View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Edinburgh Research Archive

Linguistic and Computational Analysis of Word Order and Scrambling in Persian

Siamak Rezaei (Durroei)

PhD
University of Edinburgh
1999

Declaration

I declare that this thesis has been composed by myself and that the research reported here has been conducted by myself unless otherwise indicated.

Edinburgh, 28 February 1999
Siamak Rezaei (Durroei)

Acknowledgements

I would firstly like to thank my supervisors Matt Crocker, Elisabet Engdahl and Ewan Klein, for all their advice and support. I also greatly appreciate the constructive criticism, comments and suggestions provided by my examiners Paul Bennett and Chris Mellish. I would also like to thank all the other people with whom I have had helpful and interesting face to face and email discussions about the work presented here during my PhD study. These include, but are not limited to Robert Berwick, Miriam Butt, Muhammad Dabir-Moghadam, Reza Hashemi Gask, Zelal Gungordu, Mark Johnson, Simin Karimi, David Milward, Robert Scott, Paul Smolensky, Patrick Sturt and David Tugwel.

Some of the research reported here has been published elsewhere. I would like to thank the anonymous referees of these works for their useful feedback.

Back in Iran, I would like to thank Hassan Zand, Seyed Ali Mir-Emadi and Caru Lux for providing me the opportunity to present my work in Linguistics department of Tehran University, Allame Tabatabaei university and Intelligent Systems Institute in Tehran respectively. Especially useful feedback from Mohammad-Reza Bateni, Muhammad Dabir-Moghadam, Korush Safavi, Yadullah Samareh. It was in Alame Tabatabei's talk that I had the opportunity to meet my first professor of Linguistics Saleh Husseini after almost 10 years of my first lesson in Linguistics.

My gratitude also goes to Mehrdad Fahimi for accepting me to work with him on my first NLP thesis and a machine translation (MT) project in Iran. I also would like to thank Jim Cowie, Angelic Sena, Sergei Nirenburgh, Remi Zajac and others in CRL for the period of my employment in US and the useful experience of initiating Shiraz Persian English MT project, MT research in CRL and putting up with all the troubles of avoiding Demato sanctions and getting a US work permit.

I would also like to thank my office mates, Saturnino Luz, Patrick Sturt, David Tugwel, especially Patrick and David for going through previous drafts of my thesis. My gratitude also goes to Ardeshir Geranpayeh, Bijan Hashemi Malayeri, Keyvan Maleki, late Muhammad Bagher Muhseni, Ali Shirani, Abbas Tavazoei for a friendly circle in the UK.

I also like to thank my Kurdish (and Zaza) friends on the Internet, especially Burhan Elturan, Rebwar Fatah, Kocero, Hakki, Aseman and the Kurdish Language and Linguistics committee for all their time, encouragement and help.

Finally, I would like to thank my parents, my brother and sister for their support from Iran and my friend Ali in Iran for keeping me up to date with the news from home. Last but not least, I thank Nowicka family for their hospitality during my stay in Poland.

ABSTRACT:

This thesis discusses linguistic constraints on scrambling and flexibility in word order in spoken Persian (Farsi) and presents a computational model for efficient implementation of these constraints for a subset of Persian. Linguistic phenomena which we have studied, include local scrambling, long distance scrambling, extraposition of clauses, topicalisation, case tendency and the discourse marker *ră*. The work extends previous work on Persian based on Government and Binding (GB) theory by considering the pragmatic aspects of Persian grammar and long distance scrambling.

After the introduction, we begin by examining the main structures, concepts and constraints on local scrambling in main clauses. A constraint based account of local scrambling in Persian will be presented. We then consider the extraposition of embedded clauses in Persian, as well as control and topicalisation in Persian. We propose a new approach for capturing extraposition of embedded clauses, fronting and scrambling in a uniform theory, consistent with previous findings about Persian grammar and the Government and Binding theory. The structure that we propose for complement clauses in Persian is analogous to the structure of embedded clauses and we show that case attraction in Persian relative clauses can be easily captured by this structure.

In the next part of the thesis, we survey the main achievements of previous computational work on processing Persian syntax. Then we review some formalisms which have been extended for representing scrambling in computational linguistics. We contrast the main mechanisms to deal with scrambling and flexible word order in GPSG, HPSG, different extensions to CG and TAG, and LFG. Then after a summary of work on parallel natural language processing, we present a competition-based parser for analysing a subset of Persian. The parsing system, which avoids the inefficiency of the previous approaches for parsing Persian, uses fuzzy sets for resolving conflicts and competition among different possible alternatives. The study argues for a resource based model of scrambling that takes into account gradient grammaticality and shows the consequences of such model for implementing constraints on scrambling in spoken Persian.

Finally, we highlight the underlying dynamic framework which has motivated our study. For this purpose, we turn to dynamic theories in Computer Science such as CCS (Calculus of Communicating Systems) and π-calculus Milner [1993]. Borrowing some concepts from these theories, we will discuss how communicating linguistic processes can be defined and constructed. The notion of grammatical channels for communication between processes will be introduced under a general communication based approach to syntax and grammar. We will argue that present process models in Computer Science are not powerful enough for this purpose, and some possible directions for further research will be discussed.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Goals and Motivation

This thesis analyses a range of linguistic phenomena in Persian and in addition develops proposals for formal and computational frameworks for processing these. The essential goals of this work are twofold. Firstly, to specify constraints that govern the flexibility of word order and scrambling in Persian. Secondly, to develop methods for efficient processing of scrambling constraints in Persian. This is a step towards the possibility of using the results in developing real natural language processing applications for spoken Persian.

While there have been numerous studies of formal linguistic representation of Persian, little work on computational analysis of Persian has been done. Those few computational works which deal with Persian have either ignored the linguistic constraints on Persian such as specificity, control and scrambling constraints, or they have restricted their study to written Persian. Written and spoken Persian differ in many respects, especially in word order and scrambling constraints. We have focused on the syntactic analysis of spoken Persian, with particular attention to the constraints on word order.

1.2 Linguistic Phenomena

A major area of controversy among Persian linguists has been the discourse and syntactic functions of the postposition *rā*. The discourse-oriented camp [Mogaddam, 1992a] or Mogaddam [1992b] tries to give justification for different uses of *rā* based only on constraints in

discourse, while the other syntactically-oriented camp Karimi [1990] tries to analyze syntactic constraints on those cases. In this thesis we look at both sides and examine the discourse and syntactic functions of *rǎ* in Persian. For this purpose we have looked at analogous markers in neighboring languages with more limited or wider application than *rǎ*. Another topic of our study has been the extraposition of embedded clauses in Persian, and we have proposed a structure for representing complement clauses in Persian. We will show that the same structure can be used for representing embedded relative clauses and we will propose a solution for the interesting phenomenon of case attraction in Persian relative clauses.

From a formal perspective, the constraints on scrambling (local and long distance) in Persian have not been fully studied. Therefore we have studied different formalisms which have been proposed for scrambling, and we claim that these formalisms fail to capture all the scrambling constraints in Persian.

