are further grouped into higher classes forming the linguistic categories such as Nouns, Verbs, Adjectives, and Particles. Each Lexical entry describes the functional and semantic properties. The functional properties simulate the linguistic functional rules that would participate in identifying

the functional roles when producing the functional structure. The semantic properties simulate the semantic rules that would participate in identifying the Instances, Slot Names, and Slot Values when producing the semantic structure.

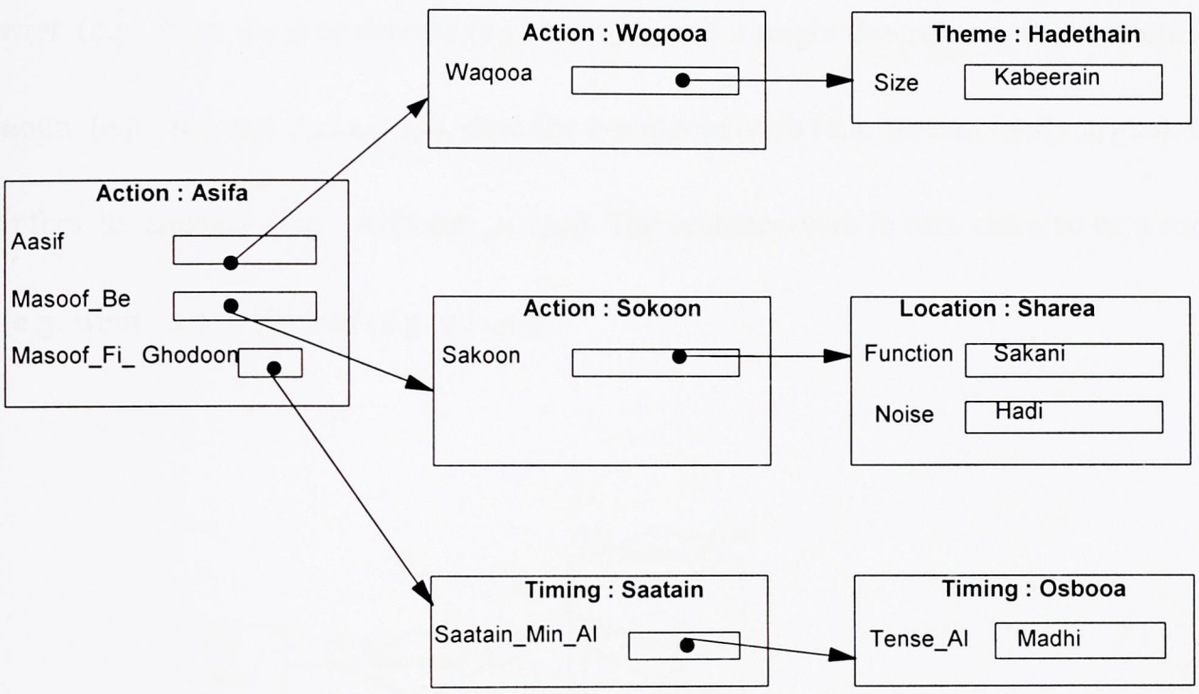


**Figure 11: Thematic roles frame hierarchy**

### 3.2.2 The Linguistic Structure

In order to identify the relationship between the text and its meaning, the structure of the language in which the text has been written has to be understood. The linguistic structure is modeled using the Entity Relationship Model [Elma-94]. The rectangles represent the entities and the diamonds represent the relationships among the entities. The two symbols separated by a comma and enclosed within parenthesis represent the cardinality of the relationships (e.g. (1,N)).

The prototype deals with two types of information, the *entry text* and the *lexicon*. The entry text is a group of words forming sentences, while the lexicon is an intelligent

representation of all the possible relationships between the words required to validate and understand the entry text.

Figure 12 shows that a *Sentence* may contain many smaller sentences and has none or many *sentence-verbs* and one or many *sentence-nouns*. The sentence-noun could be a *root* (e.g. man رجل) or *derived* (e.g. car سيارة) and it might describe another sentence-noun (e.g. red car سيارة حمـــراء), describe a sentence-verb (e.g. scream loud صاح بشدة), or refers to another (e.g. Ali's car سيارة علي). The sentence-verb in turn can also be a *root* (e.g. went ذهب) or *derived* (e.g. go يذهب).

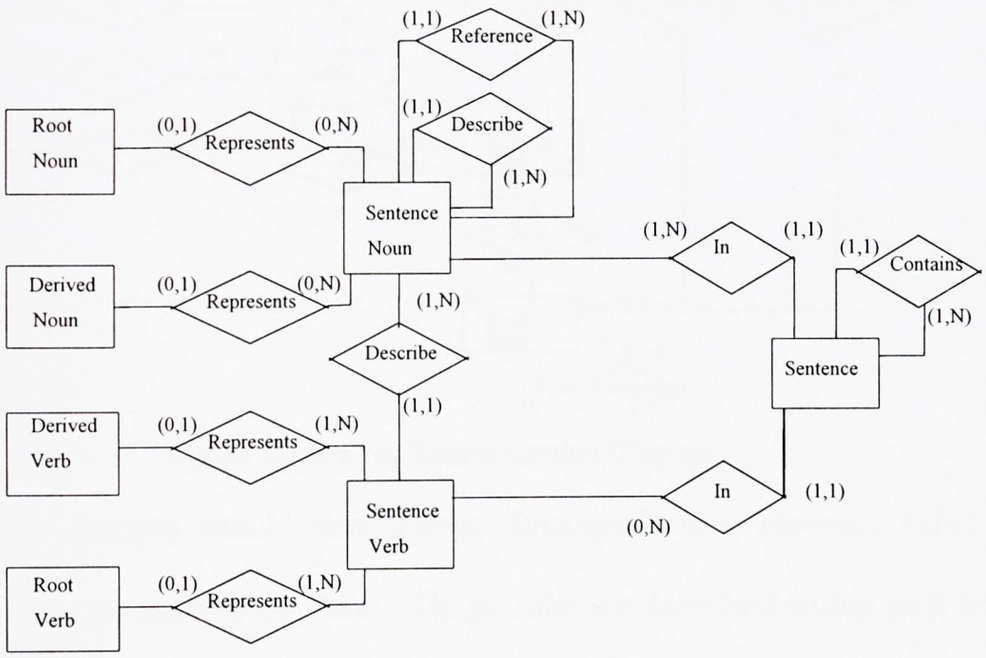


**Figure 12: Sentence Representation Diagram**

The lexicon should be a collection of lexical entries of the linguistic types *root-nouns*, *root-verbs*, and *particles*, see figure 13. The lexicon should also contain *affixes (i.e., Prefixes, Infixes, Suffixes)* along with *morphological rules* that are used for new words generation. The morphological rules build moulds by using the affixes with root-verb or root-noun to produce *derived-noun* or *derived-verb*. The derived-noun can be from

root-noun (e.g. 'two men رجلان' from 'man رجل' after suffixing it with 'ان') or from root verb (e.g. 'car سيارة' from 'moved سار' after infixing it with 'ـي' and suffixing it with 'ة'). The derived-verb can be from root-verb (e.g. 'go يذهب' from 'went ذهب' after prefixing it with 'يـ") or from root-noun (e.g. 'become rocky تصخر' from 'rock صخر' after prefixing it with 'تـــ'). There are words that require deletion of a letter from the root (e.g. حبة، حبات) or repeating the last letter (e.g. قل، قلل), etc.

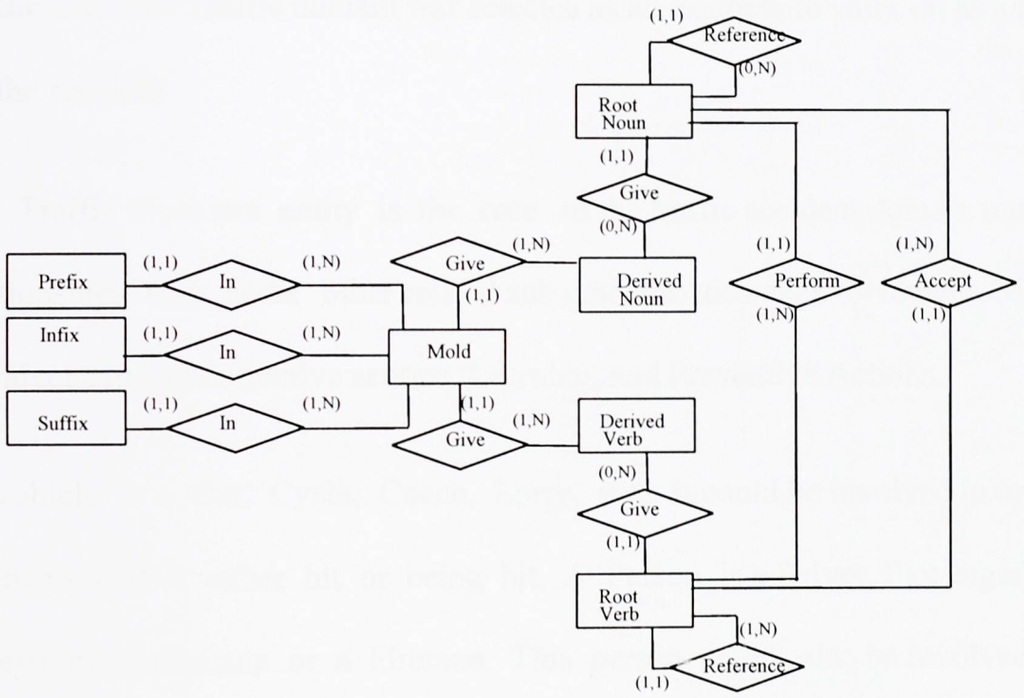


**Figure 13: Lexicon Representation Diagram**

The complete lexicon should also contain *functional rules, semantic rules,* and *domain specific common sense rules.* Those rules are described within each lexical entry as predefined information used to generate the functional, semantic and domain specific common sense knowledge structures.

### 3.2.3 The Semantic Representation

The domain object model shown in figure 14 was developed as a target to reveal how successful the natural language text conversion was and how close the system is towards automatic computer understanding. When all words, which appeared in the natural language text, are allocated in their proper representation in the domain object model either explicitly through the s-structure or implicitly through the k-structure, then and only then we can conclude that the system has captured most of the meaning of the text. The Traffic domain was selected as an example to work on as a case study for the research.

The Traffic Accident entity is the core of the traffic accident domain model, it has relationships with some other related sub domains such as Involvement, Monitoring, Accident causes, Corrective actions, Learning, and Preventive Actions.

A Vehicle is a Car, Cycle, Coach, Lorry, etc. It could be involved in one or more accidents and is either hit or being hit. A Person is a Driver, Passenger, Witness, Pedestrian, Policeman or a Fireman. This person could also be involved in one or many accidents. The person could cause an accident, be affected by it, see it, report it, or participate in the rescue. A Property could be hit by a Vehicle, and an Animal could be hit in one or many accidents.

A Person could cause one or many accidents by being reckless or by not being trained or by being ignorant of the traffic laws. Bad weather could participate in the causes of accidents. Lack of suitable maintenance of Vehicles and Roads could cause accidents as well.

Monitoring the occurrence of accidents could be achieved by different means. Police stations can do that through installed cameras on key roads, in addition to regular patrols. The Traffic jam gives a good indication of a possible occurrence of an accident ahead on the road. Damaged parts of the roads such as its signs are good indications of accidents. Persons could inform responsible authorities about some accidents.

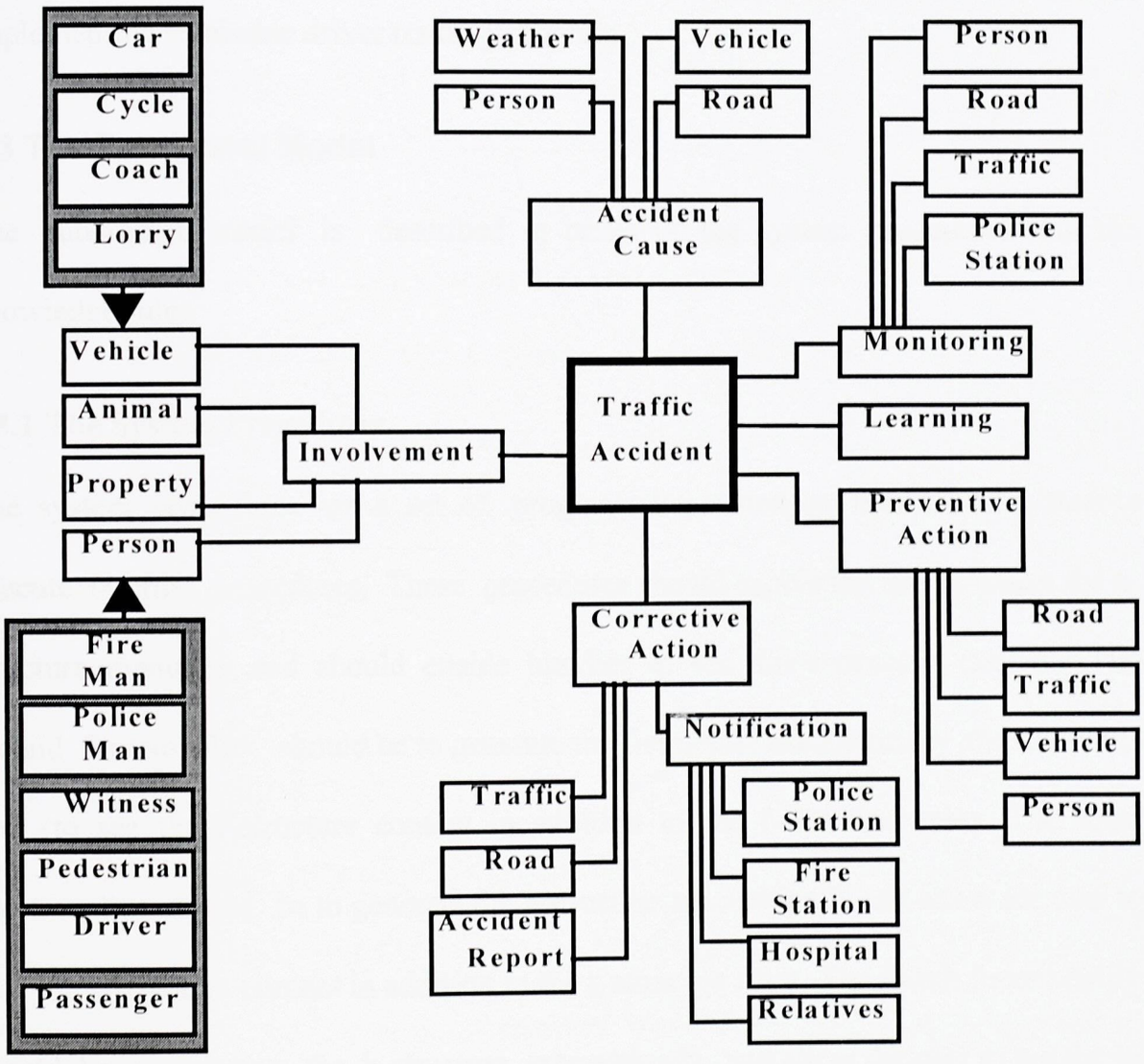


**Figure 14: Traffic domain object model**

Once an accident occurs, a number of corrective actions could be carried out such as notifying the Police Station, Fire Station, Relatives or the Hospital. The Police would detour the traffic, issue accident reports, while the Fire Brigade would clear the road

and rescue the trapped persons. The Ambulance would transfer the injured persons to hospital for treatment, while the Relatives would handle the rest of the relevant civil issues.

The learning process here could convert all the knowledge accumulated during the above activities into some suitable actions. In our case this could provide proper road maintenance, insuring continuous traffic flow, enforcing vehicle safety standards, and implementing a suitable driver training program.

## 3.3 The Functional Model

The functional model is described in terms of the system procedures and the knowledge rules.

### 3.3.1 The System Procedures

The system procedures are a set of programming statements used by the system to execute internal procedures. These procedures should enable the user to insert the c-structure manually and should enable him/her to see the c-structure contents. The second functionality should be to generate the f-structure automatically and allow the user to see the f-structure content in addition to the f-structure rules. The third functionality should be to generate the s-structure automatically and allow the user to see the s-structure content in addition to the s-structure rules. The fourth functionality should be to generate the k-structure automatically and allow the user to see the k-structure content in addition to the k-structure rules. The user should also be able to insert the lexical entries manually and be able to define the functional rules, semantic rules and the domain specific common sense knowledge rules. Figure 15 shows the data flow diagram developed according to the Gane and Sarson's notation [Fertuk-92].

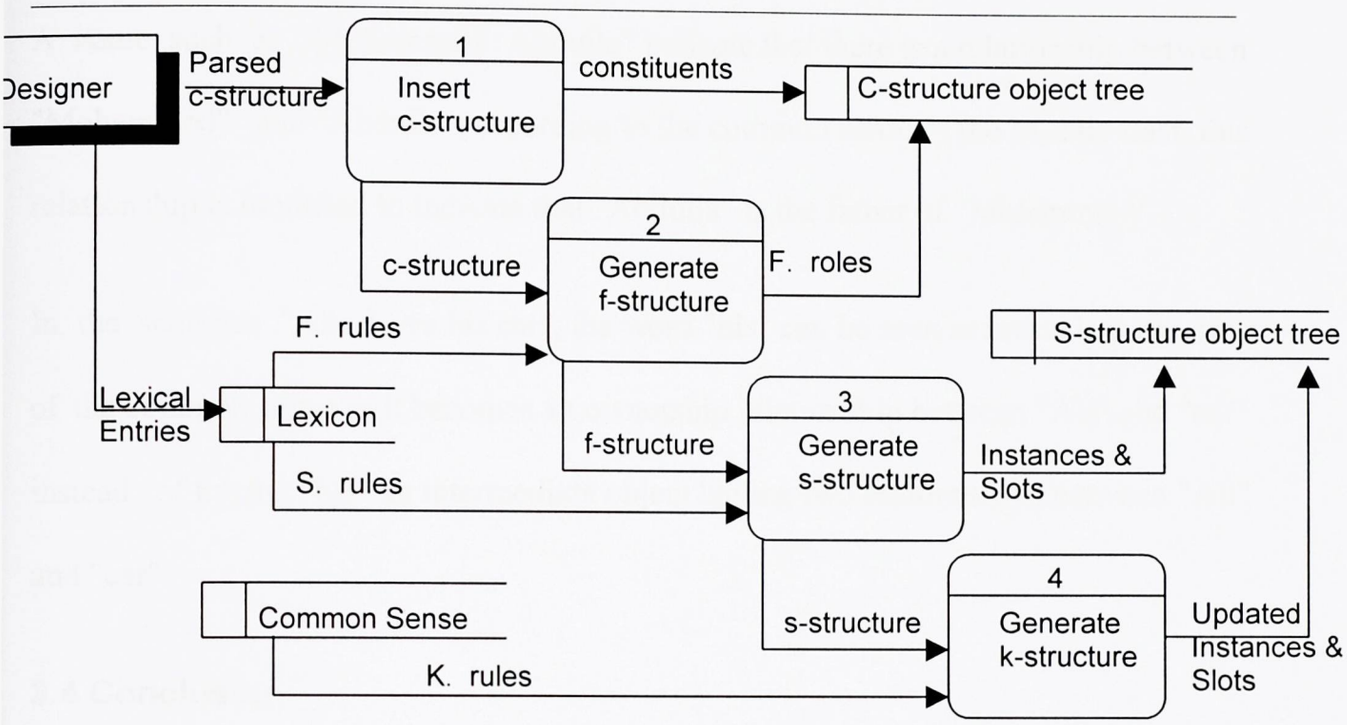


**Figure 15: Data flow diagram**

The system should pass four phases, the first phase is that the designer should manually insert the constituent structure and the lexicon definitions. The second phase is the generation of the functional structure from the constituent structure. The third phase is the generation of the semantic structure from the functional structure. The fourth phase is the generation of domain specific common sense structure from the semantic structure.