In Persian, different examples of surface word order can exist for a sentence. These word order possibilities differ in grammaticality. The canonical word order in Persian can be labeled as the most grammatical word order and variations from this canonical word order may have reduced grammaticality. But some variations are perfectly grammatical in specific discourse contexts. This is especially important for processing Persian; since the subject is unmarked and the direct object can also be unmarked, the canonical word order and scrambling constraints provide further clues for the disambiguation of grammatical functions such as subject and object. In addition, Persian is a highly pro-drop language and subjects and other constituents can be missing.

Examples of long distance scrambling in Persian, control and garden paths have rarely been considered in computational systems. Complex examples of long distance scrambling in which the scrambling constraints interact with control phenomena have not been studied.

Based on this study we propose a framework for processing cases with scrambling which is in the spirit of LFG and GPSG. In the literature, the ID/LP framework has been one of the choices and some of the approaches have tried to use this framework or extend it. So far few approaches have tried to introduce a probabilistic and robust version of word order constraints that takes into account performance parameters as well as competence syntactic parameters. Uszkoreit [1987] tried to introduce complex word order rules, but he does not add stochastic or probabilistic notions to these rules. Recently, notions such as probability,

optimality, possibility, plausibility, acceptability and graded grammaticality have been added to linguistic theories. Despite the fact that scrambling and word order introduce degrees of acceptability and graded grammaticality, the necessary acceptability or plausibility notions have not been added to the scrambling rules. In our study, we extend the word order rules by introducing a stochastic version of them. In this work, we use acceptability and plausibility interchangeably to refer to all these notions. We have only considered a limited subset of acceptability notions and future work is needed to fully incorporate all these notions.

Uszkoreit [1991] also reviewed some possible strategies for combining different kinds of constraints in declarative grammars with a detachable layer of control information. In our framework we propose adding an additional competitive layer to the static linguistic framework, thereby combining stochastic word order rules and degrees of specificity for *rā* as an activation value for linguistic process structures. There have been previous proposals for introducing competitive and dynamic frameworks, but each approach looks at dynamism from a different perspective. Our work has focused on dynamic approaches based on theoretical computer science and more specifically on process algebraic proposals such as Fujinami [1996]. A parallel competitive based parser has been implemented and some of the performance constraints on scrambling in Persian fall out naturally from the architecture of the parser.

We argue for a Resource Limitation Principle (RLP) and a Resource Barrier Principle (RBP) in Persian. These principles further constrain the possible examples of scrambling in Persian. We further claim that the RLP and RBP constraints can be used for categorizing free word order languages.

1.3 Organization

The thesis is divided into two parts: (1) Linguistic (2) Computational. Part 1 consists of Chapters 2 and 3, which contain linguistic analysis of Persian. In Chapter 2, we begin with a general discussion on Persian grammar and local scrambling in main clauses. Since local scrambling in Persian is closely connected to the semantics of nouns and specificity in the semantics of nouns, specificity and definiteness will be discussed in Section 2.2. Then after a short introduction to the structure of constituents in Persian, we discuss verb morphology. The majority of verbs in Persian are complex verbs and the study of scrambling inside a

verbal complex needs separate research that we don't discuss. Finally the canonical word order in Persian and constraints on local scrambling are formulated.

In Chapter 3 we examine embedded clauses in Persian. Firstly, we discuss the canonical position of complement clauses and discuss the problems that a non-extraposed proposal will face. In particular we look at the interaction of extraposition of complement clauses with extraposition of relative clauses and the interaction between extraposition and topicalization. These problems serve to motivate our treatment of complement clauses as being canonically pre-verbal. We propose analogous structures for representing complement clauses and embedded clauses in Persian and in this way we show that the same principles in Persian are responsible for two different phenomena, namely topicalisation and case attraction in Persian. Such an approach is a step towards the formalisation of case tendency in Persian. We will also look closely at control in Persian which interacts with long distance scrambling.

The second part of the thesis consists of Chapter 4 through Chapter 6, which contains a literature review, linguistic formalisms for representing scrambling, our implementation. In Chapter 4 we will review the main parsing approaches for Persian. Chapter 5 is primarily concerned with the treatment of word order and scrambling in different formalisms. We contrast different mechanisms for this purpose in GPSG, HPSG, different extensions to CG, TAG, and LFG. We also highlight some weaknesses of functional composition in CCG for representing scrambling. The V-Tag formal system also uses very complex mechanisms that are much more easily captured in LFG. The main weakness of LFG compared to the other formalisms is the lack of a proper solution for extending it with probabilities. Then we will discuss another point of difference between LFG and the other formalisms which is its way of encoding relations. Finally some proposals and questions are raised about functional uncertainty in LFG.

In Chapter 6, a parallel implementation of a parser for a subset of Persian that integrates the preceding proposals of the thesis will be discussed. The parser avoids the inefficiency of previous approaches for parsing Persian and uses fuzzy sets for resolving the conflicts and competition among different possible alternatives. The solution is designed by combining neural network and symbolic approaches to parsing. We will investigate the interaction between competition and resource limitation for the parsing system.

Finally Chapter 7 is the conclusion, and we will outline the major contributions of this

thesis and possible directions for further research. In this chapter we also highlight some directions for future work on a dynamic framework which has motivated our study.

Chapter 2

Persian Grammar: Main Clause

In this chapter and the following chapter I will present an overview of Persian Grammar. I will concentrate on the dialect of Modern Persian which is spoken in Tehran[1]. The main focus will be on the issue of word order and flexibility in movement of constituents in Persian. We will first introduce major constituents in Persian.

In this chapter, I shall concentrate on constructions of Persian excluding Persian embedded clauses that will be the topic of the next chapter.

2.1 Introduction

Persian is a free constituent order language. It is an Indo-European language – a south western Iranian language from the Indo-Iranian branch. Persian has been the language of those Indo-Europeans who had moved to the south of the Persian plateau. These people had called their occupied regions the land of Eras or Aryans[2] [Karimi, 1989]. Since these people were in contact with other nations, especially the Semitic people who lived to the west of the plateau, and more recently the Turks, their language has changed. The Arabization of Persian words is vividly illustrated in the Arabization of the name of the language *Fărsi* which was once called *Părsi* to which "Persian" refers[3].

The ancestor of modern Persian is old Persian (6th-3rd BC). Old Persian displays struc-

[1] There are many differences between the spoken and written languages of Persian. For a survey, see [Daryabandari, 1993].

[2] Iran and Aryan are from the same root.

[3] In Arabic, there is no /p/ sound.

tural similarities to other ancient Indo-European languages. It exhibits seven cases, three genders (feminine, masculine and neuter), and three numbers (singular, plural and dual) [Karimi, 1989]. Most of the case, number and gender inflections are no longer present in Modern Persian, and Persian has become a morphologically simple language.

The major constituent order of Persian (as Dutch and Turkish) is mainly SOV. Persian, like Italian and Turkish, is a pro-drop language. In other words the subject of a sentence can be absent (and the ending of the verb determines the person and number of the subject). Pronouns and noun phrases of Persian have no case marking inflection. Persian, like English and Arabic, is a head-initial language (except for VP). That is the complements of a phrasal category follow its head.