### 3.3.2 The Knowledge Rules

The knowledge rules are a set of statements that should be applied to a given piece of information to generate new facts. Based on certain information explicitly available in the text, and from domain specific common sense accumulated knowledge within certain community, a set of rules are formulated to generate extra information, modify existing information and remove redundant information. For example, if an accident is described as big, we can deduce extra information by domain specific common sense, that is the number of cars involved is many or the injuries are serious.

A name such as "Mohammed Abdulla" indicate that there is a relationship between "Mohammed" and "Abdulla", according to the common sense in the Middle East, this relationship is modified to indicate that "Abdulla" is the father of "Mohammed".

In the sentence "Ali drove his car", the word "his" can be seen as redundant in terms of the common sense as it becomes an ownership relationship between "Ali" and "car" instead of having "his" an intermediate object having two relationships between "Ali" and "car".

## 3.4 Conclusion

The requirements of constituent, functional, semantic, and the domain specific common sense in addition to the natural Arabic sentence and the lexicon structures were analysed in this chapter. These requirements present an important input towards completing the design phase and were analysed in view of the current structures of the Lexical-Functional Grammar theory and help visualize the need for modifying or extending the structures of the theory. The traffic domain object model was developed as an example to test the successfulness of the proposed system.

# Chapter Four

# The System Design

## 4.1 Introduction

This chapter describes the basic architecture of the proposed prototype in terms of input/output diagrams, production rules, algorithms, and inference trees. The Lexicon architecture is described in terms of categorised words within Nouns, Adjectives, Verbs and Particles tables.

The task of producing the constituent structure is not automated as it is a very mechanical process and has no research significance. Therefore it has not been considered in the design stage.

## 4.2 The System Input/output

The input/output diagrams describe the overall input-process-output mechanism.

### 4.2.1 The Functional Structure Input/output

The module that generates the functional structure should process the constituents' input from the constituent structure utilising the Lexicon and the functional rules, see figure 16.

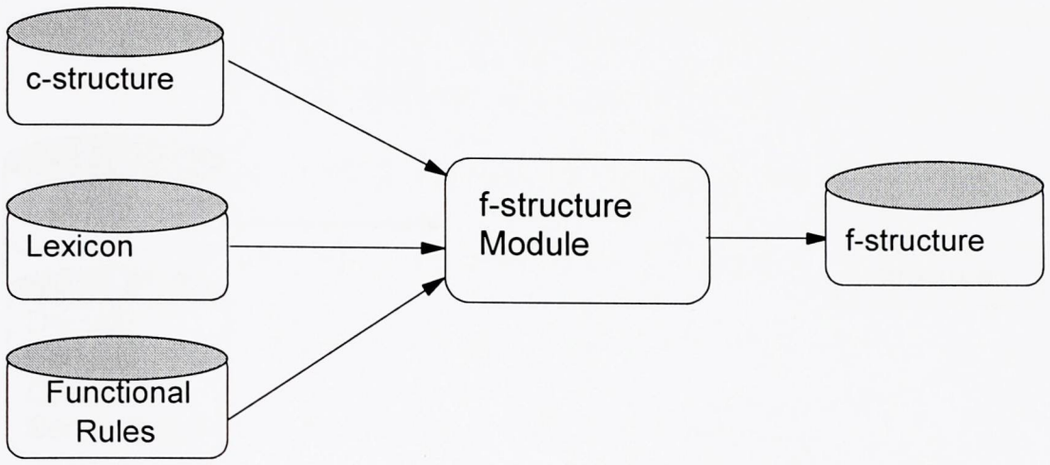


**Figure 16: Functional Structure input/output**

### 4.2.2 The Semantic Structure Input/output

The module that generates the semantic structure should process the words input from the functional structure utilising the Lexicon and the semantic rules, see figure 17.

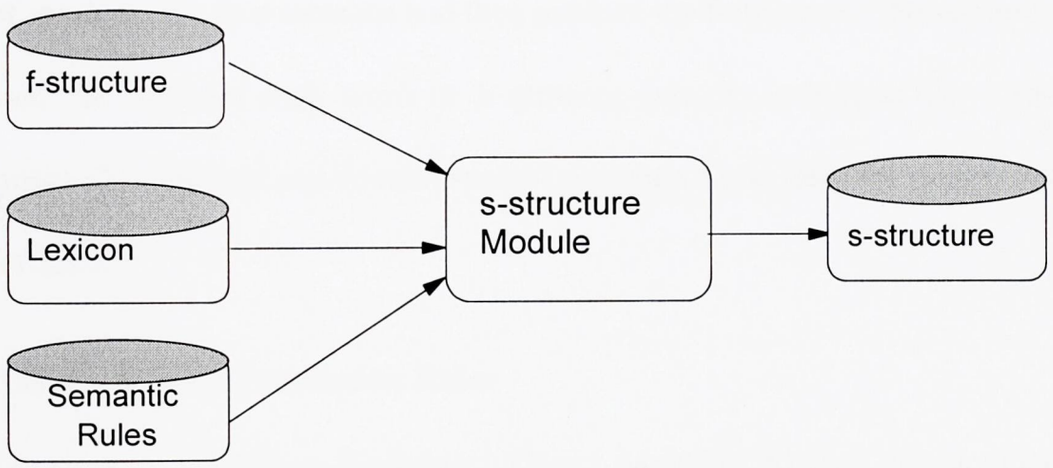


**Figure 17: Semantic Structure input/output**

### 4.2.3 The Common Sense Structure Input/output

The module that generates domain specific common sense structure should process the information input from the semantic structure utilising the Lexicon domain specific common sense rules, see figure 18.

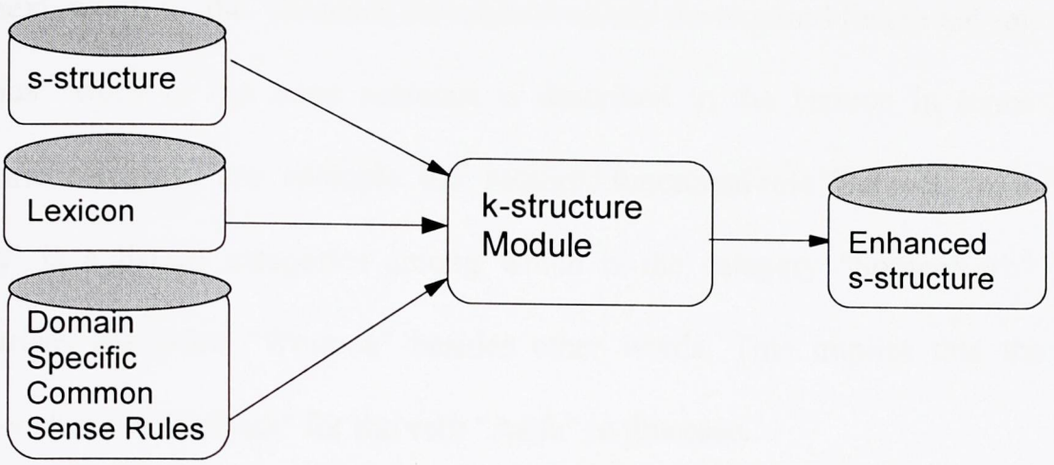


**Figure 18: Common Sense Structure input/output**

## 4.3 The Production Rules

The modules of the developed prototype use such rules to produce the different structures of the framework. The functional rules are used to identify the functional role of each word in a sentence and then produce the f-structure. The semantic rules are used to convert each word in a sentence into its corresponding s-structure component. The cultural and domain specific common sense rules are used to enhance the s-structure.

### 4.3.1 The Functional Production Rules

The functional roles such as the Subject, Object, Mubtada, Khabar, etc., are identified for each word according to the functional rules, which are in fact the Arabic language grammatical rules. The summary of the functional rules shown in figure 19 is derived from the nominal and verbal sentences and their sub phrases. These rules cover the twenty-nine sentences taken as an example within the traffic accident domain, see appendix B. There is a possibility to find out some more rules if other sentences from other domains are investigated.

The next word in the sentence that should satisfy the required functional role of the previous word in the same sentence is described in the lexicon in terms of its linguistic category. For example the required functional role "Subject" for the Verb "Asifa" is a list of categories among which is the category "SourceVerb" which categorises the word "Woqooa" besides other words. This implies that the word "Woqooa" is the "Subject" for the verb "Asifa" in this case.

**Atf** العطف: *a conjunction between 2 nouns , 2 adjectives, or 2 verbs.*
**Badal** البدل: *another specific description for an adjective or a digit for numeric noun.*
**Condition** الحال: *a description of a verb's condition.*
**Determinee** المعرّف: *a noun that follows a determinant.*
**FromObject** المفعول منه: *a description of the object from which a verb applies.*
**HowObject** حال المفعول: *an object that describes how a verb was applied.*
**Ism** الاسم: *a Mubtada for Kana or Inna or their sisters that is to be described later on.*
**Khabar** الخبر: *a description of previously mentioned Mubtada for Kana or Inna or their sisters*
**LocationalObject** ظرف المكان: *a description of the object at which a verb applies.*
**Majroor** المجرور: *a noun or verb that follows a preposition.*
**Matoof** المعطوف: *a noun or verb that follows a conjunction.*
**Mubtada** المبتدأ: *a noun that is to be described later on.*
**MudafElaih** المضاف اليه: *further clarification of a noun.*
**MutlaqObject** المفعول المطلق: *a noun that describes the type, number, or reiterates a previous noun.*
**Negated** الفعل المنفي: *a verb that has not occurred.*
**Object** المفعول به: *a noun that describes the action of the Subject using a verb.*
**Subject** الفاعل: *a description of who applies a verb.*
**TemporalObject** ظرف الزمان: *a description of when a verb applies.*
**ToObject** المفعول له: *a description of the direction of a verb's Object.*
**WhyObject** المفعول لاجله: *a description of a verb's reason.*

**Figure 19: Summary of Functional Roles**

Some of these rules are described in figure 20 in the form:-

IF *Word* is *Linguistic-Category* then *functional-role* is one or more of [*functional-rule*]

| | |
|---|---|
| Rule1: | If a word is a Verb Then the required functional role is one or more of [Subject, Object, Temporal Object, Locational Object, Conjunction] |
| Rule2: | If a word is a Noun Then the required functional role is one or more of [MudafElaih, Khabar, Conjunction] |
| Rule3: | If a word is a Proposition Then the required functional role is one or more of [Majroor] |
| Rule4: | If a word is an Adjective Then the required functional role is one or more of [Conjunction] |
| Rule5: | If a word is a Conjunction Then the required functional role is a word of the same category for the previous word |
| Rule6: | If a word is a Determinant Then the required functional role is [Determinee] |

**Figure 20: Functional Rules**

### 4.3.2 The Semantic Production Rules

The semantic rules are used to identify the semantic structure objects, in terms of the slot names, slot values, and instance names. These rules also classify the instances into their predefined subclasses, which are Actions, Themes, Timings, and Locations. Figure 21 describes some of these rules.

| | |
|---|---|
| Rule1: | All Verbs are represented as Instances in a Subclass called Actions. |
| Rule2: | All Sources "مصـــدر" derived from Verbs are represented as Instances in a Subclass called Actions. |
| Rule3: | The relationship between the Verb and its Subject is represented, in the Instance of the Actions Subclass in Rule1, as Slot Name (derived from the mold "Fael") and its value is the corresponding Subclasses identified in the f-structure that was identified as the Subject. |
| Rule4: | The relationship between the Verb and its Object is represented, in the Instance of the Actions Subclass in Rule1, as Slot Name (derived from the mold "Mafool" and suffixed with the proposition that is part of the Object) and its value is the corresponding Subclasses identified in the f-structure that was identified as the Subject or its anexes (MudafElaih).. |
| Rule5: | The relationship between the Verb and its TemporalObject is represented, in the Instance of the Actions Subclass in Rule1, as Slot Name, which is derived from the mold "Mafool" and suffixed with the proposition and/or unknown temporal circumstances that is part of the TemporalObject. The slot value is the corresponding Subclasses identified in the f-structure that was identified as the Subject or its annexes (MudafElaih).. |
| Rule6: | Propositions are part of the Slot name. |
| Rule7: | Unknown Circumstances are part of the Slot name. |
| Rule8: | The annexed Noun "MudafElaih المضاف اليه" for a "Source" of Rule2 is represented the Subject as in Rule3. |
| Rule9: | The Nouns other than those of Rule2 and Rule12 are represented as Instances in the Themes Subclass. |
| Rule10: | The adjective "Khabar خبر" is represented in terms of its parent in the Lexicon as the Slot Name, and it as the Slot Value in the corresponding Instance of that Action, Theme, or Timing Subclass. |
| Rule11: | If the Conjucted "Matoof معطوف" object is an adjective "Khabar" then is represented as in Rule10 corresponding to the same Instance. |
| Rule12: | The KnownTemporal Circumstance is represented as an Instance in the timings Subclass. |
| Rule13: | If the KnownTemporal Circumstance is part of Annexed "MudafElaih" sentence then it is referenced as the slot value in the previous (Annexed to) KnownTemporal Circumstance. |

**Figure 21: Design rules for the Semantic structure**

### 4.3.3 The Common Sense Production Rules

The k-structure is a set of rules that are extracted from a given domain depending on a certain community's culture. These rules when executed enhance the semantic structure to add more information or clarify some and delete the redundant data. A number of rules described below are extracted from the road traffic accident as the main domain. Figure 22 shows a template for the domain specific common sense update rule.

```
If      {[Condition (Thematic Role-1)]... [Condition (Thematic Role-n)]} OR
        {[Condition(Slot Name-1)]... [Condition(Slot Name-n)]} OR
        {[Condition(Slot Value-1)]... [Condition(Slot Value -n)]}
Then
        {[Update(Instance)] [Update(Slot Name)] [Update(Slot Value)]}
```

**Figure 22: k-structure update rule**

A number of rules can give the indication that an accident did happen. For example if a car hits another object such as a person, a property or another car. A car that rolls over itself or gets damaged could indicate that an accident occurred. See figure 23.

```
Rule1: If Vehicle X hits Vehicle Y Then an Accident has occurred
Rule2: If Vehicle X rolls over Then an Accident has occurred
Rule3: If a Person is Killed Or A Vehicle is Cancelled Then Accident Type is Catastrophic
```

**Figure 23: k-rules for accidents occurs**

A number of rules can make us predict that an accident could happen. For example a mechanical failure in the car, or bad weather conditions or non-compliant driving could cause accidents. See figure 24.

```
Rule4: If a Vehicle has mechanical Problems Then an Accident could occur
Rule5: If a Driver violates traffic Laws Then an Accident could occur
Rule6: If the Weather condition is Bad Then an Accident could occur
```

**Figure 24: k-rules for accidents could occur**

A number of rules can help in identifying those parties involved in an accident that has happened. For example a damaged car or property at the scene of the accident indicates that such parties are involved in an accident. A person who is in a car or property at the scene of the accident could also be considered as involved. See figure 25.

```
Rule7: If a Vehicle is damaged at the site of an accident Then this Vehicle is involved
Rule8: If a Property is damaged at the site of an accident Then this Property is involved
Rule9: If a Person is in an involved Vehicle Then the Person is involved
```

**Figure 25: k-rules for involvement in accident**

A number of rules can help in identifying the type of the person. For example a person is identified as a driver if he/she is siting in the vehicle behind the steering, the person is identified as a passenger if he/she is sitting on another seat in the vehicle. A pedestrian is the person who walks at the scene of the accident. See figure 26.

```
Rule10: If a Person is in a Vehicle And Behind the steering wheel Then the Person is a Driver
Rule11: If a Person is inside a Vehicle And Not in driving seat Then the Person is a
        Passenger
Rule12: If a Person is Not in a Vehicle And walking close to a Road Then the Person is a
        Pedestrian
```

**Figure 26: k-rules for person attribute**

A number of rules can give an indication of what sort of actions might be taken as a result of an accident. For example the hospital is to be notified if a person is injured, the fire station is notified if a person is trapped in a car and a policeman should come and issue an accident report about the accident. See figure 27.

```
Rule13: If a Person is Injured Then notify Hospital
Rule14: If a Person is trapped Then notify Fire Station
Rule15: If an Accident occurred Then issue an Accident report
```

**Figure 27: k-rules for action to be taken**

A number of rules can identify what sort of actions should be taken to prevent future accidents. For example avoiding traffic congestion and maintaining an acceptable vehicle and road safety standards. Ensuring acceptable knowledge and performance on the part of the driver also helps in preventing accidents. See figure 28.

Rule16: If Traffic flow is congested Then detour some vehicles to other roads
Rule17: If Vehicle safety mismatch standards Then request to match standards
Rule18: If Driver performance mismatch standards Then request to attend suitable training

**Figure 28: k-rules for preventive actions**

A number of rules can give an indication whether an accident did happen. For example if a person reports an accident then it is more likely that an accident did occur. A traffic jam is an indication of a possible accident occurrence. See figure 29.

Rule19: If a Traffic jam is observed Then check accident occurrence
Rule20: If a person informs about an accident Then check accident occurrence

**Figure 29: k-rules for whether an accident did happen**

The vision about learning here is that whatever is understood from the other sub domains is to be reflected in the Preventive actions, Monitoring, and Accident causes sub domains.

The most important knowledge rules are those defining the relationships among the instances of each sub domain. These rules complete the semantic structure in more detail. For example we can know the father name from the second name, and the owner of the tool from the following noun, See figure 30.