Before plunging into the details of phrases like NP, PP and S, some simple sentences of Persian are shown. In these examples SPCF stands for specificity-marker which marks specific noun phrases (for [-NOM][4] case); and EZ stands for ezafe (see Section 2.2.2).

(2.1) ali man ră did.
Ali I SPCF saw-3S
'Ali saw me.'

(2.2) man raft-am be madrese.
I went-1S to school
'I went to the school.'

(2.3) u xord sib ra.
he/she ate-3S apple SPCF
'He/She ate the apple.'

(2.4) sib ali xord.
apple Ali ate-3S
'Ali did apple-eating.' = 'Ali ate apples.'

(2.5) dăd-am man sib-e qermez ră be u.
gave-1S I apple-EZ red SPCF to he
'I gave him the red apple.'

[4]Every specific noun phrase that is not subject.

It is important to note that the writing system of Persian is an extension of Arabic writing and, as Arabic, for some vowels namely (o,e,a) there are no corresponding letters in its alphabet. As a result one of the most important markers of Persian (i.e. Ezafe-marker) is not always shown in the texts. Readers of Persian texts learn little by little when the marker should be assumed to be present; in our work we explicitly show this marker by -e (as is the case for beginners in this language).

2.2 Noun Semantics and Morphology

In this section we will review the morphology of Persian noun phrases, and will discuss definiteness, indefiniteness and specificity for nouns. We will also consider other morphemes and clitics that attach to nouns in Persian.

2.2.1 Specificity/Definiteness

Languages of the world can be divided into groups in which they have either a definite marker or a specificity marker. In some languages we might not see either of the two. In Persian, there is a marker that follows specific noun phrases under certain conditions, but there is no definite marker. Turkish and Albanian also have a marker for specificity. In contrast English, German and Kurdish are languages which have definite markers.

By specificity we have a meaning in mind, which implies that a noun phrase refers to a particular individual member (or members) of a class rather than to the class as a whole [Karimi, 1989]. In English, noun phrases with this meaning can be expressed in several ways [Weissberg and Buker, 1991]:

1. Referring to assumed or shared information, e.g.:

   (2.6) In recent years the growth of desert areas has been accelerating in *the world*.

2. Pointing back to old information, e.g.:

   (2.7) Iranian Banking authorities are developing a computerised monetary system. *The system* will be used throughout the country.

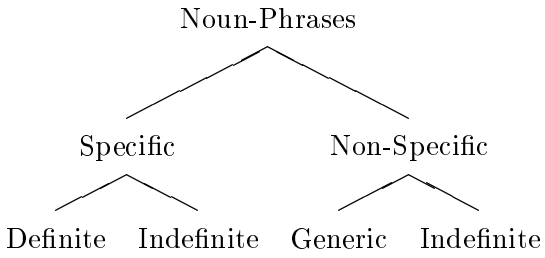
3. Pointing forward to specifying information, e.g.:

(2.8) *The man* who was here was arrested.

The specific/non-specific reading can be tested by using *it* and pronoun *one*, respectively:

(2.9) Mary was looking for a bike, and She found *one*. (non-specific)
She found *it*. (specific)

The following diagram from [Karimi, 1989] further clarifies the distinction between specificity with respect to definite, indefinite, and generic NP's.



In this diagram the difference between the definite NP's and specific NP's is that the former is presumed to be known to the hearer whereas the latter is not.

In summary specific noun phrases, definite or indefinite have one feature in common: they denote a specific individual.

In Persian there is no article or suffix for marking definiteness, but there is a suffix /-i/ for marking indefinite noun phrases and a postposition *ră*[5] for marking specific noun phrases under certain conditions. But it doesn't come after specific subjects.

Nouns in Persian are generally treated as generics when they are not followed by any semantic marker[6], although there are counterexamples. Other aspects of the semantics of Persian noun phrases are not very straightforward; and any modification of a noun phrase may change its semantics [Windfuhr, 1979]. These cases are illustrated by further examples:

1. Bare noun in subject position:

   In this position, bare nouns can act as specific subjects, if they are already mentioned in the previous context. Alternatively they can act as generics.

[5]The postposition *ra* in spoken language may appear as *ro* or as */-o/*.
[6]By semantic marker we are considering *-i* and *ră* that follow the nouns in Persian.

(2.10) gorg dar jangal ast.
'wolf in jungle is-3S'
'The wolf is in (the) jungle.'
wolves are in jungle.

2. -i marked nouns in subject position:

When a noun is marked with -i it is usually treated as an indefinite.

(2.11) gorg-i dar jangal ast.
wolf-IND in Jungle is-3S
'a(some) wolf is in (the) jungle.'

In (2.11) *-i* may be an indefinite marker. In this case we are referring to some wolf.

Persian grammarians, in addition to '-i of indefiniteness' (yă tankir) consider another role for *-i* as '-i of unit' (yă vahdat). In English, a comparable ambiguity exists with *a(n)* e.g. *I am looking for a car*, where *a* may imply *one* particular car or any car [Windfuhr, 1979]. In Persian, if *-i* is a '-i of unit' then we can have the same meaning by deleting *-i* and putting *yek* (one) before the noun:

(2.12) yek gorg dar jangal ast.
one wolf in Jungle is-3S
'a wolf is in (the) jungle.'

Still there are cases where *-i* may be none of the above. In (2.13) *-i* acts as a restrictive relative clause marker and the -i marked *gorg* is more specific[7] than the corresponding *gorg* in the previous examples.

(2.13) gorg-i ke to goft-i, dar jangal ast.
wolf-RES that you said-3S, in Jungle is-3S
'The/a certain wolf that you said is in (the) jungle.'

Other modifiers of a noun also contribute to an increase in the specificity of the noun.

[7] In Persian, specificity is a continuous notion

3. Bare noun as a specific object: In this example *gorg* (wolf) is a specific object which should be mentioned in the previous discourse.

(2.14) u gorg ră did.
s/he wolf SPCF saw-3S
'S/he saw the wolf.'

We can make nouns plural by adding the plural suffix *-hă*:

(2.15) u gorg-hă ră did.
s/he wolves SPCF saw-3S
'S/he saw the wolves.'

By this we also add to the specificity of the noun and *ra* becomes obligatory[8].

(2.16) * u gorg-hă did.
s/he wolves saw-3S
'S/he saw the wolves.'

4. Bare noun as a specific object (species):

In the next example *gorg* refers to the species of wolves.

(2.17) gorg ră nabăyad šekăr=kard.
wolf SPCF shouldn't hunt=did-3S
'One shouldn't hunt wolves.'

5. Bare noun as a non-definite object (species):

While in the following, *gorg* refers to the generic noun.

(2.18) gorg nabăyad šekăr=kard.
wolf shouldn't hunt=did-3S
'One shouldn't hunt wolves.'

The exact meaning of the sentence can be represented by a new noun-incorporated predicate, *wolf=hunting*; the sentence means *one shouldn't do wolf=hunting*,

[8] As we discuss later, *ră* is obligatory for specific direct objects

There is also an *-i* suffix for making abstract nouns or adjectives from adjectives and concrete nouns, as shown in

(2.19) băzăr + /-i/ ==> băzări

'market' 'common of the market'

(2.20) pir + /-i/ ==> piri

'old' 'oldness'

To conclude this section we present another example of the change of the semantics of a noun by affixing -i to it. In the following example, adding *-i* to *ăbejo*(beer) makes it a count noun [Windfuhr, 1979]:

(2.21) ăbejo u xord.
beer he drank-3S
'He drank beer.'

(2.22) ăbejo-i u xord.
beer he drank-3S
'He drank (a glass of) beer'

2.2.2 Ezafe

Ezafe is used in most constructions in Persian. It is specified by the occurrence of a morpheme *-e* before phrasal complements and modifiers that follow the head [Samiian, 1983]. When *-e* attaches to a word that ends with a vowel, a *y* is also inserted before *-e*. An example is shown in (2.23).

(2.23) ro + -e → ro-ye

The function of Ezafe in NP's is very similar to *no* in Japanese. In NP's, Ezafe sometimes acts like 'of' as in "destruction of the city" and it also comes before the adjectives (in Persian adjectives come after the noun that they modify):

(2.24) xord-an-e sib
eat-ing-EZ apple
'Eating the apple'

(2.25) sib-e qermez
apple-EZ red
'red apple'

Ezafe is also present in prepositional phrases and comes after the preposition:

(2.26) barăy-e ali
for-EZ Ali
'For Ali'

Ezafe can be found in adjective phrases (AP):

(2.27) montazer-e ali
waiting-EZ Ali
'Waiting for Ali'

According to Samiian [1983], ezafe in Persian is not a preposition and it is transformationally inserted inside phrases. But ezafe also has a syntactic function. According to Karimi [1989] ezafe in Persian transfers the case of the head noun to its complements. [Karimi and Brame, 1986] has argued that all phrases contained in an ezafe construction are noun phrases and it further argues that adjectives in Persian structurally behave like nouns.

In the next chapter we will further discuss some interesting examples of ezafe case marking. Here are more examples of ezafe constructions:

(2.28) sib-e qermez-e ali
apple-EZ red-EZ Ali
'The red apple of Ali'

(2.29) roy-e dar-e madrese-e siămak.
on-EZ door-EZ school-EZ Siamak
'On the door of Siamak's school.'

In (2.28) the noun phrase is specific. In general noun phrases containing a genitive noun phrase are specific in Persian [Karimi, 1989].

In passing, it should be mentioned that ezafe in Persian is a clitic [Shaghaghi, 1993]. In the following section we will refer to some other clitics[9] in Persian.

2.2.3 Clitics

In Persian there are suffixes which are attached to the verb, preposition and to the head of the genitive constructions, and co-index with nouns in these constructions.

Here are the set of these suffixes in Persian.

Pers/No	Single	plural
1st	-am	-im
2nd	-i	-id
3rd	-ad/#	-and

Table 2.1: Subject Inflections

Pers/No	Single	plural
1st	-am	-emăn/amăn
2nd	-et/at	-eton/aton
3rd	-eš/aš	-ešon/ašon

Table 2.2: Oblique (Direct or Indirect Object, Preposition and Genitive)

The oblique suffixes that are attached to the verb, preposition, and to the head of the genitive constructions are instances of clitic pronouns. As Hashemipour [1989] has shown, unlike the subject verb inflections these are clitics and not inflectional affixes. Hashimpour argues that these meet the criteria of clitics given by Pullum and Zwicky. Because [Hashemipour, 1989]:

- First, they are positioned relative to syntactic constituents (zero-level heads in X-bar notation and not roots or stems).
- Second, unlike verb inflection, clitic pronouns are optional.
- Third, the clitic pronouns exhibit a low degree of selection with respect to their host.

[9] We will restrict our study to the clitics that replace nouns. Shaghaghi [1993] studies the clitics in Persian, based on the test criteria for clitics in [Zwicky and Pullum, 1983] and [Zwicky, 1983]. Here is a list of some clitics in Persian :
indefinite and restrictive /-i/ ; short forms of *ră* , *va* (and), *ast* (to be) and *ham* (too)

As we see again here, there is a distinction between the subject suffixes and the non-subject ones which are clitics, we refer to them as oblique clitics.

The subject suffixes attach to the verb and usually agree with the subject of the sentence.

subject:	man raft-am	→	raftam
	I went-1S		I went
direct object:	u ră did-am	→	didam-eš
	he SPCF saw-1S		I saw-him
dative :	be u goft-am	→	goftam-eš
	to him told-1S		told-him
possessive:	ketăb-e u	→	ketăb-eš
	book-EZ he		his book
Preposition:	az u	→	az-aš
	from he		from him

These suffixes are used in the process of topicalisation in Persian.

(2.30) ali ră, [ketăb-eš] ră xund-am.
Ali SPCF [book-his] SPCF read-1S
'As for Ali, I have read his book.'

(2.31) ali, ketăb-eš ma'ruf ast.
Ali book-his popular is-3S
'Ali, his book is popular.'

(2.32) ali ră, [did-am-(eš)].
Ali SPCF [saw-1S-(him)
'Ali, I have seen him.'

(2.33) ali ră, [goft-am-(eš)].
Ali SPCF [siad-1S-(him)
'Ali, I told him.'

As we discussed earlier, *ră* only appears after the nouns which were not subjects (*ră* as a specific oblique marker). Again we can see a distinction between subject and non-subject nouns here. In the case of the non-subjects we observe that the specific[10] oblique marker *ră* appears during the so-called process of topicalisation. But it doesn't appear after a noun

[10]Note that an element co-indexed with a clitic pronoun is always specific.

which is co-indexed with a clitic with a subject functional role. As we have seen, the solution of Karimi captures in a principled way many of the complexities of *rā* in Persian. We will later discuss this issue further.

2.3 Major Constituents

In this section we will consider the nominal and verbal constituents of Persian such as noun phrase, prepositional phrase, adjective phrase, complex verbs and finally main clause. As we said earlier, the discussion of embedded clauses is left to the next chapter.

2.3.1 Noun Phrase

Noun Phrase is one of the phrasal constructions of Persian, in which the order of categories is fixed. In the following, the general order of NP is shown [Samiian, 1983]:

(2.34) NP → N (NP) (AP) (PP) (NP|S)

The Noun (N) is the head of the phrase. The second noun phrase is an attributive noun that modifies the first noun:

(2.35) sabzi-e ăš.
vegetable-EZ stock
'Stock vegetable.'

The AP (Adjective Phrase) also modifies the first Noun. This AP should have no complement, e.g. :

(2.36) sib-e bozorg-e qermez.
apple-EZ big-EZ red-EZ
'The big red apple.'

The prepositional phrase is of time or location; e.g.