Rule21: If a person name is followed by another person name Then the second is the father of the first
Rule22: If a person name follows a tool name Then the person is the owner of the tool
Rule23: If the Age of a person is mentioned Then his Birth Date is the current date minus his age

**Figure 30: k-rules for relationships among Instances**

## 4.4 The Generation of various structures

The generation process of the functional, semantic, and domain specific common sense structures are described in terms of high level algorithms and detailed flowcharts.

### 4.4.1 Generating the Functional Structure

Generating the functional structure is described in the flowchart of figure 31 that is described in the following abstract steps:-

1. Get the constituent structure
2. Get the first word in the sentence
3. Identify its functional role
4. Get its functional rules one at a time
5. Match the next words with each rule
6. Identify their functional role
7. Repeat steps 3 to 6 for all words in the sentence
8. Repeat steps 1 to 7 for all sentences

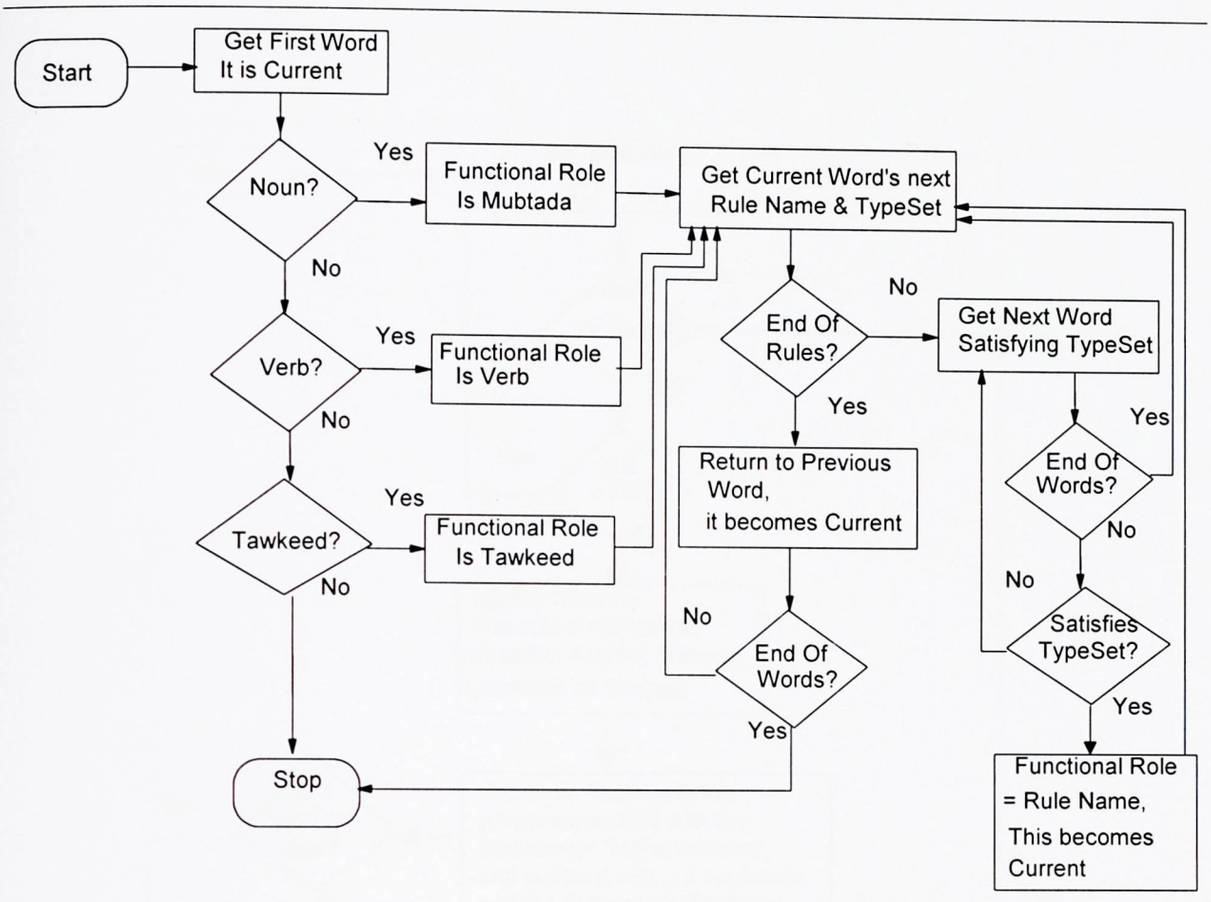


**Figure 31: Functional Structure Flowchart**

### 4.4.2 Generating the Semantic Structure

Generating the semantic structure is described in the flowchart of figure 32 that is described in the following abstract steps:-

1. Get the functional structure
2. Get a word at a time
3. Identify the database objects
4. Get word's functional structure requirements
5. Identify the database relationships
6. Repeat steps 4-5 for all functional requirements
7. Repeat steps 2-6 for all words
8. Repeat steps 1 to 7 for all sentences

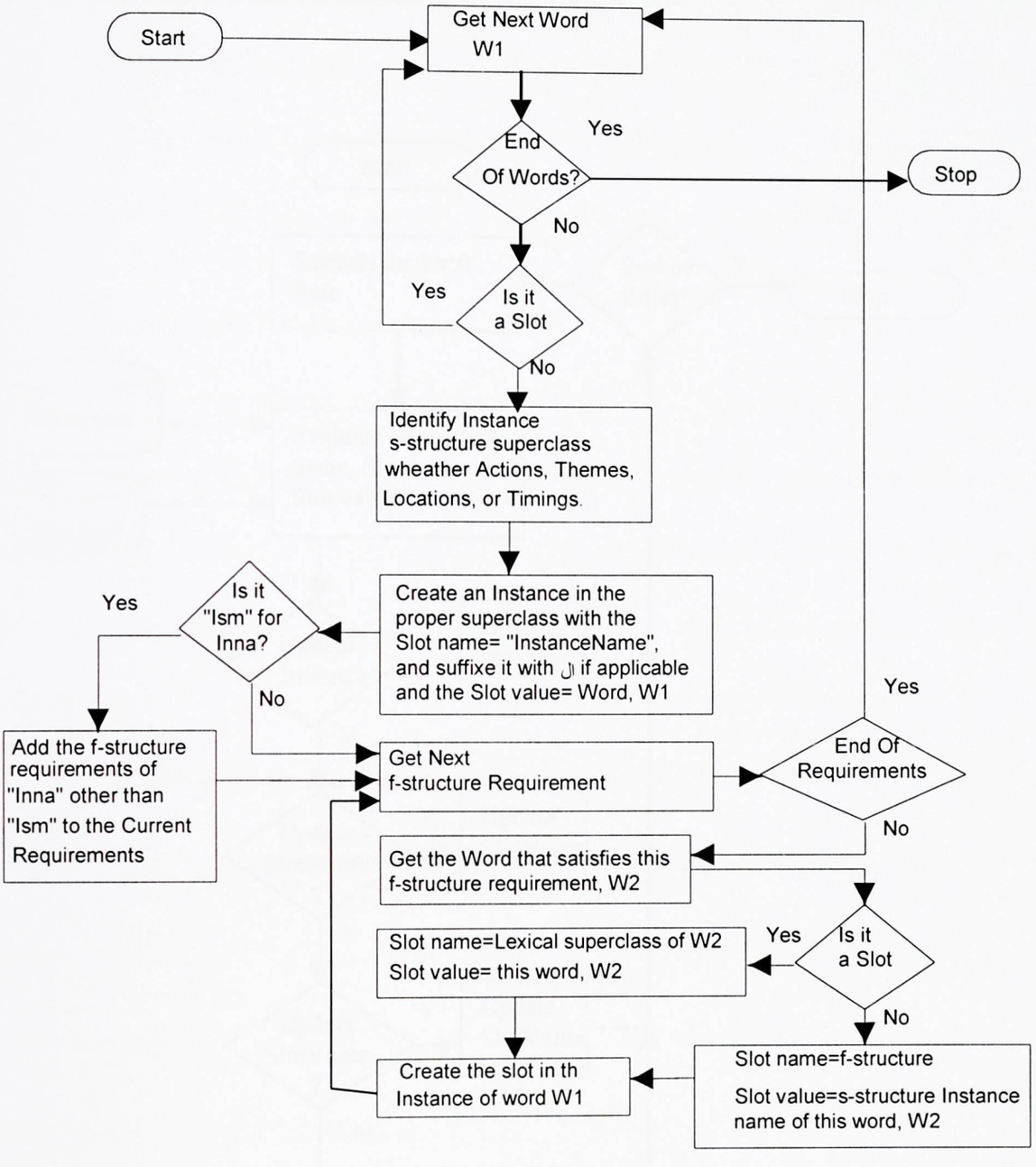


**Figure 32: Semantic Structure Flowchart**

### 4.4.3 Generating the Common Sense Structure

Generating the domain specific Common Sense structure is described in the flowchart of figure 33 that is described in the following abstract steps:-

1. Get the semantic structure
2. Execute next rule set
3. Evaluate the semantic objects

4. Update the semantic objects

5. Repeat step 2-4 for all the rules

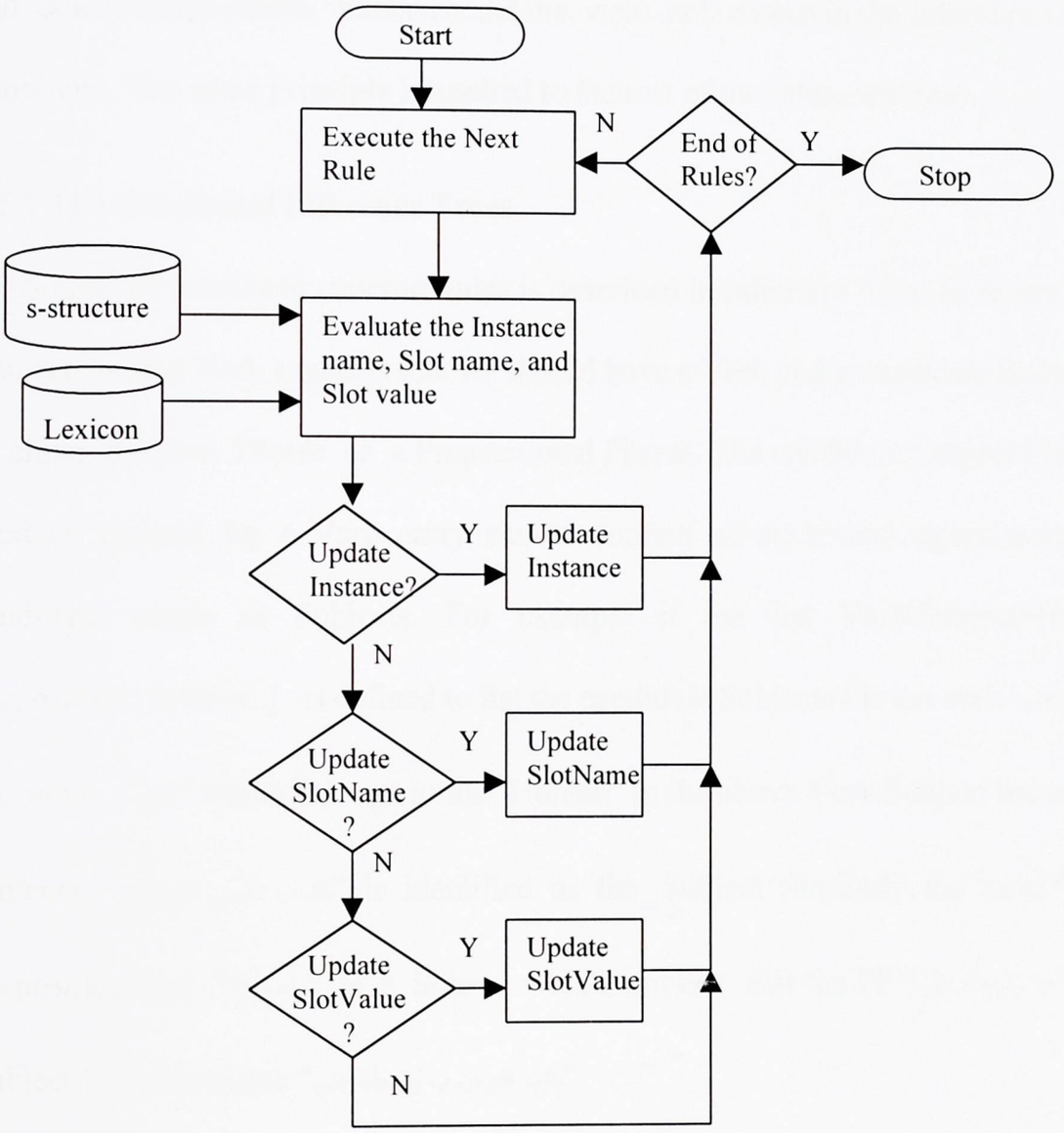


**Figure 33: Common Sense Structure flowchart**

## 4.5 The Inference Tree

The inference tree is a graphical representation for the framework structure rules using the formats in [Igni-91]. The rectangles represent an assertion, which is a category of a lexical entry or sometimes the words. The triangles represent the OR while the half circles represent the AND. The Conclusion is written inside a circle, and a circle

inside a square represents an intermediate conclusion. The arrows show the rule flow from the input to the output. Some examples are described in the opposite rectangle to clarify the inference tree. We know that the NP, for example, is composed of nouns and other complements, but such details were not shown in the inference trees for simplicity. The same principle is applied to the rest of the inference trees.

### 4.5.1 The Functional Inference Trees

A sample of functional structure rules is described in inference trees. In figure 34, the Subject of any Verb suggests that we should have a Verb and a candidate Subject that is either a Noun Phrase or a Prepositional Phrase. The candidate Subject List at the Lexicon defined for a Verb entry should contain all the lexical superclasses of the candidate words as Subjects. For example if the list Verb.Subject=[Human, Preposition, Source..] is defined to list the candidate Subjects for the verb "كتب" , Then the noun "علي" which belongs to the "Human" in the above Verb.Subject list as in the sentence "كتب علي الدرس" is identified as the Subject. Similarly, the word "ب" is a Preposition and "واسطة" is a Source which implies that the PP "ب واسطة علي" is the Subject in the sentence "كتب الدرس ب واسطة علي".

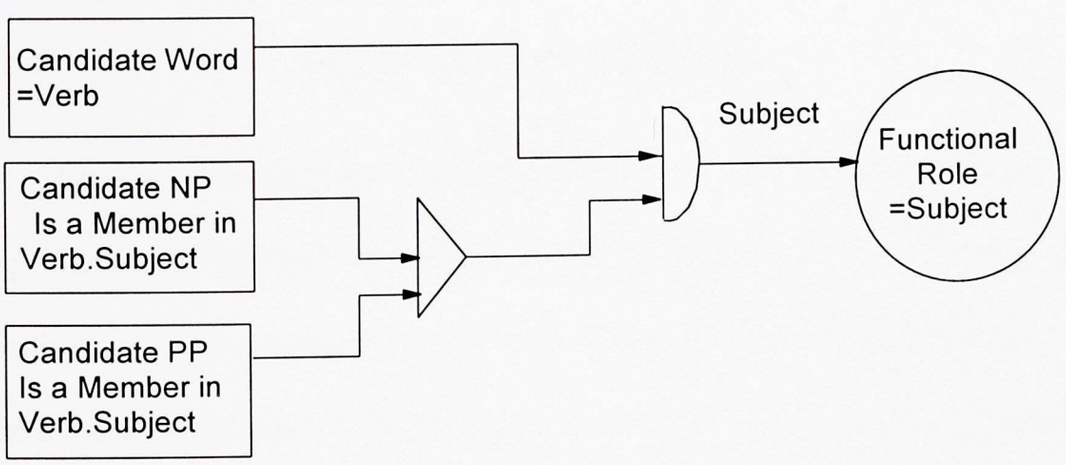


**Figure 34: Subject rule Inference Tree**

Figure 35 below shows that the Object of any Verb is either a Noun Phrase, a Prepositional Phrase, or Tawkeed. The Object List at the Lexicon of a Verb entry should contain all the lexical superclasses of the candidate words as Objects. For example If the acceptable candidate word as an Object is of a class in the list Verb.Object=[Tool, Proposition, Tawkeed..] for a Verb such as "قال", Then the Object is the NP "السيارة سـريعة" which starts with the word "السيارة" which belongs to the class Tool in the sentence "قال علي السيارة سريعة". Similarly in the sentence "قال علي ان السيارة سريعة" the Tawkeed "ان السيارة سـريعة" is the Object, and in the sentence "قال علي ب ان السيارة سريعة" the PP "ب ان السيارة سريعة" is the Object.

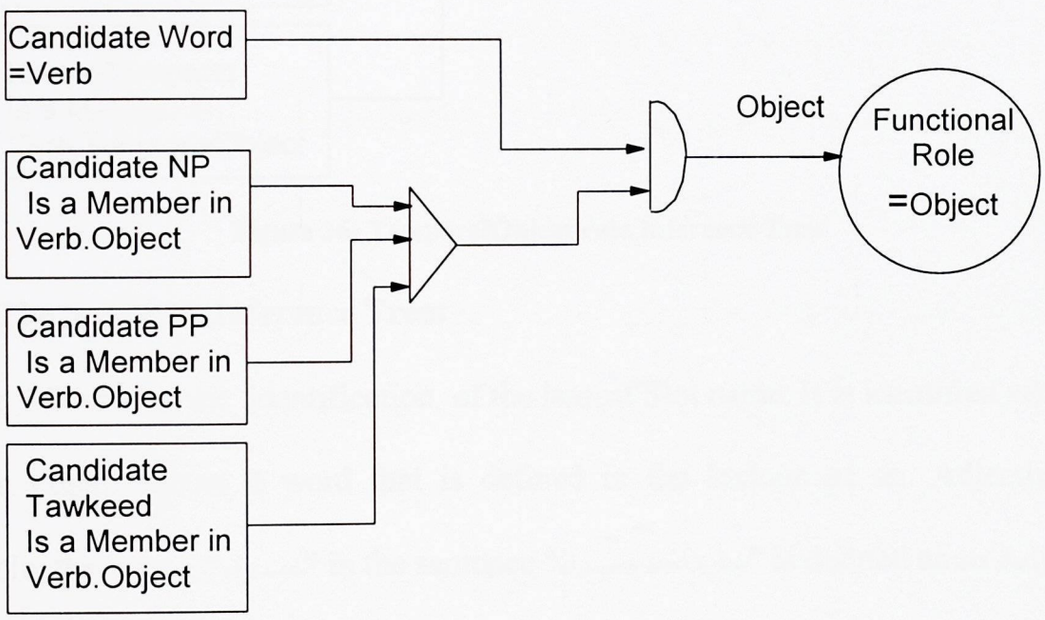


**Figure 35: Object rule Inference Tree**

Figure 36 below shows that the TemporalObject of any Verb is a Prepositional Phrase having Known or UnKnown Temporal word as the Majroor (i.e., the next word to the proposition). The TemporalObject List for a Verb entry in the Lexicon should contain all the lexical superclasses of the candidate words as TemporalObject. For example If the Verb.TemporalObject=[ Preposition , KnownTemporal, UnKnownTemporal..] for

Verb= "جـاء" Then the word "امس" which belongs to KnownTemporal in the sentence "جاء علي امـس" is the TemporalObject. Similarly in the sentence "جاء علي في امس" the PP"في امـس" is the TemporalObject, and in the sentence "جـاء علـي قبـل امـس" the UnKnownTemporal "قبل" is the TemporalObject.