(2.37) sib-e roy-e zamin.
apple-EZ on-EZ ground
'The apple on the ground.'

The last NP is a genitive noun phrase as in:

(2.38) sib-e ali.
apple-EZ Ali
'Ali's apple.'

We could also modify a NP with a sentence (i.e. a relative clause), this sentence can also be extraposed to the end of the sentence which contains the NP:

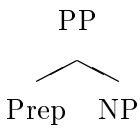
(2.39) sib-e qermez-i ră $[_{Rel}$ ke did-i] xord-am.
apple-EZ red-REL SPCF $[_{Rel}$ that see-2s] ate-1S
I ate the red apple that you saw.

(2.40) sib-e qermez-i ră xord-am $[_{Rel}$ ke did-i.
apple-EZ red-REL SPCF ate-1S $[_{Rel}$ that see-2S]
I ate the red apple that you saw.

A more detailed analysis of the noun phrase domain is available in [Samiian, 1983].

2.3.2 Prepositional Phrase

PP's in Persian consist of a preposition followed by a NP :



Depending on the preposition an Ezafe may be added to the end of it. For each preposition this Ezafe is either obligatory, optional or forbidden after the preposition[11]:

(2.41) ro-ye zamin
on-EZ ground (Obligatory)
'On the ground'

[11] *dar* is a preposition that can vanish before time and location noun phrases.

(2.42) to(-ye) madreseh
In(-EZ) school (Optional)
'In school'

(2.43) dar madreseh
In school (forbidden)
'In school'

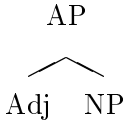
Prepositions strictly subcategorise for an obligatory noun phrase, but there are some prepositions that act like nouns and can occur alone or form PP's that become the complement of some prepositions. e.g.

(2.44) ro-ye zamin
on-EZ ground
'On the ground'

(2.45) on ro
that on
'That surface'

2.3.3 Adjective Phrase

It is not obvious whether the category AP exists in Persian, since adjectives behave syntactically like nouns. Nevertheless the general format of AP in Persian as shown by Samiian [1983] is :



(2.46) montazer-e ali
waiting-EZ Ali
'Waiting for Ali'

(2.47) Montazer-ăn-e qatăr
waiting-PluralMarker-EZ train
'People waiting for the train'

2.3.4 Verbal Complex

So far the nonverbal phrasal categories of Persian have been considered. In this section we will consider the structure of Persian verbs. In addition to simple verbs, in Persian there are also more sophisticated complex verbs. In this section we will first discuss the morphology of verbs in Persian, and then we will examine complex verbs.

Verb Morphology

Verbs in Persian have two roots: present-tense and past-tense; by adding affixes to these roots we derive the appropriate verb. In addition to these two roots, the participle form of the verb is formed by adding -*e* to the past-tense root.

Here are some examples of simple verbs of Persian and their morphology:

(2.48)

a.	raft-am.	b. na-raft-am.
	go-1S	NEG-go-1S
	'(I) went.'	'(I) didn't go.'
c.	mi-raft-#.	d. ne-mi-raft-#.
	CONT-go-3S	NEG-CONT-go-3S
	'(He/She/It) was going.'	'(He/She/It) wasn't going.'
e.	mi-rav-id.	f. ne-mi-rav-id.
	CONT-go-2P	NEG-CONT-go-2P
	'(You) are going.'	'(You) are not going.'
g.	be-rav-id.	h. na-rav-id.
	SUBJ-go-2P	NEG-SUBJ-go-2P
	'If you go.'	'If you don't go.'

Affixes (such as the negative marker 'NEG' and the continuous tense marker 'CONT', and the subjunctive marker 'SUBJ') prefix to the verb forms. The possible order for verb morphology is shown in Fig 2.1 [Rezaei, 1992].

The order is shown in a transition network diagram. In this diagram PRESENT and PAST stand for present and past roots of the verb respectively, while PRS-INF, PST-INF, PC-INF correspond to present inflection, past inflection and participle inflection. By following the appropriate arcs[12] one can generate the appropriate inflection for a tense. We will use examples of different tenses of the verb throughout the chapters[13]. Coming out of state 1 we can obtain the appropriate present (state 10), continuous (state 11), reported past (state 17), remote past (state 18) and so on.

[12] # stands for empty arc and NEG stands for negative prefix *ne*.

[13] In this diagram we have restricted ourselves to a traditional classification of Persian verbs according to tense, while more recent analysis of Persian verbs shows that an aspectual classification better represents the verb system in Persian. See [Windfuhr, 1979].

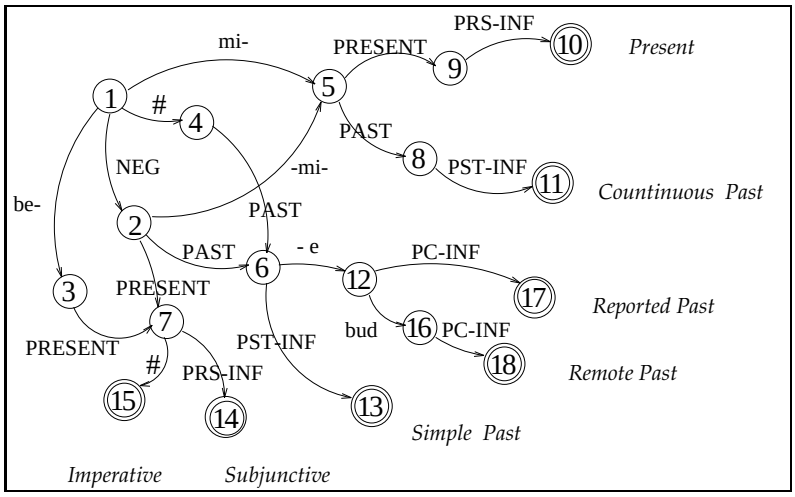
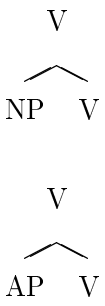


Figure 2.1: Persian Verb Morphology System

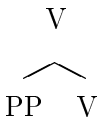
These affixes plus some auxiliaries like *bud-an* (i.e to be) that come after the verb, produce the possible forms of the verb. These forms may correspond to one or more types of the verb according to a tense/aspect classification[14].

Complex Verbs

In Persian there exists a mechanism for deriving verbs from adjectives and nouns. The new verbs that are created in this manner are called complex verbs. The mechanism is often used for deriving verbs from foreign or borrowed nouns and adjectives. At present most Persian verbs are of this kind, and even simple verbs are being replaced by these complex verbs. These complex verbs can be compared to the similar idiomatic verbs in English like 'go dutch', 'take a seat' and 'make a speech' [Aghbar, 1981].



[14]The interested reader can see [Windfuhr, 1979] and [Kamyar, 1992] for further discussion and references.



Complex verbs in Persian consist of a preverbal adjective, noun or prepositional phrase argument followed by an auxiliary root which is a simple verb root, but the adjacency of these two parts is not always necessary and depends on the complex verb. In Turkish the preverbal argument should immediately precede the auxiliary part, but the auxiliary verbs can be selected from a more restricted set compared to Persian. Here are some examples of complex verbs:

(2.49) komak kard-and.
help do-3P
'(they) helped.'

(2.50) zamin na-xord-i.
ground NEG-eat-2S
'(You) didn't fall.'

(2.51) bozorg shod-im.
big become-1P
'(We) grow.'

It is clear by now that all the grammatical affixes attach to this auxiliary root[15]. The number of verbs that act as the auxiliary is limited, and the subcategorisation of a complex verb is not the same as the subcategorisation of the auxiliary from which it is derived. Each complex verb has its own subcategorisation frame.

2.3.5 Sentence and Clause

The phrasal categories that we have described make up the major constituents of Persian clauses. By assigning one important (propositional) role or adjunct (modal) role to each constituent in a clause we can obtain grammatical clauses. These clauses can be nested or coordinated like other phrases to obtain more complex clauses.

[15]Mohammad and Karimi in their recent work propose that these verbs are instances of light verbs like *suru* in Japanese and similar cases in other languages [Mohammad and Karimi, 1993].

In the following we have shown three common types of Persian kernel sentences:

- Intransitive

(2.52) ali raft.
Ali went-3S
'Ali went.'

- Transitive

(2.53) ali ketăb ră xănd.
Ali book SPCF read-3S
'Ali read the book.'

- Stative

(2.54) ali mehrabăn ast.
Ali kind is-3S
'Ali is kind.'

(2.55) ali dar băq ast.
Ali in garden is-3S
'Ali is in the garden.'

The subcategorisation frame of a verb determines the arguments which should exist in a well-formed clause[16]. These arguments are usually marked by some grammatical markers as in most free word order languages.

In Persian if there is an argument corresponding to the subject role, it normally agrees with the verb endings, but there is no marker for it:

(2.56) man raft-am.
I went-1S
' I went.'

The direct object, when it is specific, should be marked by placing the specific marker *ră* after it:

[16]Persian verbs have been studied under the model of case grammar [Aghbar, 1981].

(2.57) ali siămak ră did.
Ali Siamak SPCF saw-3S
Ali saw Siamak.

Turkish and Persian use a very analogous mechanism for object marking, and the notion of specificity is a determining factor for both. [Browne, 1970] is one of the early works that discusses this similarity[17].

In Persian, clausal arguments appear canonically before the verb. But they are usually obligatorily extraposed to the end of the sentence, these arguments are usually marked by a clause marker *ke* that comes in front of the extraposed clause:

(2.58) ali goft ke sib ră did.
Ali said that apple SPCF saw-3S
'Ali said that he saw the apple.'

(2.59) ali či ră goft?
Ali what SPCF said
'What did Ali say?'

Still there are linguists who believe that the canonical position for clausal arguments is after the verb, and there is no obligatory extraposition (see [Karimi, 1989]).

Other grammatical functions and roles are often marked by grammatical markers (like prepositions). For example dative relations are often accompanied by *be*:

(2.60) ali ketăb ră dăd be man.
Ali book SPCF gave-3S to I
'Ali gave the book to me.'

When the markers are not present, the argument roles are distinguished by other features of the arguments (such as animate agent, inanimate instrument) or the default SOV order of Persian.

[17] Karimi argues that the specific object marker in Persian *ră* is a specific *oblique* marker. The corresponding marker in Turkish is only a specific accusative marker. In this regard, the marker in Persian has a more general function compatible with oblique marking in Iranian languages. Note that from discourse point of view there is a difference between Persian and Turkish unmarked objects.

2.4 Constraints inside a Clause Boundary

2.4.1 Passive and Causative

The phenomenon of passivisation in Persian is a controversial issue. Since Persian is a free word order language and the object of a sentence can easily be preposed to the beginning of the sentence, there is no need for a mechanism like passivisation to prepose the object, and it seems that the main reason for making passive sentences in languages like English is to bring the object into the focus. There are some Persian linguists (e.g. Moyne [1974]) who believe that in Persian there is no passive. On the contrary, however most linguists claim that there does exist such a process in Persian and believe that for passivisation there is a strict morpho-syntactic process. But all linguists agree that this process is not a general way for passivisation and is applicable only to a subclass of verbs. In this process, the passive is expressed by the (perfect) participle of the verb (Past_root + -eh) and the full paradigm of *shav* (to become) (i.e. shodam shodi shod shodim etc) e.g. :

(2.61) a. ketăb ră xănd-am b. ketăb xănd-eh sho-d.

book SPCF read-1S ==> book read-PRTCPL became-3S

'I read the book.' 'The book was read.'

It is important to note that the most obvious difference between the passive in Persian and European languages like English is the fact that the passive in Persian has no overt agent. The process described above is not a general one because for complex verbs this process is not often applicable and there is usually a complex passive verb for each transitive complex verb, with the same preverb argument but a different auxiliary part[18]:

(2.62) baqi gozăštan ==> baqi măndan

'to leave' 'to be left'

(2.63) kotak zadan ==> kotak xordan

'to beat' 'to be beaten'

[18]This auxiliary part is usually from *shav* (to become) or from *xord* (to eat).

A general mechanism for passivisation exists in Persian that is always used in common speech and is applicable for most transitive verbs (both simple and complex)[19]. In this process the agent is deleted and the verb inflection is changed to third person plural form [Windfuhr, 1979], e.g:

(2.64) ali man ră did. man ră did-and.
Ali I SPCF saw-3S ==> pro I SPCF saw-3P
'Ali saw me.' 'I was seen.'

In this process the assumption is that the agent of the act is unknown and need not be mentioned [Moyne, 1974].

As is now clear, the so called passive in Persian is a transformation that changes the subcategorisation frame of a transitive verb. There is yet another construction - called the causative construction, that can change the subcategorisation frame of a verb by introducing a causative agent into it; the new verb will be a transitive verb that agrees with the causative agent. For some verbs there exists a morphological process for creating their causative counterparts but this is not a general rule (see Dabir-Mogaddam [1982]):

(2.65) Root + ăn/ăin → causative

dav + ăn → davănd
'run' 'make s.o. run'

(2.66) u dav-id.
he ran-3S
'he ran.'

(2.67) u ali ră dav-ăn-d.
he Ali SPCF run-CAUSE-3S
'he made Ali run.'

[19] See Samareh [1989].

2.4.2 Verb Preposing

In Persian it is possible to bring the verb of a sentence to its beginning. This phenomenon of verb preposing is triggered by a discourse function of some sort in main clauses. The preposed verb reveals an emphatic interpretation or an interrogative interpretation [Karimi, 1989][20]. In the following examples the first sentence is an instance of emphatic interpretation, while the second is an example of interrogative interpretation. The two are distinguished by having different prosody.

(2.68) raft rămin.
went-3S ramin
Ramin went.

(2.69) raft-i to?
went-2S you
'Did you go?'