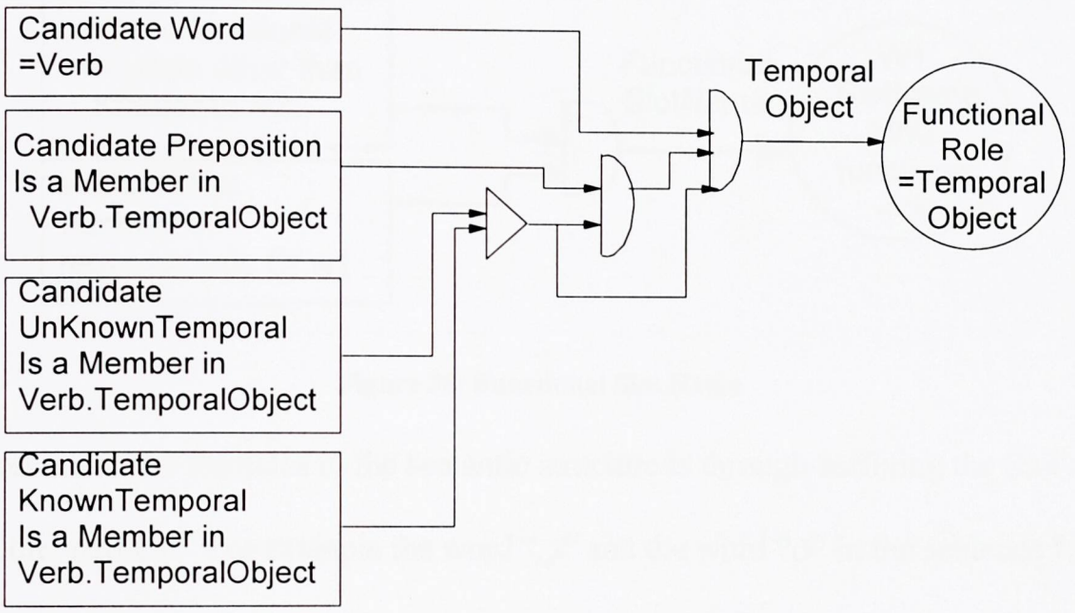


**Figure 36: TemporalObject rule Inference Tree**

### 4.5.2 The Semantic Inference Trees

Figure 37 shows the identification of the lexical Slot name. It is identified when the system comes across a word that is defined in the lexicon as an Adjective. For example the word "حمـراء" in the sentence "سيارة احمد حمـراء" is defined as an Adjective which belongs to the Lexical Superclass "Color" which implies creating a slot having the name "Color".

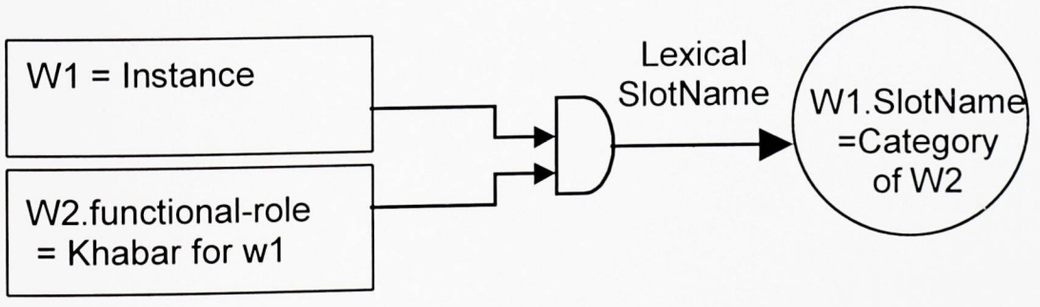


**Figure 37: Lexical Slot Name**

A Slot name is also identified according to the functional role of a word in the f-structure. For example the word "احمـد" has the functional role "MudafElaih" for the previous word "سـيارة" in the above sentence, this implies creating a slot having the name "MudafElaih", see figure 38.

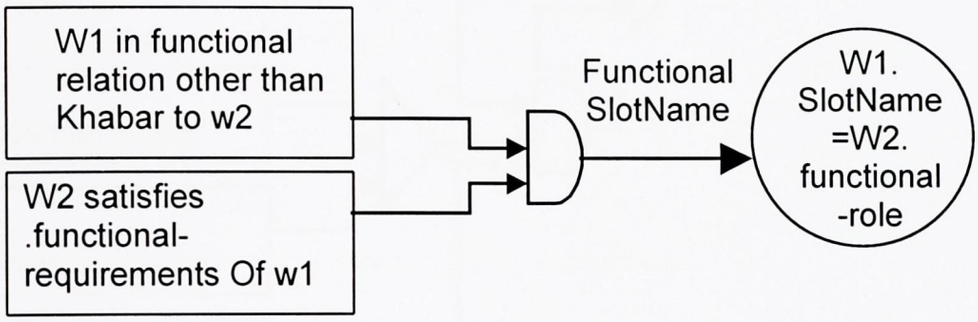


**Figure 38: Functional Slot Name**

Representing the Particles in the semantic structure is through suffixing the Slot name with the particle. For example the word "في" and the word "ال" in the sentence "قلمي في الـشـنطة" are defined as Particle which implies creating a slot having the two words as part of its name in this case "Khabar_في_ال". Suffixing Slot name can also be achieved when the system comes across a word that is defined in the lexicon as an UnknownTemporal or UnknownLocational provided that their functional role is not found to be "Determinee" (i.e., preceded with the particle "ال"). For example the word "خلـف" in the sentence "وقفت خلف الباب" gives use to a slot name such as "Object_خلف_ال", see figure 39.

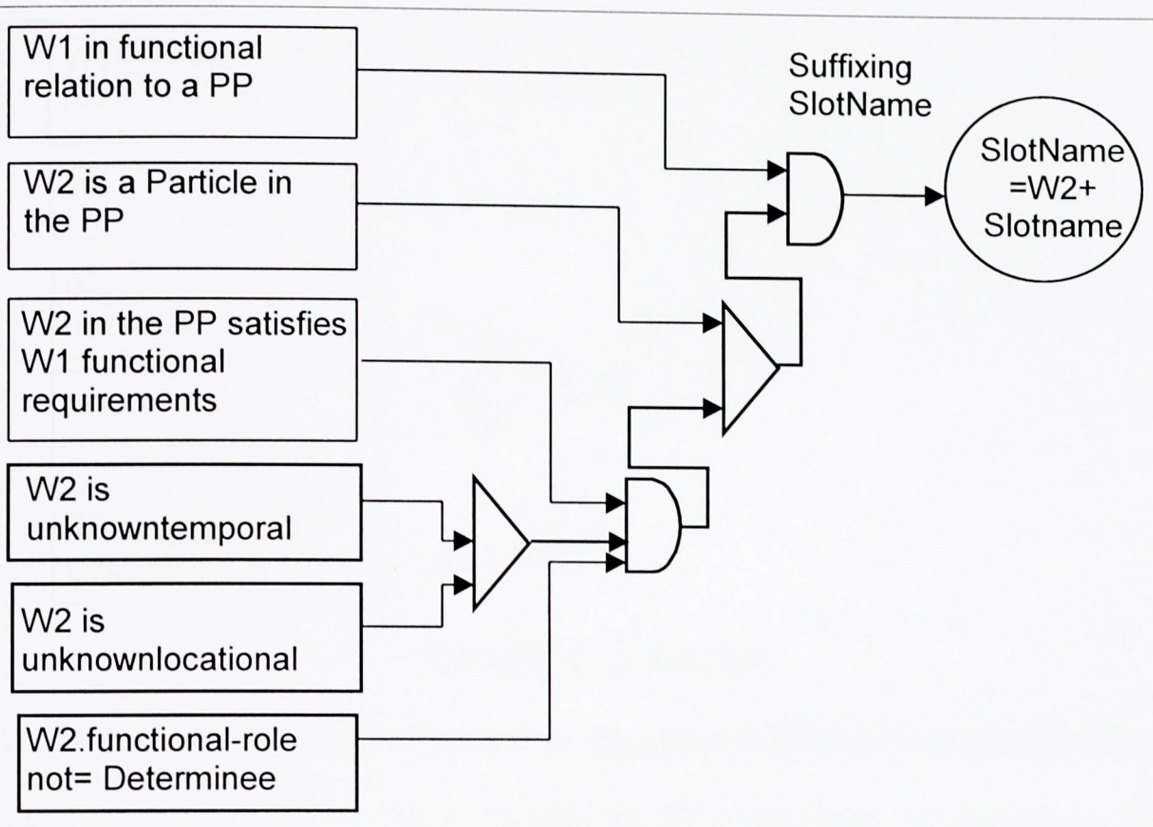


**Figure 39: Suffixing Slot Name**

Words that are adjectives are represented as direct slot values. In the above example, the word "حمـــراء" is the value of the slot name "Color". Direct slot values can also be identified when the system comes across a word that is defined in the lexicon as an UnknownTemporal or UnknownLocational provided that their functional role is found to be "Determinee" (i.e., preceded with the particle "ال"). For example the word "خلف" in the sentence "رجعت الى الخلـــف" results in a slot name such as "Object_الى_ال" and slot value such as "خلف", see figure 40.

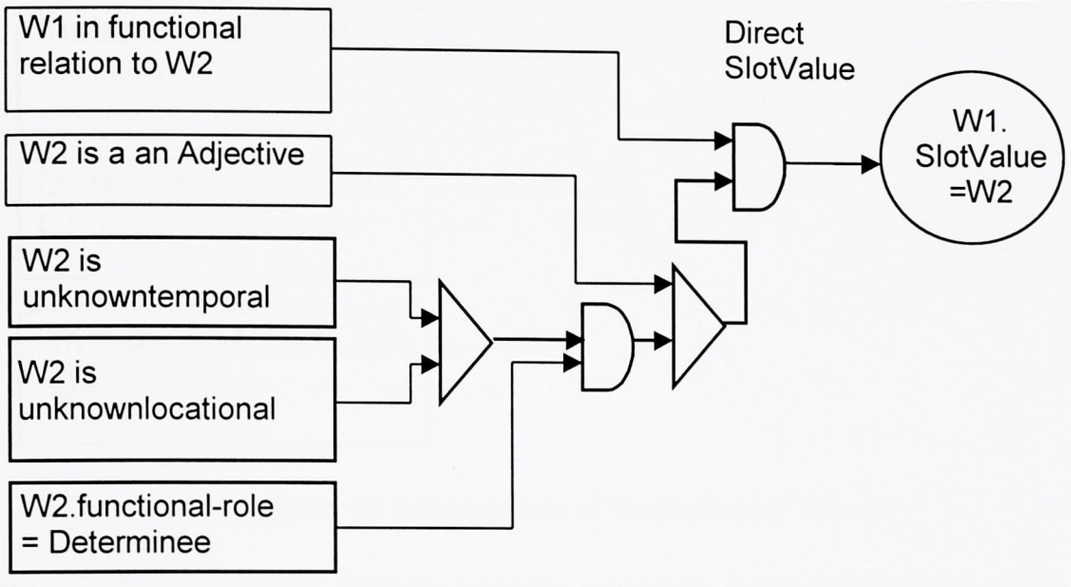


**Figure 40: Direct Slot Name**

Words that are identified as Instances in an earlier stage in the semantic structure are linked to their related instances through having their names as slot values, hence identifying indirect slot values, see figure 41.

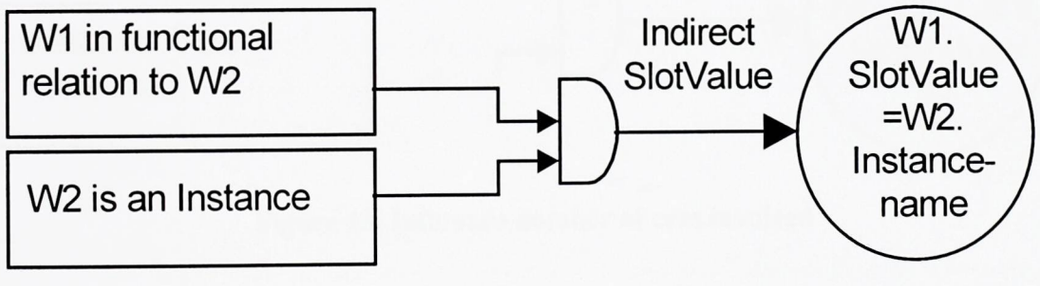


**Figure 41: Indirect Slot Name**

### 4.5.3 The Common Sense Inference Trees

In the Arabic culture, the "MudafElaih" Slot Name in an Instance is changed to "FatherName" by the rule: "If the category of the value of the Instance name is "Human", then this Instance has a Slot name = "MudafElaih", and The category of the value of this "MudafElaih" Slot is "Human"" (see figure 42).

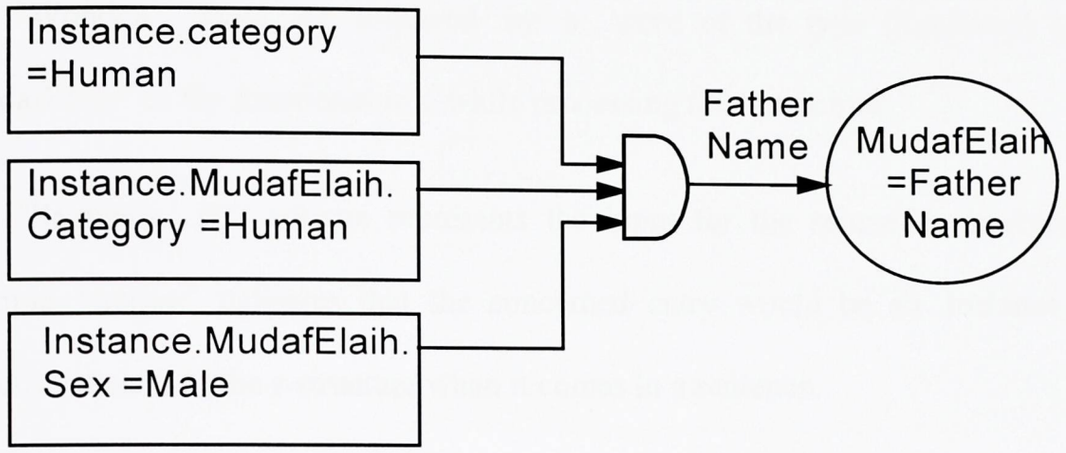


**Figure 42: Inference tree of "FatherName" rule**

Figure 43 shows another domain specific common sense rule when a new slot is created. If an accident is described as a big accident, then in a certain culture this implies that the number of cars involved is 3 or more.

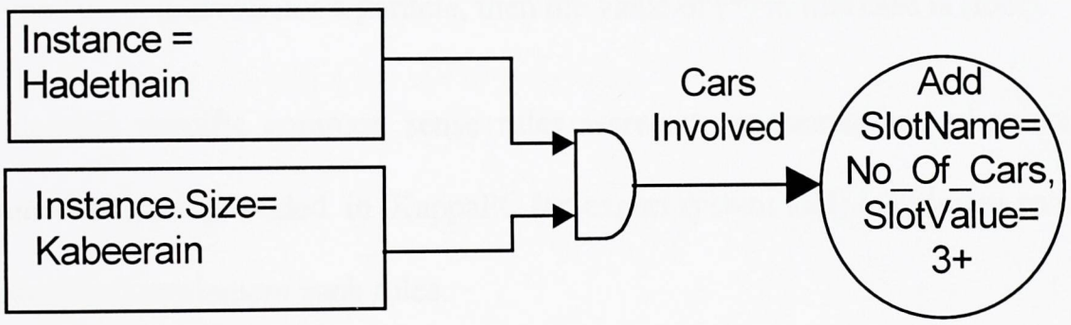


**Figure 43: Inference number of cars involved**

## 4.6 The Lexical Entry Representation

Lexical entries are described in the form of tables, which have a number of columns. For example Table 1 describes the lexical category "Noun". The Sub Category column groups the lexical entries that help in producing the generic Functional, Semantic, and domain specific Common Sense Knowledge rules.

The "Functional Rules Required" column represent the input for the Functional structure, for example the requirement "MudafElaih" for the noun "Woqooa" indicates

that "Woqooa" could be followed by a word of the type (Subjectal) having "MudafElaih" as the functional role while processing the f-structure.

The "Semantic ID" column represents the input for the semantic structure. For example "Action" indicates that the concerned entry would be an Instance in the Actions subclass in the s-structure when it comes in a sentence.

Similar entries are presented in table 2 for Adjectives, table 3 for Verbs, and table 4 for Particles. Note that Functional Rules Required for the Particles are marked with (*). This means that these requirements are dependent on the previous word's requirements. For example if the previous word is a verb that requires an "object" of the type "tool" that follows a particle, then the value of (*) in this case is (tool).

The domain specific common sense rules were not represented in the Lexicon. The inference engine provided in KappaPC (an expert system tool) is efficient enough to represent and implement such rules.