2.4.3 Local Scrambling inside a Clause

Canonical word order of Persian

The movement of categories inside Persian clauses is so free that even if we do not consider the free movement of prepositional phrases, there still exists a high level of movement for other categories (but of course restricted by a set of constraints). For example all the orders of constituents which are shown in Table 2.3 are possible in main clauses[21] Karimi [1989]:

S:Subject	O:Object V:Verb
SV	VO
SOV	OVS
VS	V
OSV	VSO
OV	SVO

Table 2.3: Possible Orders Inside a Clause

These possible orders are restricted by the following constraints:

[20] In Persian verb preposing is not the only mechanism for making interrogative sentences; change of intonation is another common mechanism.

[21] Karimi [1989] proposes that in subordinate clauses, preposing of verb is not possible because *comp* position is already full in these clauses.

1. If a noun phrase is specific, but there is no *rǎ* after it then it can't be the object.
2. *rǎ* does not appear with subjects, objects of prepositions or predicate nominals.
3. Objects which come after the verb should be specific (+specific). i.e *rǎ* is obligatory[22].
4. The subject agrees with the verb. But the subject can be left out, specially when the verb is not third person.
5. If neither subject nor object is specific and both agree with the verb then the object is the noun phrase before the verb (in this case the object comes in its canonical position). That is, the possible word orders in this case are *SOV* and *OVS*.

The *rǎ* marking in 1-2 above helps to clarify the functional relation (object or subject) of a noun phrase and restricts the set of possible interpretations and hence possible word orders in a sentence.

Note that the canonical position for dative objects and destination adverbials is after the verb. Hence in this regard Persian is a split word order language and the noun phrases corresponding to these should be treated separately.

Extraposition

In this section we will review some examples of extraposition in Persian. Since we are not dealing with embedded clauses in this chapter, we will postpone the discussion on extraposition of embedded clauses to the next chapter. Most of this section is based on [Qolamalizade, 1993].

Gholam-ali-zadeh considers many examples of extraposition in Persian. We will look at the extraposition of adjective phrases and prepositional phrases.

Adjective phrases in Persian are usually extraposed from inside a noun phrase, if that noun phrase is immediately preceding the verb of the sentence.

(2.70) u ketǎb-i binazir xarid.
he book-RES special bought-3S
'He bought a special book.'

[22]Karimi [1989] claims that any noun phrase that comes after the verb should be specific, but this is too restrictive.

(2.71) u ketăb-i xarid binazir.
he book-RES bought-3S special
'He bought a special book.'

There are some performance restrictions on extraposition of adjectives; the larger the distance between the head of the extraposed phrase and the extraposed element, the less likely the extraposition.

(2.72) * u ketăb-i az dastforuš-e kenăr-e xyăbăn xarid binazir.
he book-RES salesman-EZ next-EZ street bought-3S special
'Ali bought a special book from the street salesman.'

In the previous examples -i is the restrictive marker and adjectives can be extraposed if they are restrictive modifiers[23] of the head noun. In the following sentence, the head *pezešk* is not marked with a restrictive marker and hence it is not possible to extrapose the adjective[24].

(2.73) * u ăn pezešk-e ast hazegh.
he that doctor-EZ is expert
'He is that expert doctor.'

In addition to adjectives, prepositional phrases modifying a noun phrase can be extraposed [Qolamalizade, 1993][25].

(2.74) kelăs-i dar term-e ăyande taškil xăhad shod darbăre nahv-e zabăn-e
class-RES in term-EZ next making will become-3S about syntax-EZ language-EZ
fărsi.
Persian.
'A class for Persian syntax will be set up in the next term.'

[23]-i is a restrictive marker in Persian. We will discuss it thoroughly in the next chapter when we present restrictive relative clauses. Any restrictive modifier of a noun can be semantically derived from a corresponding restrictive relative clause, modifying the head noun.

[24]From discourse point of view, the extraposition moves the adjective to a background position and the trace becomes more focused or polarised.

[25]He also considers other interesting examples of extraposition of PPs from adjective phrases. These happens in predicative sentences and the extraposition in these cases can alternatively be considered as a case of local scrambling inside the verbal complex.

Again the prepositional phrase should be a restrictive modifier in order to be extraposed, i.e the head noun should be marked with /-i/. For the extraposition to occur, the distance between the trace of the extraposed phrase and its landing site is important. But in contrast to the extraposition of the adjective phrase, we see more freedom for this. In other words the extraposition is not restricted to the elements preceding the verb.

Gholam-Ali-zadeh considers many performance reasons for extraposition of [restrictive] modifiers. For example the greater the length of a modifier, the more likely it is to be extraposed [Hawkins, 1994]. Or the more the modifier introduces a gap between the arguments of the verb, the more likely it is to be extraposed. We also observe that the type of the modifier is a factor for extraposition. For example extraposition of adjectives is more difficult than the extraposition of prepositional phrases. Probably adjectives are generally shorter than PP's, cf. Hawkins [1994]. As we will see in the next chapter the finite clauses have more freedom in this regard.

In Chapter 6 we will propose two further performance constraints which restrict the flexibility of word order in Persian.

2.4.4 Wh-questions

In Persian it is not necessary to bring the interrogative pronoun to the beginning of the sentence and for making a wh-question we can substitute an argument in a sentence with the appropriate wh-pronoun, e.g.:

(2.75) hasan ki rā did?
Hasan who SPCF saw-3S?
'Whom did Hasan see?'

But it is still possible to bring the Wh-pronoun to the beginning of a clause (and sometimes by crossing the boundary of a clause).

(2.76) ki rā hasan did?
who SPCF Hasan saw-3S?
'Whom did Hasan see?'

In general Wh-pronouns are treated as similar to personal pronouns in Persian.

2.5 Conclusion

In this chapter after introducing the semantic notions of specificity and definiteness, we discussed the specific marker and indefinite marker of Persian language. We also studied other inflections that attach to the noun (e.g. Ezafe and clitics). Then we showed the internal structure of noun phrases and prepositional phrases in Persian and gave examples of complex verbs in Persian. Following this we specified the constraints on constituent order inside Persian main clauses and reviewed examples of extraposition and topicalisation of noun phrases. In the next chapter, we will discuss embedded clauses and long distance scrambling in Persian. The interaction between discourse marking and scrambling will also be discussed.

Chapter 3

Persian Embedded Clauses

In the previous chapter we looked at the constituent structures in Persian competence grammar and the constraints on local scrambling. In Chapter 6 we will look at a parser, a *performance* model for processing these constituents, that takes into account the flexibility of word order and the word order constraints. In addition, we will investigate the existence of performance constraints on scrambling in Persian. Do such constraints exist in Persian?

Before delving into the processing model for scrambling in Persian, we need to examine embedded clauses in Persian, and analyse examples of complex fronting and long distance scrambling out of the complement clauses (which normally appear post-verbally as subjunctive finite clauses). The first problem that we will tackle, in order to give an account of scrambling out of these clauses is the canonical position of embedded clauses in Persian. Specifically we will address the question whether the complement clauses in Persian originate canonically before the main verb of the sentence or after it. I will argue in favor of the traditional account according to which complement clauses are located preverbally and I will show that they are only subject to (long distance) scrambling after they have been extraposed to postverbal position.