**Table1: Nouns category in the Lexicon**

| Sub Category | Noun | Functional Rules Required | Semantic ID |
|---|---|---|---|
| SourceVerb | Woqooa | MudafElaih(Subjectal) | Action |
| SourceVerb | Sokoon | Khabar(Adjective) | Action |
| Location | Sharea | Khabar(Adjective) | LocationalObject |
| Subjectal | Hadethain | Khabar(Adjective) | Themes |
| UnKnown Temporal | Ghodoon | MudafElaih (KnownTemporal) | SlotNameSuffix |
| Known Temporal | Saatain | MudafElaih(Proposition) | TemporalObject |
| Known Temporal | Osbooa | Khabar(Determinant, Adjective) | TemporalObject |

**Table2: Adjectives category in the Lexicon**

| Sub Category | Adjective | Functional Rules Required | Semantic ID |
|---|---|---|---|
| Size | Kabeerain | Atf(Wa) | Slotname |
| Function | Sakani | Atf(Wa) | Slotname |
| Noise | Hadi | Atf(Wa) | Slotname |

**Table3: Verbs category in the Lexicon**

| Sub Category | Verb | Functional Rules Required | Semantic ID |
|---|---|---|---|
| Transitive | Asifa | Subject (SourceVerb),<br>Object (Proposition(Be),<br>SourceVerb) | Actions |

**Table4: Particles category in the Lexicon**

| Sub Category | Particle | Functional Rules Required | Semantic ID |
|---|---|---|---|
| Atf | Wa | Matoof(*) | SlotNameSuffix |
| Preposition | Be | Majroor(*) | SlotNameSuffix |
| Determinant | Al | Determinee (*) | SlotNameSuffix |

## 4.7 Conclusion

This chapter has designed the necessary components to generate the functional, semantic, and domain specific common sense structures according to the basic and extended Lexical-Functional Grammar framework. This chapter has also designed the necessary components of the lexical entries along with their functional and semantic rules that serve in generating the above structures.

# Chapter Five

# Implementation

## 5.1 Introduction

The prototype is developed in KappaPC V2.3 for Windows 95 with Arabic support on an IBM compatible 486 processor or higher. KappaPC is an Object Orientated development tool that is suitable for research purposes. It provides the required tree representation for the theoretical structures (C, F, S, and the Lexicon) for the Natural language sentences in addition to the domain specific Common sense rules (k) inference engine. KappaPC also provides the programming functions to manipulate the represented tree. KappaPC best suited the implementation of this work as compared to other languages such as Prolog because i) it is a rapid prototyping and development object orientation tool, ii) it provides a knowledge representation and inference facility, iii) it provides an Arabic language user interface.

A user interface was built incorporating the Arabic alphabet that allowed actual Arabic text to be manipulated. Figure 44 shows the main menu of the prototype, which contains options for the constituent structure, functional structure, semantic structure and domain specific common sense structure.

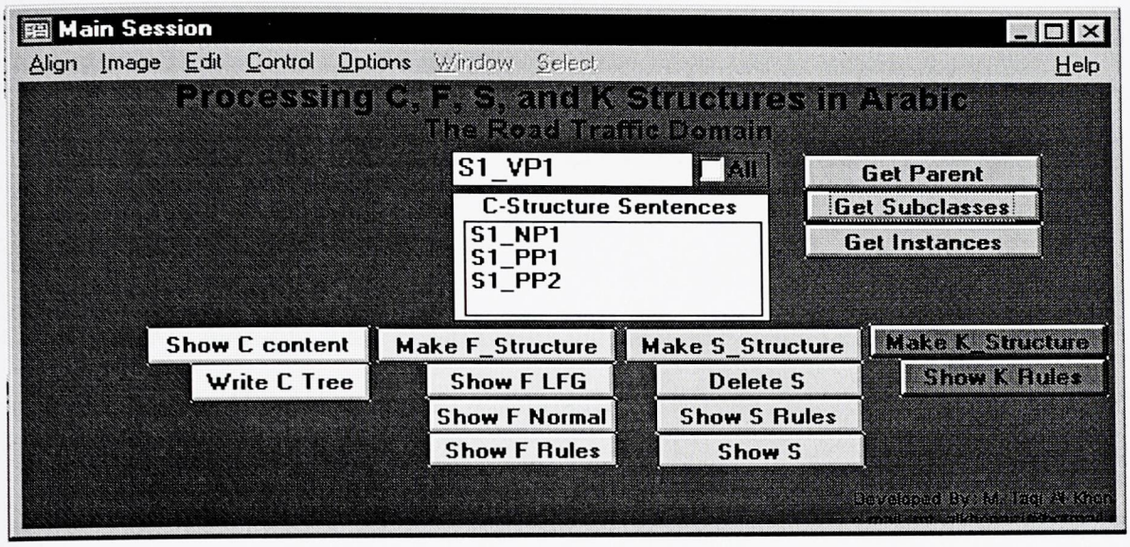


**Figure 44: Main menu of the prototype**

The database is implemented through having three hierarchies, the first accommodates the C and f structures together, the second accommodates both the S and K structures, and the third is for the Lexicon.

## 5.2 Assumptions

The assumptions are a number of conditions that are beyond the direct scope of this work, but assumed to be available in order to simulate the complete picture.

### 5.2.1 Constituent Structure Assumptions

1. The c-structure is available and correct
2. Processing is performed a sentence at a time, excluding words that link sentences (e.g. Then ثم، where حيث).
3. Diacritisation to identify the word function (e.g., رجلُ is Subject {with Dhamma} where رجلَ is Object {with fatha}, hence one lexical entry) is not required in the Natural sentence.
4. Normalisation is performed where possible. This includes applying the morphological rules manually and converting the compound sentence into its simple sentences, see figure 45.

5. Special characters such as commas, parentheses, are not processed.

The Arabic sentence:

قالت ان كل العربات التي كانت في الحادث قد تم فحصها وكل من لديه معلومات عليه الاتصال برقيب الشرطة (سوندرز) على هاتف رقم (0181-3-11212).

The transliteration:

Qalat Inna Kul AlArabat Allati Fi AlHadeth Qad Tamma Fahsoha Wa Kul Man Ladayhe Malomat Alayhe Alltisal BeRaqeeb AlShorta (Sonders) Ala Hatif Raqam (0181-3-11212)

The translation:

She said that all cars that were involved were tested and all those who have information should contact the police sergeant on number (0181-3-11212)

The Normalisation

قال ت ان كل ال عربات التي كان ت في ال حادث تم فحْص ها

ان كل مَن لدى ه معلومات على ه ال اتصال ب رقيب ال شرطة سوندرز على هاتف رقم ر11212_3_0181

The transliteration of Normalisation:

Qal t Inna Kul Al Arabat Allati Fi Al Hadeth Qad Tamma Fahso ha

Kul Man Laday he Malomat Alay he Al Itisal Be Raqeeb Al Shorta Sonders Ala Hatif Raqam

0181-3-11212

**Figure 45: Normalisation assumption**

### 5.2.2 Functional Structure Assumptions

1. Each word in a sentence is functionally linked to the rest of the words back to the first word, which is the main predicate in the sentence.
2. Number and applicability of the f-structure rules are as derived from the sample natural Arabic text described earlier.
3. The functional role is associated with a word but applicable to its block of words, which have their own functional roles. For example, the functional role Khabar associated with "In" the first word in the PP "In the bag" means all the PP is Khabar while "the" is Majroor, again all NP "the bag" is Majroor and "bag" is Determinee.

4. More than one word can satisfy one rule in a sentence (e.g. three Khabars can be for one Mubtada as in "Ali is Clever, Handsome, and Polite")
5. A functional role of a word w1 for w0 can be overridden by a closer word w2 to w1. For example in the sentence "ان علي ه ال اتصال", "ه" is first identified as Ism for "ان", but overridden as Majroor for "على" since "على" is closer to "ه" than"ان".
6. The Rule's priority is defined in the Rule's Requirements list sequence. For example the Ism comes before the Khabar in the "ان" Rule's Requirements list. This means that a word that can satisfy both rules would be Ism (e.g., the Pronoun "ه" can be Ism if it comes alone in the next NP after "ان" because of the priority, but Khabar when it comes as Majroor in a PP.
7. The Functional role "Object" is sometimes given a temporary qualifier to be such as "ToObject" or "FromObject". This is a better meaning than saying "Object2" or "Object3" especially when this is going to be replaced during the K-structure phase.

### 5.2.3 Semantic Structure Assumptions

1. Any word becomes one of an Instance, slot name or suffix of slot name, or a slot value.
2. Any Instance is grouped in one of Actions, Themes, Timing or Locations subclasses.
3. When a word is an Instance then all its f-structure requirements become its slots (e.g., MudafElaih).
4. The computer understanding of the natural Arabic language text is achieved by building the proposed Object oriented database prototype. In other words, if the anticipated number of instances are created and the relationships between them are

put in place together with the correct slot names and values, then the understanding is achieved.

### 5.2.4 Common Sense Structure Assumptions

1. Traffic accident is the main domain, and the sub domains are, accident involvement, Monitoring, Causes of Accidents, Corrective Actions, Learning, and Preventive Actions
2. Only sample rules are implemented for the main domain and each sub domain found in the selected traffic accident text.
3. The k-structure is a set of domain specific common sense rules that add or delete Instances, or changes slot names or slot values of the Instances.
4. Certain rules are implemented according to the local culture e.g. a father's name being the second name of a person in the Arab community.

### 5.2.5 Lexicon Assumptions

1. Each word is categorised correctly in the lexicon
2. Morphology results are bypassed. For example, there is a Lexicon entry for "سيارات Cars" although it is the plural of "سيارة A Car"). Another example is that the proposition "على Over" is written with the letter "ى" (without the two dots under it) at its end when it comes alone, while "ى" is converted into "ي" (with the two dots under it) when it is followed by "ه" as in "عليه".
3. Digits must be prefixed by any character from the alphabet because KappaPC object names must start with an alphabetical character. For example "26" should be prefixed with "ر" to look like "ر26" (Arabic is written from right to left).

4. Diacritisation is necessary in the Lexical entry to identify the word's linguistic category, (e.g., "Transport نقْل " is Source where "Transferred نقَل " is Verb, hence 2 lexical entries)

## 5.3 System Modules

The prototype functionality is contained in the constituent module, functional module, semantic module and domain specific common sense module.

### 5.3.1 Constituent Structure Module

As mentioned in the assumptions earlier, the constituent structure is assumed to be available and correct. This module allows the end user to view the contents of the constituent structure in addition to simulating the conventional c-structure for the purpose of visualization and documentation.

The constituent structure object tree should be done manually and a naming convention has to be used to adhere to KappaPC's naming limitation and for the reader to follow up the decomposition process of the chosen text and production of the c structure. For example, subclass named S1_VP1 refers to the Verb Phrase number 1 at Sentence number 1. The Instance named S1_V1 refers to the first verb in the first sentence. This Instance has the slot value "عصف", which is placed in a slot named "Constituent", see figure 46. Figure 47 shows the constituent structure object tree of a full sentence.

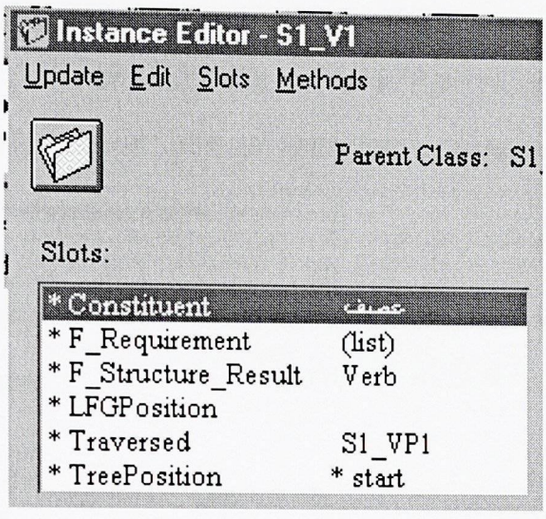


**Figure 46: c-structure content in Constituent slot**

### 5.3.2 Functional Structure Module

The f-structure module consists of a number of functions and methods. The main menu triggers the First function that identifies the sentence as being a Nominal or Verbal from the first word. The result of this is written as a functional role (e.g., Mubtada, Verb, etc) in the F_Structure_Result slot of that Instance. Figure 48 shows that the functional role of the word "وقـوع" is "Subject". This change triggers a KappaPC method that activates a KappaPC function that identifies the functional relationship for the current word with the next words in a recursive way. These relationships are described in the lexicon as a number of possible rules that may be partially or completely satisfied. This function repeats the same process until all the relationships of all the words in the hierarchy are processed.

**An Arabic Natural sentence:**

عصف وقوع حادثين كبيرين بسكون شارع سكني هادئ في غضون ساعتين من الاسبوع الماضي

**The transliteration:** asifa woqooa hadithain kabeerain be sokoon sharea sakani hadi fi ghodoon saatain min alosbooa almadhi

**The translation:** The occurrence of two big accidents has stormed a residential road within two hours of the last week.

```
S1_VP1
"   |___S1_V1----عصف"
"   |___S1_NP1"
"   |   |___S1_N1----وقوع"
"   |   |___S1_NP2"
"   |       |___S1_N2----حادثين"
"   |       |___S1_Adj2----كبيرين"
"   |___S1_PP1"
"   |   |___S1_P1----ب"
"   |   |___S1_NP3"
"   |       |___S1_N3----سكون"
"   |       |___S1_NP4"
"   |           |___S1_N4----شارع"
"   |           |___S1_Adj4----سكني"
"   |           |___S1_ConjP1"
"   |               |___S1_Conj1----و"
"   |               |___S1_Adj5----هادئ"
"   |___S1_PP2"
"       |___S1_P2----في"
"       |___S1_NP5"
"           |___S1_N5----غضون"
"           |___S1_NP6"
"               |___S1_N6----ساعتين"
"               |___S1_PP3"
"                   |___S1_P3----من"
"                   |___S1_NP7"
"                       |___S1_Det1----ال"
"                       |___S1_N7----اسبوع"
"                       |___S1_AdjP1"
"                           |___S1_Det2----ال"
"                           |___S1_Adj6----ماضي"
```

**Figure 47: Constituent structure object tree**

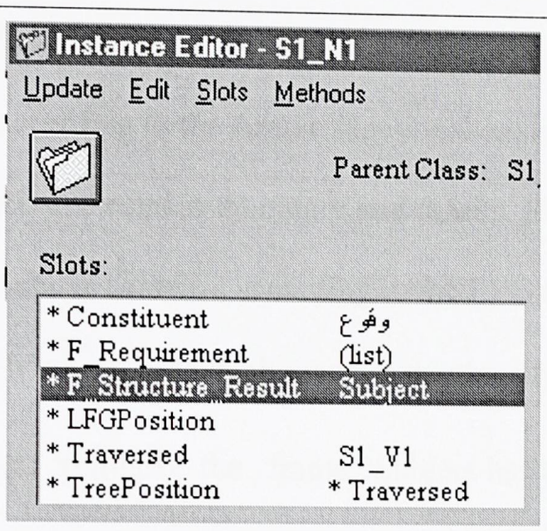


**Figure 48: f-structure result in F_Structure_Result slot**

If a word is a Particle or any word that makes a Semi Sentence (شبه جملة), the bypass function skips such a word looking for the next word that satisfies the f-structure rule while processing the functional requirements. This also triggers a method, which passes the requirements from the current word to the particle in order to find the suitable next word. For example, assume that the current word requires a LocationalObject that is satisfied by a word of a lexical superclass "location". If the next word is a Preposition then the required next word should remain of the lexical superclass "location" to be the Majroor for the Preposition. The Preposition here becomes the LocationalObject for the current word. This is an implementation of the notion that the particle's meaning is not complete unless followed by a meaningful word.

The Priority function contributes mainly in controlling the rule's priority, for example a word of the lexical class Pronoun can be both Ism and Khabar. Inna "ان", is Khabar if it is part of a subsentence (e.g., ان على هـ) and Ism if it is alone (e.g., ان هـ). Thus the Particle that comes between the Inna and the Pronoun, which is the Inna's Khabar in this case, has the priority to take the Pronoun as Majroor rather than Inna takes it as Ism.

This module also allows the end user to see the functional structure according to the LFG formalism and according to the Arabic functional analysis as well. It also allows the user to see the functional rules in summary and details. Figure 49 shows the output of this module for the above sentence according to the LFG format and shows that the first predicate is the Verb "عصف" which requires the three frames ^Subject, ^Object, and ^TemporalObject. Similarly the frame ^Subject has the predicate "وقوع" which requires the frame ^MudafElaih, and so on.

```
"Verb [Pred: عصف  (^Subject)  (^Object)  (^TemporalObject) "
"   Subject [Pred: وقوع  (^MudafElaih) "
"      MudafElaih [Pred: حادثين  (^Khabar) "
"         Khabar [Pred: كبيرين  "
"   Object [Pred: ب  (^Majroor) "
"      Majroor [Pred: سكون  (^MudafElaih) "
"         MudafElaih [Pred: شارع  (^Khabar)  (^Khabar2) "
"            Khabar [Pred: سكني  "
"            Khabar2 [Pred: و  (^Matoof) "
"               Matoof [Pred: هادئ  "
"   TemporalObject [Pred: في  (^Majroor) "
"      Majroor [Pred: غضون  (^MudafElaih) "
"         MudafElaih [Pred: ساعتين  (^MudafElaih) "
"            MudafElaih [Pred: من  (^Majroor) "
"               Majroor [Pred: ال  (^Determinee) "
"                  Determinee [Pred: اسبوع  (^Khabar) "
"                     Khabar [Pred: ال  (^Determinee) "
"                        Determinee [Pred: ماضي  "
"*** End S1_VP1 ***"
```

**Figure 49: Output of the Functional Structure**

The functional Structure is described in the general frame as shown in figure 50. The Pred (i.e. predicate) is the first word (i.e. Word-1, .., Word-n) in a phrase. The "^" symbol indicates that the following symbol (i.e., F-Requirement-1) is the functional roles required by the current word (i.e. Word-1, .., Word-n). The words that satisfy the required functional roles are in turn decomposed similarly.

```
f-structure =
  [Pred(Word-1) (^F-Requirement-1)... (^F-Requirement-n)
     F-Requirement-1    [Pred(Word-2) (^F-Requirement-1-1)... (^F-Requirement-1-n)
                        F-Requirement-1-1 [f-structure-1-1 ...]
                        F-Requirement-1-n [f-structure-1-n ...]
                  ...
     F-Requirement-n    [Pred(Word-n) (^F-Requirement-n-1)... (^F-Requirement-n-n)
                        F-Requirement-n-1 [f-structure-n-1 ...]
                        F-Requirement-n-n [f-structure-n-n ...]
```

**Figure 50**: **f-structure general frame**

### 5.3.3 Semantic Structure Module

The s-structure module consists of a number of functions and methods. The main menu triggers the main function that identifies and creates all the Instances and their Slot names and values. While doing this, it calls a second function, which tries to suffix the slot names with the particles and all suffixing words.

The module starts with the first word of a Sentence and tries to identify its Superclass in the s-structure. The Superclass could be an Action, Themes, Timings, or Locations. Actions and Themes are identified from the slot name "S_ID" in the lexical word entry, while Timings, and Locations are identified from the f-structure results (i.e., TemporalObject, or LocationalObject). If it is not one of the four Superclasses then it is a full slot name, partial slot name or a slot value. Once a word has been identified as an Instance belonging to any of the above four Superclasses, it is created in that hierarchy. This Instance should then have a slot of the name "InstanceName" and a slot value is the word itself. The slot name is suffixed with "ال" if this word's Functional role is Determinee. Then the slots for this Instance are created from the f-

structure requirements according to the s-structure rules. The Slot name is the f-structure requirement and the slot value is the Instance name of the word that satisfies this requirement. In case the word is an Adjective (e.g. "Red"), the slot name becomes the lexical class name of the word (e.g. "Colour"), and the word itself becomes the slot value. Some slot names will remain as the f-structure requirement until the k-structure is processed to change them to domain specific common sense names as applicable.

This module also allows the end user to see the contents of the semantic structure in addition to showing the semantic rules. The natural sentence shown in figure 47 is composed of eighteen words. Seven of those have been converted into instances such that three into Actions (i.e., عصف، وقوع، سكون), one into a Theme (i.e., حادثين), two into Timings (i.e., ساعتين، اسبوع), and one into a Location (i.e., شارع). Another seven words suffixed slot names (i.e, ب، و، في، غضون، من، ال، ال). The last four became slot values (i.e., كبيرين، ماضي، سكني، هادئ). The first Instance "S_S1_V1" is classified in the Action superclass and has four slots. The first slot is "InstanceName" and its value is "عصف". This shows the identity of the Instance. The second slot is "Subject" which has the value "S_S1_N1". This is a pointer to the second Instance "وقوع" which simulates the relationship between the four Instances. The third slot is "Object_ب" having the value "S_S1_N3" which points to the third instance "سكون". The last slot is ""having the value "S_S1_N6" which points to the fifth instance "ساعتين". See figure 51.

```
"InstanceName ----------------  is: عصف"
"Subject ---------------------  is: S_S1_N1"
"Object_ب --------------------  is: S_S1_N3"
"TemporalObject_في_غضون ------- is: S_S1_N6"
"***=== Actions S_S1_V1===***"

"InstanceName ----------------  is: وقوع"
"MudafElaih ------------------  is: S_S1_N2"
"***=== Actions S_S1_N1===***"

"InstanceName_ب --------------  is: سكون"
"MudafElaih ------------------  is: S_S1_N4"
"***=== Actions S_S1_N3===***"

"InstanceName ----------------  is: حادثين"
"Size ------------------------  is: كبيرين"
"***=== Themes S_S1_N2===***"

"InstanceName ----------------  is: ساعتين"
"MudafElaih_من_ال ------------  is: S_S1_N7"
"***=== Timings S_S1_N6===***"

"InstanceName_من_ال ----------  is: اسبوع"
"Tense_ال --------------------  is: ماضي"
"***=== Timings S_S1_N7===***"

"InstanceName ----------------  is: شارع"
"Function --------------------  is: سكني"
"Noise_و ---------------------  is: هادئ"
"***=== Locations S_S1_N4===***"
```

**Figure 51: s-structure Output**

The Semantic Structure hierarchy is summarized in the general frame as shown in figure 52. It decomposes into four superclasses, which are the Themes, Actions, Timings, and Locations. These superclasses can have a number of relevant Instances that can have the c-structure names prefixed with the letter "S". These Instances would contain a number of relevant slot names generated based on the functional role, Lexical category of an Adjective, a current slot name suffixed by a particle, or a Circumstance word that is not preceded by a determinant particle. These slot names have pairs of slot values of Instance names as pointers, or actual words of the lexical

categories such as Noun, Verb, Adjective, or a Circumstance word that is preceded by a determinant particle.

```
Structure [S-structure]
    SubClass [Themes][Actions][Timings][Locations]
        Instance [S+c-structure object Name]
            Slot Name {[Functional Role] [Adjective SuperClass] [Slot Name + Particles]
                      [Non-Determinant Circumstances]}
            Slot Value {[Instance][Noun][Verb][Adjective][Determinant Circumstances]}
```

**Figure 52: s-structure general rule**

### 5.3.4 Common Sense Structure Module

The k-structure module is a set of KappaPC functions that execute a number of KappaPC rules. All rules are extracted from domain specific common sense and presented in If-Then statements. These rules are then invoked through a number of forward chaining functions. This process runs through a number of restructuring iterations, see figure 53.

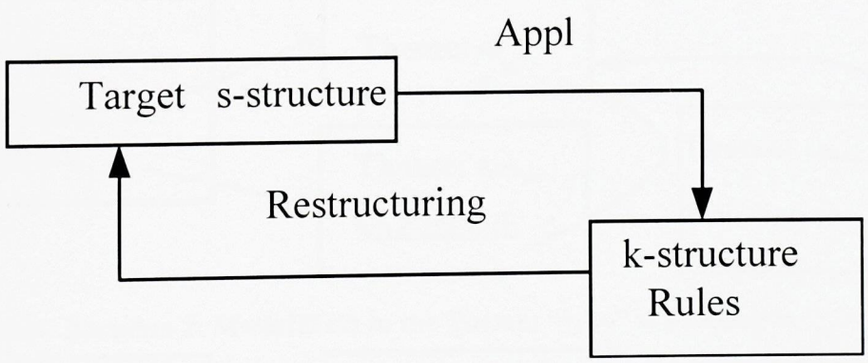


**Figure 53: Restructuring the s-structure**

The iterations could start with completing the relationships, renaming the proper Slot names, and then replacing the Slot values, and may end in deleting some Instances. For example, the sentence "ساق علـي سـيارة ـــه Ali drove his car" passes through four

iterations, see figures 54a to 54e. The S-structure result shows the Action "ساق Drove" pointing to "علي Ali" as the Subject and to "سيارة Car" as Object. The Pronoun "ـــه his" is the MudafElaih for "ســيارة Car". After the first iteration the pronoun is pointing to "علــي Ali", After the second iteration "علي Ali" becomes the MudafElaih for "سيارة Car", after the third iteration the Slot name "MudafElaih" becomes "Owner", the Pronoun is deleted after the last iteration.


**Figure 54a: Iteration 0: s-structure Result**

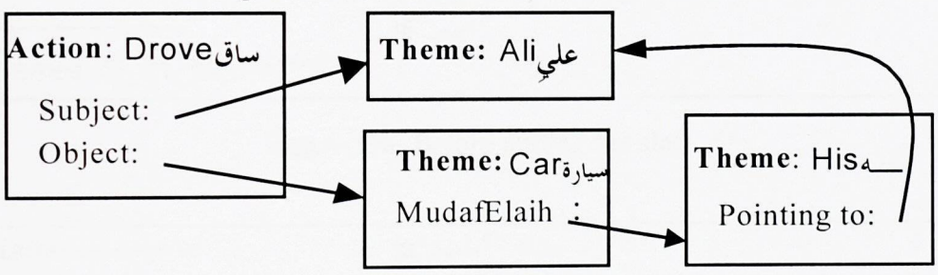


**Figure 54b: Iteration 1: Theme: "ـــه" is pointing to "علي Ali"**

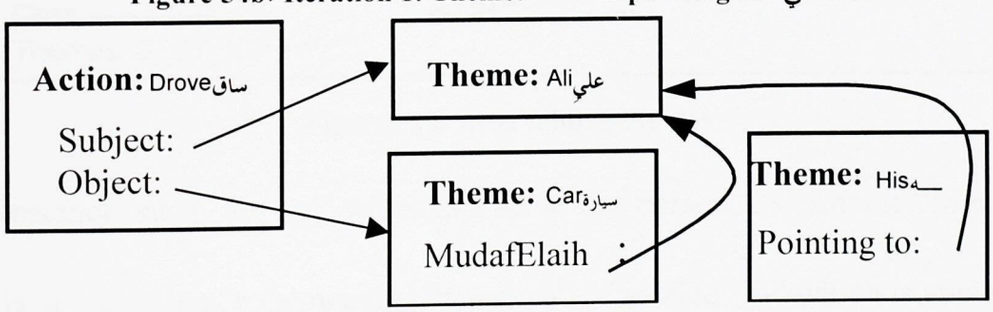


**Figure 54c: Iteration 2: MudafElaih in the Theme: "سيارة" is pointing to "علي Ali"**

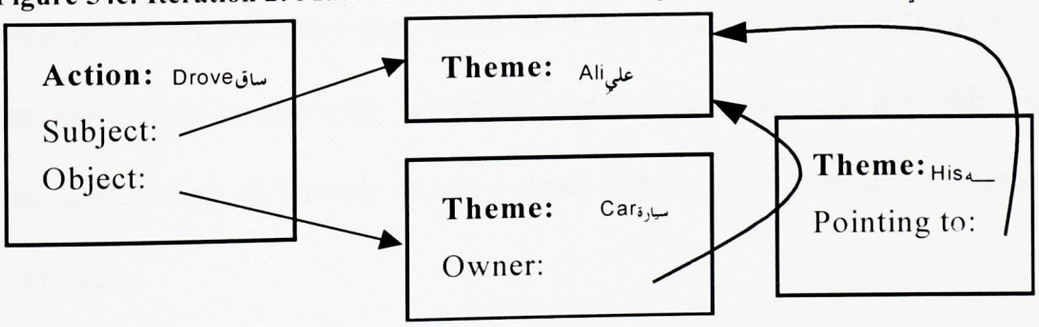


**Figure 54d: Iteration 3: MudafElaih in the Theme: "سيارة" is changed to "Owner"**

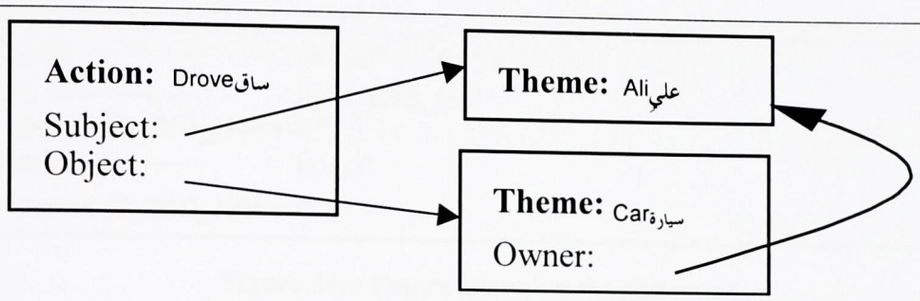


**Figure 54e: Iteration 4: Theme: "ـــه" is deleted**

The result of the k-structure is an enriched version of the s-structure. This result can be viewed using the s-structure module. For example in an Instance such as "حادثين" which means "Two Accidents", a slot name "Size" of a value "كبيرين" which means "Big" indicates that the number of cars involved is 3 or more. Therefore an additional slot of the name "**No_of_Cars**" is added and its value is "**3+**" which means 3 or more cars, see figures 55a and 55b.

```
"InstanceName ----------          is: حادثين"
"Size ------------------------    is: كبيرين"
"***=== Themes S_S1_N2===***"
```

**Figure 55a: Before adding the slot**

```
"InstanceName ----------          is: حادثين"
"Size ------------------------    is: كبيرين"
"No_of_Cars ----------            is: 3+"
"***=== Themes S_S1_N2===***"
```

**Figure 55b: After adding the slot**

In an Instance such as "انتوني" which is a name of a person, a slot name "MudafElaih" which is a functional role meaning "annexed" of a value "هيل" which is another name of a person, which indicates that the second person is the father of the first. That is why the slot name is changed to "Fathername" , see figures 56a and 56b. This rule could be understood differently in a non-Arab cultures where the second name could be the surname.

```
"InstanceName ----------          is: انتوني"
" MudafElaih ----------           is: S_S10_N5"
"***=== Themes S_S10_N4===***"
"InstanceName ----------          is: هيل"
"***=== Themes S_S10_N5===***"
```

**Figure 56a: Before changing the slot name**

```
"InstanceName ----------          is: انتوني"
" FatherName ----------           is: S_S10_N5"
"***=== Themes S_S10_N4===***"
"InstanceName ----------          is: هيل"
"***=== Themes S_S10_N5===***"
```

**Figure 56b: After changing the slot name**

The domain specific common sense knowledge rules are expressed in terms of the general rule as shown in figure 57. The *Condition* is the status of any Object in the domain indicating whether it exists or possesses a certain value or relationship to another object. The *Update* in the Result is creating new Instance or Slot Name, deleting existing ones, renaming them or changing their values.

```
If  {[Condition (Thematic Role-1)]... [Condition (Thematic Role-n)]} OR
    {[Condition(Slot Name-1)]... [Condition(Slot Name-n)]} OR
        {[Condition(Slot Value-1)]... [Condition(Slot Value -n)]}
Then
        {[Update(Instance)] [Update(Slot Name)] [Update(Slot Value)]}
```

**Figure 57: k-structure general rule**

### 5.3.5 Lexicon Module

KappaPC features are used to enter the lexical entries grouped according to their linguistic categories. Moreover, the functional and semantic rules are incorporated into these lexical entries.

The functional structure rules are simulated in two phases, the Lexical and procedural phases. In the lexical phase a number of slot names and values containing the

functional properties of each rule are applied to the hosting lexical entry. For example the Verb "اصطدم" is a lexical entry of the category Transitive which belongs to the linguistic class known as Verb. This verb should have a Subject that should be of the category " Tool" such as "سيارة". Therefore a slot is created inside the lexical entry "اصطدم" with the name "Subject" and a value "Tool" which means that any lexical entry of the category Tool is a good candidate for being a Subject. See figures 58 and 59.

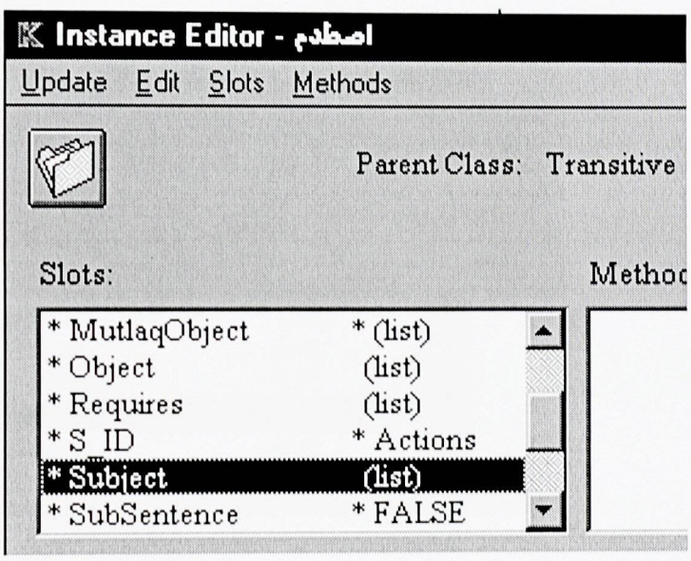


**Figure 58: f-structure rules are simulated in Slot names**

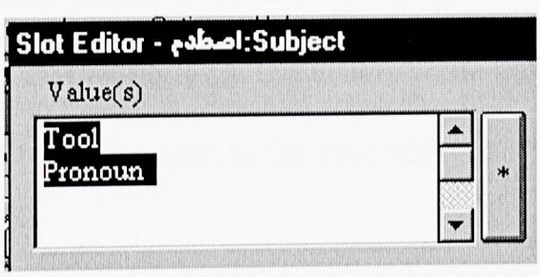


**Figure 59: f-structure rules are simulated in Slot values**

During the procedural phase, a number of IF-Then statements are embedded in the KappaPC procedure that will refer to these slots while traversing the c-structure tree in order to identify the functional roles such as this Subject.

The semantic structure rules are simulated in two phases, the Lexical and procedural phases. In the lexical phase a number of slot names and values containing the semantic properties of each rule applied to the hosting lexical entry. For example the Verb "اصطدم" is a lexical entry that should be an Instance in the Actions group in the s-structure. Therefore a slot name called "S_ID" is created inside that lexical entry having the value "Actions" which means that this word is an Instance of the Actions subclass. See figures 60.

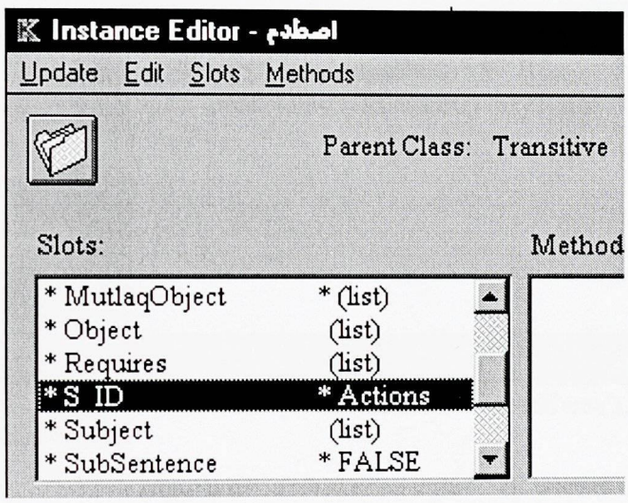


**Figure 60: s-structure rules are simulated in Slot names and values**

During the procedural phase, a number of IF-Then statements are embedded in the KappaPC functions that will refer to these slots while traversing the f-structure tree in order to identify the Instances such as the case with "اصطدم" above.

The domain specific common sense knowledge structure rules are not represented in the Lexicon. The KappaPC inference engine was used to represent them, as it is powerful enough to produce the proposed k-structure. Figure 61 shows the representation of the FatherName k rule in the KappaPC inference engine. In this representation, "t|Themes" in the Patterns: section means that for all Instances "t" in the Themes subclasses of the s-structure. In the "If:" section, there are a number of conditions that a) look if there exists a slot with a name "MudafElaih" in an Instance

"t", b) the second condition is that the category of the found Instance "t" should be "Human", c) the third condition is that the category of the slot value of that slot name should also be "Human". The conclusion of this rule should be a replacement of the slot name from "MudafElaih" to "FatherName".

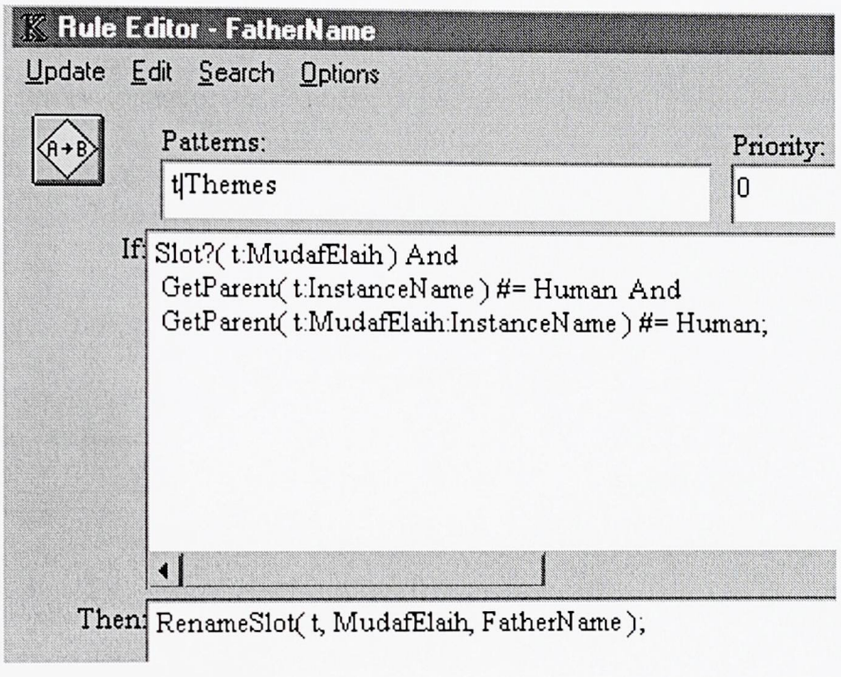


**Figure 61: Inference engine represents k rules**

This engine was not suitable for processing the f-structure and the s-structure because the inference engine processes identified Instances in identified Subclass while the Instances and Subclasses are variables in the f-structure and the s-structure.

The Lexical entries are implemented in four main categories, the Noun, Adjective, Verb and Particles. Each category is decomposed into a number of subcategories such that the Noun is decomposed into Circumstantial, Pointer, Pronoun, Proper, etc. See Figure 62.

```
Lexicon
"   |___Noun"
"   |   |___Circumstancial"
"   |   |   |___KnownTemporal"
"   |   |   |___Location"
"   |   |___Pointer"
"   |   |___Pronoun"
"   |   |___Proper"
"   |   |   |___Animal"
"   |   |   |___Human"
"   |   |   |___Tool"
"   |   |___Source"
"   |   |   |___SourceVerb"
"   |   |   |___SourceMarrah"
"   |   |___Jens"
"   |   |___Mawsool"
"   |___Adjective"
"   |   |___Size"
"   |   |___Noise"
"   |   |___Function"
"   |___Particle"
"   |   |___Tawkeed"
"   |   |___Proposition"
"   |   |___Determinant"
"   |___Verb"
"       |___Transitive"
" "
```

**Figure 62: Lexical Categories**

## 5.4 Implementation of Rules

The functional, semantic and domain specific common sense designed rules are implemented in KappaPC in different forms. Some of the implemented rules are described below.

### 5.4.1 Implementation of Functional Rules

Each of the functional rules is defined in the relevant lexical entry.

The rule "^Subject" in:-

"(^Subject) On: عصف Of Type: Transitive, Is Satisfied by: [SourceVerb, ] "

is applicable to the Lexical Entry "عصف" (i.e., Asifa "Stormed") which is of the class Transitive. The candidate word that satisfies this rule should be of the class SourceVerb and follows "عصف" in the c-structure hierarchy.

The rule "^Object" in:-

"(^Object) On: عصف Of Type: Transitive, Is Satisfied by: [ب, SourceVerb, ]"

is applicable to the Lexical Entry "عصف". The candidate phrase that satisfies this rule should be a prepositional phrase of the proposition "ب" and another word of the class SourceVerb and follows "عصف" in the c-structure hierarchy.

The rule "^TemporalObject" in:-

"(^TemporalObject) On: عصف Of Type: Transitive, Is Satisfied by: [في, UnKnownTemporal, ]"

is applicable to the Lexical Entry "عصف". The candidate phrase that satisfies this rule should be a prepositional phrase of the proposition "في" and another word of the class UnKnownTemporal and follows "عصف" in the c-structure hierarchy.

### 5.4.2 Implementation of Semantic Rules

The s-structure rules are defined as slot names and values in the lexical entries, which are complemented with a number of If-Then statements in the KappaPC functions.

The rule:-

"Actions Is applicable to all Instances of the Subclass: Kana&Sisters"

suggests that any lexical entry defined in the Kana&Sisters superclass should result into creating an Instance in the Actions subclass of the s-structure. This rule is applicable on the lexical entry "كان" of the superclass "Kana&Sisters" when producing the s-structure.

The rule:-

"Locations Is applicable to all Instances of the Subclass: Location"

suggests that any lexical entry defined in the Location superclass should result in the creation of an Instance in the Locations subclass of the s-structure. This rule is applicable on the lexical entry "مستشفى" of the superclass "Location" when producing the s-structure.

The rule:-

"SlotNameSfx Is applicable to all Instances of the Subclass: Conjunction"

suggests that any lexical entry defined in the Conjunction superclass should result in suffixing the slot name with that entry in the s-structure. This rule is applicable to the lexical entry "و" of the superclass "Conjunction" when producing the s-structure.

The rule:-

"SlotNameValue Is applicable to all Instances of the Subclass: Density"

suggests that any lexical entry defined in the Density superclass should result in the creation of a slot with the name "Density" and the value is that entry in the s-structure. This rule is applicable to the lexical entry "مركزة" of the superclass "Density" when producing the s-structure.

The rule:-

"Themes Is applicable to all Instances of the Subclass: Human "

suggests that any lexical entry defined in the Human superclass should result in the creation of an Instance in the Themes subclass of the s-structure. This rule is applicable to the lexical entry "انثوي" of the superclass "Human" when producing the s-structure.

The rule:-

"Timings Is applicable to all Instances of the Subclass: KnownTemporal"

suggests that any lexical entry defined in the KnownTemporal superclass should result in the creation of an Instance in the Timings subclass of the s-structure. This rule is applicable to the lexical entry "ساعتين" of the superclass "KnownTemporal" when producing the s-structure.

### 5.4.3 Implementation of Common Sense Rules

Some of the k-structure rules described earlier are implemented automatically by having an explicit positive Instance mentioned in the text as in the case of the Accidents Occurs. Some of the rules are left for future work as in the case of Accident Causes where it is beyond the scope of this research.

The phrase "وقوع حادثين", which means "the Occurrence of two Accidents", resulted in the creation of two Instances in the Object Model, a positive Action "وقوع" and a Theme "حادثين". This implies that an accident did occur, which is sufficient to answer "Yes" to a future query such as "هل وقع حادث؟ Did an Accident Occur?"

The involvement in an accident has been implemented via the functional relationships between the Actions and the Themes. For example, in the phrase "اصطدمت سيارتان two cars collided", "سيارتان two cars" is involved in an accident because their Instance in the Themes are pointed to by the slot "Subject" from the Instance "اصطدمت collided" in the Actions.

The adjective of a person could explicitly be mentioned as in "سائق السيارة انتوني The Car Driver Antony". This implies that the named person is the driver. In an implicit phrase such as "انتوني جالس خلف عجلة القيادة وقت الحادث Antony was sitting behind the steering at the time of the accident", Antony should be recognized as the driver if the Locational_Object =

"خلف عجلة القيادة behind the steering wheel" and the Temporal_Object ="وقت الحادث, at the time of the accident" for the Action " جالس was sitting ".

## 5.5 Conclusion

This chapter showed how a parsed constituent structure is used as an input to the functional structure module that produces the functional structure. The semantic structure module takes the functional structure as input and produces the semantic structure. The domain specific common sense module enhances the semantic structure through a number of iterations. The lexicon declares the words and groups them according to their linguistic categories and incorporates the functional and semantic features that are necessary for the rules. KappaPC provided the necessary features that made it possible to make the constituent, functional and semantic structures in terms of object trees. Moreover, KappaPC is a powerful representation environment for objects, frames, rules and various inference mechanisms.

# Chapter Six

# The System Testing and Evaluation

## 6.1 Introduction

The input and output of each module is produced independently and can be tested and evaluated in separate stages. In other words, the end user has the opportunity to run each module alone which allows him/her to trace any errors in the output of the preceding module before it is carried forward to the succeeding one.

Some figures are presented in this chapter for each structure including the lexicon. Such statistics reflects the percentage of success of each structure compared with the initial objectives set for each module.

## 6.2 The Evaluation Criteria

### 6.2.1 Generality

It is the degree of successful representation of each structure in the framework. For example, the constituent structure is 100% general if it can represent all the natural language sentences. Similarly, the functional structure is 100% general if it can represent all the constituent structure. The case is same for the semantic and the domain specific common sense structures.

### 6.2.2 Selectivity

It is the degree of non-sentences that each structure in the framework can identify as problematic. This applies mainly on the constituent structure as the natural language sentence could be entered in a wrong grammar format, while the rest of the structures receive filtered constituent structure.

### 6.2.3 Understandability

It is the degree of clarity and simplicity of each structure in the framework. For example, the understandability of the constituent structure is 100% if the user can see all the contents of the structure and can relate all the tokens of the structure to all the words of the natural language words. Similarly, the contents of the functional, semantic, and common sense structures should all be seen and related to the previous structures in order for them to be 100% understandable.

## 6.3 Evaluation of the Constituent Structure

The output of the constituent structure was designed and entered in a KappaPC object tree manually. It fully matches the functional input requirements after implementing the constituent structure assumption.

A story in the traffic accident domain was selected as a working example for the research. This story contained eight compound sentences originally. After implementing the constituent structure assumptions such as the normalization, they became 29 parsed sentences.

More over, the original text contained 329 words, which after creating the c-structure have become 430 words. This is due to the fact that the number of words added in

separating the affixes (e.g., ال، ب، ـــــه، ...) from the original words is greater than the number of deletions of the joining words (e.g., عم).

The c-structure is 100% general as it managed to represent all the correct words in the natural Arabic text given.

The selectivity is applied to the c-structure as all words mentioned in the assumptions are excluded.

The understandability is 100% as all contents can be viewed online and all constituents can be related to the original text.

## 6.4 Evaluation of the Functional Structure

The functional structure is 100% general as all 29 constituent structure sentences have been converted into functional structure according to both the LFG framework and the Arabic grammatical classic representation. Twenty generic functional rules were generated to process the 29 sentences. These rules are represented in the Lexicon for the categories and inherited by all the lexical entries, which makes the number of generic rules equal to the number of the words mentioned above.

The selectivity does not apply to the functional structure as it receives a filtered constituent structure, which has no problems.

The functional structure is 100% understandable as all of its contents can be viewed online and its tokens can be related to the constituent structure.

Initially during the analysis and design phase, the functional structure was presumed to be in an independent object tree. Later on during the implementation phase, it was

found more convenient to add the functional roles and relationship pointers as attributes to each constituent of the constituent structure. The advantage of this is that both the constituent structure and the functional structure are then represented in the same object tree.

## 6.5 Evaluation of the Semantic Structure

The semantic structure is 100% general as all of the 430 words mentioned above and processed functionally have been converted into 247 Instances in a semantic structure object tree with four superclasses. These Instances contained 475 Slots amongst which 209 are relationships. The number of generic semantic rules identified for this structure is 6, which are represented in the Lexicon as slots in each lexical entry. These rules produced Instances, slot names and slot values explicitly.

The selectivity is not applicable here because the functional structure is generated automatically and is not problematic.

The semantic structure is 100% understandable as all of its contents can be viewed online and its objects can be related to the functional structure and the lexicon.

Many slot names and slot values need to be refined implicitly making domain specific common sense knowledge an important resource for deriving additional information that is necessary to restructure the semantic structure.

## 6.6 Evaluation of the Common Sense Structure

The domain specific common sense knowledge structure required the most exhaustive rules in order to refine all the semantic structure. A sample of 26 rules has been identified on the traffic accident domain. Nine of these rules have been implemented

which showed encouraging results. Some of the implemented rules renamed the slot names, some changed the slot values, some created new objects and some deleted redundant objects.

The execution of domain specific common sense rules could be triggered after the production of the semantic structure or prior to the process of answering a request from the end user. The first option is done in batch mode, which enhances user transaction response time, but on the other hand it may process more than what the user needs, unnecessarily increase the population of the database. The second option processes the desired information, but the user response time is longer. It is found that the first option is better at least for the current prototype version.

The generality of the k-structure is different for each rule depending on the domain. For example the ownership rule (i.e. Ahmed's Car) is 100% general for all domains (i.e. Ahmed's Shirt, Ahmed's Story, etc). On the other hand, the Driver's rule (i.e. setting behind the steering wheel) is applicable to the traffic domain only.

The selectivity is not applicable here because the semantic structure is generated automatically and is not problematic.

The common sense structure is 100% understandable as all of its contents can be viewed online and its objects can be related to the common sense rules and the lexicon.

## 6.7 Evaluation of the Lexicon

The Lexicon contained all the parsed words as Instances grouped in 21 subclasses that are further grouped in four main superclasses that consist of Noun, Verb, Adjective

and Particle. These Instances are grouped and inserted manually (A process which has already bean investigated and systems are already been implemented for it) having their slots to describe the functional and semantic properties discarding the constituent and morphological properties. These properties helped in generating the functional structure and the semantic structure efficiently.

Diacritization of the words in the Lexicon proved to be an efficient solution to ambiguities in the categorisation. This is because logically it is not the responsibility of a parser or semantic engine to categorize words, this is fairly the responsibility of the linguists.

The domain specific common sense knowledge rules were not described in the lexicon object tree because they were described as inference rules utilizing the kappaPC's inference engine functionality.

## 6.8 Comparison to Similar Systems

The system described in this work is better than the Computer Based System for Understanding Arabic Language CBSUAL, Xerox Morphological Analyzer (XMA) and the Arabic-To-English Machine Translator (ATEMT) in a number of points.

The CBSUAL system is implemented on a specially formatted Arabic text to solve exercises in Mechanics for school students after translating it into French. The work of this thesis on the other hand is implemented on a full natural Arabic text composed of 29 sentences in a traffic accident domain.

Both the XMA and ATEMT systems implement only the constituent and the functional structures of LFG theory. The work of this thesis on the other hand

implements the semantic and the common sense knowledge structures on the top of the previous structures.

## 6.9 Conclusion

The system described in this work is better than similar system in that it is implemented on a natural Arabic language text rather than on a specific formatted text. More over, it implements the semantic and the common sense knowledge structures rather than just the constituent and functional structures.

The constituent structure assumptions described earlier pose some limitations that have to be rectified in the future. For example the special characters such as the parenthesis "(" or ")" should have a defined meaning that is processed in the C, F, S, and K structures or otherwise deleted from the original text.

Combining the functional and semantic structure in one object tree is found to be more convenient and efficient. The user interface is managed to extract the constituent structure and the functional structure while traversing the same object tree.

The semantic structure is the first level of capturing the meaning from the text. All words are converted either to instances, slot names or slot values. Nearly half of the parsed words became instances, while the number of parsed words is almost equal to the number of the created slots, half of which are simulating the relationships among the instances.

The implementation of domain specific common sense rules represents the second step towards capturing the meaning embedded in the natural text. It appeared after

implementation that domain specific common sense structure is a process of restructuring the semantic structure by adding, modifying or deleting structure objects.

The lexicon was designed and inserted manually having into consideration some restrictions such as excluding the morphological analysis. However, it provided an efficient input method to generate the functional structure and the semantic structure as well.

# Chapter Seven

# Conclusions and Future work

## 7.1 Conclusions

The work presented in this thesis represents an approach towards computer understanding of a complete story written in natural Arabic language. This approach is based on the lexical-functional grammar theory, which involves the four structures in processing: the constituent structure, the functional structure, the semantic structure and domain specific common sense structure. This approach was automated through developing and implementing a prototype using KappaPC Version 2.3 on MS Windows 95 with support for Arabic on an IBM compatible PC 486 platform. The prototype showed encouraging results because it processed the three related phases: syntax, semantics, and pragmatics. The prototype managed to represent the meaning in such a way that it is suitable for future use such as interrogations or machine translation.

The natural Arabic sentences are manually parsed and inserted into a constituent structure object tree. This object tree is a hierarchy of subclasses that represent the

constituent rules such as Noun Phrase, Verb Phrase and Prepositional Phrase. The Instances of this object tree accommodate the natural words.

The Arabic functional structure is produced successfully from the constituent structure according to the lexical functional grammar theory. The functional role of each word is identified and placed in the constituent structure tree. The functional relationship is another component that is identified and stored with the functional role. This process is performed for each constituent in order to maintain the functional completeness of a sentence. The designed functional properties in the lexicon participated in resolving the free word order ambiguities.

A technique was developed from the thematic roles frame representation to produce the semantic structure from the functional structure automatically and successfully. Four thematic roles were used, which are the Actions, Themes, Locations and Timings. The semantic structure is produced in a separate object tree.

The domain specific common sense knowledge structure was implemented successfully, which contributed to understanding of some of the Arabic eloquence. It became clear that the domain specific common sense structure is a set of rules extracted from specific domains and cultures and serves to enhance the semantic structure.

In the lexicon, the words were grouped according to their linguistic categories and were described in terms of their functional and semantic properties. Diacritisation that distinguishes the word category (e.g., Verb, Source, etc.), is a very critical issue in written Arabic Language from both the user and the application perspectives. The user would find it more practical to write Arabic without diacritisation, for example the

user would write "تمر" to mean "dates" (i.e., noun) in one place and to mean "pass" in another. This causes a number of ambiguities to the application as it should distinguish between the two words above. The word is of a class "Verb" in the first location and "Source" in the second. In this case the Lexicon should be designed to accommodate the words along with their diacritisation, while allowing the user to enter them not diacritised in the input sentence so the sentence is processed twice, once dealing with the word as a Verb and another as a Source. The application should be able to decide whether both or one of the two interpretations is correct. The Lexicon should not consider the diacritisation that distinguishes the functional role (e.g., Subject, Object, etc.), as this is the responsibility of the functional structure module.

The work presented here covers a major portion of the Arabic Grammar rules that are captured from the 29 sentences, see appendix B. These sentences were selected from a traffic domain example extracted from a newspaper story.

This work has laid down the foundation to be used in a number of industrial applications such as query answering systems and machine translation. For example, in the Arabic-To-English Machine Translation (ATEMT) system [Shih-98] the author mentioned that the preposition phrase is not handled properly in some cases. For example, the preposition "مـــن" which means *from* in the source sentence "حقه من العناية" should be excluded during the transformation process to have the target sentence *his rightful attention* without the preposition *from.* This is believed to be because the ATEMT system transforms the functional structure of the Arabic sentence to the English functional structure directly. Therefore, this problem could be resolved after

processing the semantic and domain specific common sense structure in which redundant information is deleted and new information is generated.

## 7.2 Future work

Integrating this prototype in a parser that generates the constituent structure would result in automating the entire system for natural Arabic text Understanding.

All functional rules should be identified and implemented to cover the complete Arabic language grammar.

More semantic rules that cover many different domains should be investigated. The possibility of integrating the different domains towards generating a comprehensive semantic structure needs to be evaluated.

Exhaustive identification and implementation of domain specific common sense rules is necessary to complement the semantic structure. The domain specific common sense rules in different domains and cultures give better results and more completeness.

The other entities in the traffic domain model that were developed in this work such as the cause of accidents, corrective actions, preventive actions, monitoring process and learning all need further consideration. The accidents are due to a number of causes that are either fed to the system or produced as a result of a learning process. One way is to classify the verbs in the lexicon as candidates for corrective actions. The key attribute for the verb to be a corrective action is that when its anticipated results are the restoration of the original status to the end users affected by the previous damaging verb. For example the verb "تعالج treats" is a verb that restores the health that

was damaged by a previous verb such as "جرح wounded". The verb "اصطدم collided" for example is not a corrective action. The Preventive Action requires looking for facts as reasons for accidents and then using these to generate some advice to answer the relevant future queries.

The Monitoring process is a collection of reporting activities or devices. For example, a police patrol operates in the main roads watching and reporting on the current status. Another example is installing a number of cameras in the main spots to transmit the traffic status on those spots. This information streamed from the monitoring process provides valuable opportunities to take the preventive and corrective actions.

Learning from experience within the traffic domain is a very important educational process that involves extracting rules from repeated traffic related facts. For example from processing the records in the accident reports, the system will deduce the causes of accidents and the preventive actions in addition to providing an input for the monitoring process.

## References

[Abus-85] Ahmed Abusaeed and Hosain Sharara, "Daleel AlEmla wa AlEarab", Dar AlElm Lel Malayeen, 1985.

[Alaa-94] Al-A'ali M; Girgis M., "Arabization: Actuals & Objectives", The proceedings of the 4th International Conference and Exhibition on Multi-lingual Computing, London, UK, April 1994, Pages 9.1.1-9.1.10.

[Alai-96] Alain Marcoux, " Population Change-Natural Resources-Environment Linkages in the Arab States Region", http://www.undp.org/popin/fao/arabstat.htm, FAO Population Programme Service, April 1996.

[Alfr-72] Alfred V. Aho and Jeffrey D. Ullman, "The Theory of Parsing, Translation, and Compiling", Volume I, Prentice Hall, 1972..

[AlHa-80] Bahaeldeen AlAqueeli AlHamadani, "Sharh Ibn Aqueel, V1", Dar AlQalam, 1980.

[Ali-94] Nabil Ali,, "The Arabs and the Information Age العرب وعصر المعلومات", Aalam Al Maarifa 184, April 1994.

[Alja-88] Ali Aljarem; Mustafa Ameen, "Alnahw Alwadheh",Dar Alfordoos, 1988.

[Anto-94] Antoine Dahdah, A Dictionary of Arabic Grammar in Charts and Tables, Library of Libanan, 1981.

[Bees-98] K. R. Beesley, "Arabic Morphological Analysis on the Internet", the proceedings of the 6th International Conference and Exhibition on Multi-lingual Computing, University of Cambridge, London, 17-18 April, 1998, pages 3.1.1 - 3.1.10.

[Bolc-96] Leonard Bolc and others, "A Survey of Systems for Implementing HPSG Grammars", Technical Report 814, IPI PAN (Institute of Computer Science, Polish Academy of Sciences), http://www.sfs.nphil.uni-tuebingen.de/~adamp/Papers/1996-survey/ , October 31, 1996.

[Carl-94] Pollard, Carl and Ivan A. Sag. 1994. Head-Driven Phrase Structure Grammar.Chicago: University of Chicago Press and Stanford: CSLI Publications.

[Chan-87] Li-Li Chang; Juei-Chu Huang; et.al., "Classifications and co-occurrence restrictions in Chinese simple noun phrase", Proceedings of International conference on Chinese and oriented language computing, pp 107-110.

[Covi-94] Michael A. Covington, Natural Language Processing for Prolog Programmers, Prentice Hall, 1994.

[Davis -96] Tony Davis, "Chapter 4 HPSG: a brief description and some revisions", http://www-csli.stanford.edu/~tdavis/thesis-ps.html , July 7, 1996.

[Elma-94] Elmasri and Navathe, Fundamentals of Database, Addisson-Wesley, Second Edition, 1994.

[Elna-89] Ayman El-Naggar, "A finite state automata of the Arabic grammar, IEEE international workshop on tools for Artificial Intelligence. Architectures, Languages and Algorithms, 1989, p. 693-9.

[Fedd-93] Feddag A., & Foxley E., A Lexical Analyser for Arabic, International Journal of Man-Machine Studies, Vol. 38, iss: 2, pp. 313-330.

[Fertuck-92] Len Fertuk, Systems Analysis and Design with Case Tools, Wm. C. Brown Publishers, 1992.

[Frie-69] Friedman J., "A Computer System for Transformational Grammar", SMART Collection: cacm-1885, http://paris.lcs.mit.edu/~bvelez/std-colls/cacm/cacm-1885.html , CACM June, 1969.

[Gazd-90] Gerald Gazdar & Chris Mellish, Natural Language Processing in Prolog, Addison wesly, 1990.

[Ghei-89] Dr. Mervat Gheith; Magdy Aboul-Ela, "A computer based System for Understanding Arabic Language", Selected proceedings of the workshop on Computer Processing of Arabic Language, Dar A-razi, Kuwait, 1989, 14-16 April, pp 219-229.

[Green-98] Georgia M. Green, "Fundamentals of HPSG", http://lees.cogsci.uiuc.edu:80/~green/, July 7, 1998.

[Jame-87] James Allen, Natural Language Processing, Benjamin/Cumming Publishing Company, 1987.

[John-76] John McCarthy, "An Example for Natural Language Understanding and the AI Problems it Raises", http://www-formal.stanford.edu/jmc/mrhug/mrhug.html, 1976.

[Jong-98] Jong-Yul Cha, Implementations, http://cogsci.uiuc.edu/~jycha/, HEYUM's Linguistics Page, 1998.

[Kapl-82] Ronald M. Kaplan, and Joan Bresnan,`Lexical-Functional Grammar: A formal system for grammatical representation. In Joan resnan, editor, The Mental Representation of Grammatical Relations, pages 173-281. The MIT Press, Cambridge, MA, 1982.

[Kapl-89] Kaplan, Ronald M. `The formal architecture of Lexical-Functional Grammar'. In Chu-Ren Huang and Keh-Jiann Chen, editors, Proceedings of ROCLING II, pages 1-18, 1989. Also in Journal of Information Science and Engineering, vol. 5 no. 4, pp. 305-322, 1989.

[Kapl-96] Ron Kaplan and John Maxwell, Grammar Writer's Workbench for Lexical Functional Grammar, http://www.parc.xerox.com/istl/groups/nltt/medley/, 1996.

[Ling-98] Linguistic Theories, Lexical Functional Grammar, Linguistic Resources, Language and Linguistics, http://www.systranmt.com/~jyang/ling.html, 1998.

[Mira-96] Mirayati M; Al Tian M, "Arabisation and the Computer التعريب والحاسوب",Symposium of Computer Arabisation, Syria, 9-11 December 1994. Pp 1-32.

[Nara-94] Naraayanan A., & Hesham L., Finite-state abstraction on Arabic morphology, Artificial Intelligence Review, vol. 7, iss: 6, 1993-1994.

[Noam-98] Noam Chomsky, " Theory of Grammar: ", http://www.science.mcmaster.ca/Psychology/psych2h03/language/tsld006.htm, 1998.

[Robe-98] Robert BAUD, Christian LOVIS, Laurence ALPAY, Anne-Marie RASSINOUX, Jean-Raoul SCHERRER, Anthony NOWLAN, Alan RECTOR, "Modeling for Natural Language Understanding", http://mbi.dkfz-heidelberg.de/helios/doc/nlp/Baud93a.html, 1998.

[Shih-98] Mohammed Shihadah and Paul Roochnik, “Lexical-Functional Grammar as a Computational-Linguistic Underpinning to Arabic Machine Translation”, the proceedings of the 6th International Conference and Exhibition on Multi-lingual Computing, University of Cambridge, London, 17-18 April, 1998, pages 5.8.1 - 5.8.9.

[Step-98] Stephen R Schmidt, "Introduction to Language", Cognitive Psychology, MTSU, http://www.mtsu.edu/~sschmidt/Cognitive/Language/langout.html, 1998.

[Susa-96] Susanne Riehemann, "Head-Driven Phrase Structure Grammar: Some leading ideas", http://hpsg.stanford.edu/hpsg/leading-ideas.html ,1996.

[Thay-89] A. Thayse, From Modal Logic to Deductive Database, Introducing a Logic Based Approach to Artificial Intelligence, Wiley, Chichester, 1989.

[Thay-91] A. Thayse, From Natural Language Processing to Logic for Expert Systems, A Logic Based Approach to Artificial Intelligence, Wiley, Chichester, 1991.

[Wins-92] Winston H., Artificial Intelligence, 3rd Edition, Addison-Wesley, 1992.

# Appendix A

## List of Publications

This is a list of the publications that I published jointly with my supervisors and colleagues during the progress of the research. Find their abstracts on my homepage at http://mtaqi.hypermart.net.

**Al-Khonaizi M.**, al-A'ali M., and Al-Zobaidie A., "The common sense knowledge structure", the 5th International Conference and Exhibition on Multi-lingual Computing, ICEMCO-98, Cambridge University, April 1998, Pp2.3.1-2.3.9.

**Al-Khonaizi M.**, al-A'ali M. and Al-Zobaidie A., "The Functional and Semantic Structures for the Arabic Natural Sentence", the 4th IEEE Int. Conference on Electronics, Circuits, and Systems (ICECS'97), Cairo, Egypt, December 15-18, 1997.

**Al-Khonaizi M.**, Yamani A., Al-Zobaidie A. and al-A'ali M., "A New Declaratives & Interrogatives Approach to Natural Arabic Text: A Computational Linguistics Approach", the 5th International Conference and Exhibition on Multi-lingual Computing, ICEMCO-96, Cambridge University, April 1996, Pp 3.13.1to 3.13.10.

**Al-Khonaizi M.**, and al-A'ali M., "An Arabisation Methodology for very large computer applications", the 5th International Conference and Exhibition on Multi-lingual Computing, ICEMCO-96, Cambridge University, April 1996, Pp 3.6.1 to 3.6.9.

**Al-Khonaizi M.**, Al-A'ali M., &Al-Zobaidie A., "Al-Khazoon: A Database System for Natural Arabic Language Text", in Processing Arabic, Report 8, Nijmegen 1995, Ditters E. (Ed.), Institute for the Languages & Cultures of the Middle East, -Nijmegen University Holland, ISSN 0921-9145, Pp 107-114.

Al-Zobaidie A., **Al-Khonaizi M.**, Yamani A., al-A'ali M, "Common Sense Knowledge Representation for Natural Language Processing: an LFG approach", in Processing Arabic, Report 8, Nijmegen 1995, Ditters E. (Ed.), Institute for the Languages & Cultures of the Middle East, -Nijmegen University Holland, ISSN 0921-9145, Pp 47-78.

**Al-Khonaizi M.**, Al-Aali M., and Al-Zobaidie A., "The Classification Approach for Natural Arabic Language Understanding", The Egyptian Computing Journal, Vol. 22, No.1, 1994, The Institute of Statistical Studies & Research, Cairo University, Pp 59-76.

Yamani A, **Al-Khonaizi M.**, Al-Zobaidie A. and Al-A'ali, "Understanding Declaratives and Interrogatives of Arabic text: A Computational Linguistic Approach", The Second Symposium of the Arabization of Computers, Vol. 2, Riyadh, Saudi Arabia, March 1994.

**Al-Khonaizi M.**, Al-Aali M., and AL-Zobaidie A., "Understanding Natural Arabic Text", Proceedings of the International Conference of the Human-Computer Interaction, Vienna, 20-22, September 1993, Pp 407-408.

**Al-Khonaizi M.**, Al-Aali M., and AL-Zobaidie A., "Al-Khabier Al-Arabi, An Expert System for Natural Arabic Language Understanding", The proceedings of the Second National Expert Systems and Development Workshop ESADW-93, Cairo, 2-6 May, 1993, Pp 193-200.

# Appendix B

## The 29 sentences used as an example for this work

### The Original Arabic Text

The original text described below is composed of 8 compound sentences before implementing the c-structure Assumptions. It is a translated copy of an English text describing road traffic accidents. This domain has been selected as an example for the implementation purpose.

عصف، وقوع حادثين كبيرين، بسكون شارع سكني هادىء، وذلك في غضون ساعتين من الاسبوع الماضي ①. ولا زال احد السائقين بالمستشفى في وضع حرج يعاني من اصابات في الرأس بينما تعرض عدد من السيارات وحافلة لنقل الركاب لأضرار كبيرة او انها تلفت تماما ②. كما تهدم الحائط الامامي ورواق احدى المنازل، وقد وقعت هذه المأساة في اوكيهامبتون كرسنت بمنطقة وبلنج في الساعة 6:45 مساء من يوم الاربعاء وذلك عندما اصطدمت سيارتان وجها لوجه احداهما من نوع اوستي ايجرو والاخرى من نوع سيرا كوسورث ③. هذا وقد اخرج سائق سيارة الاوستي (انتوني هيل) والذي يبلغ من العمر 26 عاما، من سكان ريكلمارش رود بمنطقة بلاك هيث، من سيارته بواسطة قوة اطفاء منطقة بلمستيد وذلك بقطع اجزاء من سيارته ليتمكن من الخروج من بين حطامها ④. ومن ثم اخذ الى المستشفى (كوين ميري) بمنطقة سيدكب، ثم نقل بعد ذلك الى قسم "العناية المركزة " حيث ذكر يوم الاثنين بأنه خرج من غرفة العناية المركزة وان صحته في تحسن ⑤. اما سائقة سيارة السيرا (جولي اندروود) وهي من سكان نور شميرلاند افنيو بمنطقة ديلنج فتعاني من اصابات في الوجه، وقد أخرجت من مستشفى (كوين ميري) يوم الجمعة الماضي ⑥. كما ان اربعة اطفال كانوا داخل السيارة اصيبوا باصابات طفيفة ⑦. وبعد ذلك بساعتين فقط وفي اثناء وجود الشرطة في مكان الحادث لمعاينته تدحرجت حافلة لنقل الركاب الى الخلف على طريق ايكسهوث بعد تعطل فراملها واصطدمت بعدة سيارات واصطدمت بمؤخرة سيارة شرطة سرية وتحركت الحافلة عبر الشارع حيث هشمت حائط امامي ثم اصطدمت بسيارة كانت تقف على مدخل جانبي ثم اصطدمت بالمنزل مما ادى الى تهدم الرواق وقد تعرض اثنان من رجال الشرطة الى اصابات خفيفة، اما سائق الحافلة وهو من سكان بير كشير فلم يصب بأذى، وتقول الشرطة انه من حسن الحظ ان ثلاثين طفلا من مدرسة بمنطقة ميدلسيكس كان قد تم اخلائهم من الحافلة في وقت مبكر من اليوم بعد اكتشاف عطل ميكانيكي فيها وقد وجهت الشرطة نداء الى شهود عيان للحادث الاول كما قالت ان كل العربات التي كانت في الحادث قد تم فحصها وكل من لديه معلومات عليه الاتصال برقيب الشرطة (سوندرز) على هاتف رقم (180-3-11212) ⑧.

## The Text after Normalization

After implementing the c-structure assumptions, the original natural Arabic paragraph has become 29 sentences as described below.

(1) "عصف وقوع حادثين كبيرين ب سكون شارع سكني و هادئ في غضون ساعتين من ال اسبوع ال ماضي"
(2) "لازال احد ال سائقين ب ال مستشفى يعاني من اصابات في ال راس في وضع حرج"
(3) "تعرض عدد من ال سيارات و حافلة ل نقْل ال ركاب ل اضرار كبير ة"
(4) "ان ها تلفت تماما"
(5) "تهدم ال حائط ال امامي و رواق احدى ال منازل"
(6) "وقعت هذه ال مأساة في اوكيهامبتون كرسنت ب منطقة وبلنج في ال ساعة ر6_45 مساء من يوم ال اربعاء"
(7) "اصطدم ت سيارتان وجها ل وجه"
(8) "احدا هما من نوع اوستي ايجرو"
(9) "ال اخرى من نوع سيرا كوسورث"
(10) "اخرج سائق سيارة ال اوستي انتوني هيل الذي يبلغ من ال عمر ر26 عاما من سكان شارع ريكلماش ب منطقة بلاك"
"بلاك هيث من سيارة ه ب واسطة قوة اطفاء منطقة بلمستيد ب قطْع اجزاء من سيارة ه ل يتمكن من ال خروج من بين حطام ها"
(11) "اخذ الى مستشفى كوين ميري ب منطقة سيدكب"
(12) "نقل الى قسم ال عناية ال مركزة بعد ذلك"
(13) "ذكر يوم ال اثنين ان ه خرج من غرفة ال عناية ال مركزة و ان صحة ه في تحسُن"
(14) "سائق ة سيارة ال سيرا جولي اندروود من سكان طريق نورشميلاند ب منطقة ديلنج تعاني من اصابات في ال وجه"
(15) "اخرج ت من مستشفى كوين ميري يوم ال جمعة ال ماضي"
(16) "ان اربعة اطفال كان وا داخل ال سيارة اصيب وا ب اصابات طفيفة"
(17) "بعد ذلك ب ساعتين فقط و في اثناء وجود ال شرطة في مكان ال حادث ل معاينة ه تدحرج ت حافلة ل نقْل ال"
(18) "اصطدم ت ب عدة سيارات"
(19) "اصطدم ت ب مؤخرة سيارة شرطة سري ة"
(20) "تحرك ت ال حافلة عبر ال شارع"
(21) "هشم ت حائط امامي"
(22) "اصطدم ت ب سيارة كان ت تقف على مدخل جانبي"
(23) "اصطدم ت ب ال منزل"
(24) "ادى الى تهدُم ال رواق"
(25) "تعرض اثنان من رجال ال شرطة الى اصابات خفيف ة"
(26) "سائق ال حافلة من سكان بيركشير لم يصب ب اذى"
(27) "تقول ال شرطة ان ه من حُسن ال حظ ان ثلاثين طفلا من مدرسة ب منطقة ميدلسيكس تم اخلاء هم من ال حافلة"
"حافلة في وقت مبكر من ال يوم بعد اكتشاف عطْل ميكانيكي في ها"
(28) "وجّه ت نداء ال شرطة الى شهود عيان ل ال حادث ال اول"

(29) "قال ت (ان كل ال عربات التي كان ت في ال حادث تم فحْص ها) (ان كل مَن لدى ه معلومات على ه ال اتصال ب رقيب سوندرز ال شرطة على هاتف رقم ر180_3_11212)"