In addition to these questions we should deal with *control*, which interacts with long distance scrambling in Persian. As we will show in Chapter 6, as a result of a competition between long distance scrambling and control in Persian, examples of garden paths [Crocker, 1995] can arise in Persian. Since garden paths generally pose further difficulties for natural language processing models in Chapter 6 we will tackle the performance aspect of the processing model and look at the possibility of specifying performance constraints in order to

impose further constraints on the system.

In order to look at control in Persian, we need also to discuss infinitives and non-finite clauses in Persian. In Persian, control normally occurs in subjunctive finite clauses which are to some extent analogous to the subjunctive clauses in Greek[1]. Although, the existence of control in Modern Greek is controversial (see Patrikakos [1995]), the evidence from Hashemipour [1989] on finite control in Persian suggests that the analogy with Greek may be misleading. also look at the existence of control in non-finite clauses and its possible link with control in extraposed finite clauses.

In the following we will first study Persian embedded clauses. We will address the issue of the rightwards movement of embedded clauses in Persian. We will also refer to other work which has tried to account for the properties of postverbal clauses in an alternative manner and argue against them.

We will further discuss the phenomenon of fronting (analogous to topicalisation in other languages) and long distance scrambling, and different examples of these phenomena will be discussed based on the preverbal canonical position assumption.

Next we will consider different examples of control in Persian and argue for the existence of control in Persian finite and non-finite clauses. Finally we will propose a new approach for capturing extraposition of embedded clauses, as well as fronting and scrambling in them, in a uniform theory consistent with previous findings about Persian grammar and general linguistic theory. This forms the starting point for the implementation of a parser for Persian sentences. We propose a structure for representing embedded clauses and their extraposition. This structure takes into account the existence of barriers in front of long distance movement and it can account for case attraction in Persian embedded clauses.

3.1 Finite Clausal Arguments

The position after the verb is the place where finite clausal arguments appear in Persian. For verbs which subcategorise for a clausal argument (e.g. *say* in *he said that ...*) the clausal argument (if present) appears at this position. In the following we show some instances of

[1]Greek was the official language of Iran for more than 100 years after the capture of Iran by Alexander the Great (around two centuries BC), during the rule of Greek Selucid in Iran and in the first era of Parthian rule in Iran.

finite clausal arguments.

(3.1) u aqide dărad [ke ahmad sib ră xord].
he belief have-3S [that Ahmad apple SPCF ate-3S]
'He believes that Ahmad ate the apple.'

(3.2) u fekr=kard [ke ahmad či xordeh ast].
he thought=do-3S [that Ahmad what eaten is-3S]
'He wondered what Ahmad has eaten.'

(3.3) u fekr=kard [ke ahmad če-tor sib ră xordeh ast].
he thought=do-3S [that Ahmad what-way apple SPCF eaten is-3S]
'He wondered how Ahmad has eaten the apple.'

(3.4) u goft be man [ke ahmad sib ră xord].
he told-3S to I [that Ahmad apple SPCF ate-3S]
'He said to me that Ahmad ate the apple.'

It is important that in most cases the clausal argument of the verb can also appear as a simple NP or PP[2]. In these cases the phrase corresponding to the clausal argument canonically appears before the verb:

(3.5) u in măjară ră be man goft.
he this adventure SPCF to I told-3S
'He told me this adventure.'

(3.6) u (in_i ră) be man goft [ke Ali zerang ast]$_i$.
he this SPCF to I told-3S [that Ali clever is]
'He told me that Ali is clever.'

[2]See Section 3.3 for examples of this for control verbs such as 'try' or 'persuade'.

In the last example the NP *in* co-indexes with the whole sentential argument. Examples like this have motivated some linguists such as Moyne and Carden [Moyne and Carden, 1974] to propose that[3] :

1. Sentential arguments originate in pre-verbal position in Persian.
2. they are dominated by an NP.
3. they are moved to the post-verbal position by an obligatory extraposition rule.

To my knowledge Karimi [1989] is the only linguist who presents arguments indicating that Persian finite sentential complements are not dominated by an NP node, and as a result they do not originate in the pre-verbal position. As in Dutch, the issue about whether sentential arguments originate post or pre-verbally may be controversial[4].

3.2 Non-finite Clausal Arguments

In general the infinitives of Persian (like Arabic) are treated as noun phrases and they can appear anywhere in the sentence[5]. It has been proposed under a transformational framework that all finite sentential arguments of Persian are derived from these non-finite clauses [Moyne and Carden, 1974]. That is (3.8) is derived from (3.7):

(3.7) u [raft-an-e ahmad be sinamă] ră goft.
he [go-INF-EZ Ahmad to cinema] SPCF told
'He said Ahmad's going to the cinema.'

(3.8) u goft [ke ahmad berav-ad sinamă].
he told-3S [that Ahmad go-3S cinema]
'He told Ahmad to go to the cinema.'

But there are some restrictions on these non-finite clauses. At first sight, it seems that in non-finite clauses the position immediately after the verb is occupied by a subject or object. This is why (3.9) is ambiguous.

[3]For a survey of this refer to Karimi [1989].
[4]We will further discuss this issue in Section 3.3.
[5]Note that these are actually noun phrases. As a result, unlike finite clauses they cannot co-index with an expletive like element.

(3.9) u [nasihat=kard-an-e ahmad] ră goft.
he [advice=giving-INF-EZ Ahmad] SPCF told
'He said [Ahmad's advising (someone)].'
'He said [(someone) to advise Ahmad].'

The position preceding the verb is also reserved for indefinite objects. If the position immediately after the verb is occupied by an object - rather than subject - then the subject may not be expressed at all[6]:

(3.10) * nasihat=kard-an-e ahmad -e hasan.
advice=giving-INF-EZ Ahmad -e Hasan
'Hasan's advising Ahmad.'

There are other constraints governing these clauses which we will illustrate by the following examples:

(3.11) raft-an-e ahmad be sinamă.
go-INF-EZ Ahmad to cinema
'Ahmad's going to the/a cinema.'

(3.12) raft-an[-e] sinamă-e ahmad.
go-INF-EZ cinema-EZ Ahmad
'Ahmad's going to a cinema.'

Here "going=cinema" acts as a non-finite complex predicate, and the generic destination noun seems to be incorporated semantically in the infinitival clause. There is a yet more common mechanism for expressing the same clause:

(3.13) cinema raft-an-e ahmad.
cinema go-INF-EZ Ahmad
'Ahmad's going to a cinema.'

[6]In this regard, this kind of clausal noun phrase is more general than its counterpart in Arabic, and more restricted compared to Turkish. In Arabic, the infinitive appears before the other constituents of the clause and expressing the subject and the object in these clauses at the same time is sometimes impossible. In Turkish all the constituents appear before the verb but there is no difficulty in expressing the subject.

Noun incorporation is possible for other instances of infinitives and destination phrases. But we cannot incorporate dative arguments. This is illustrated in the following examples:

(3.14) a. qazǎ dǎdǎn-e ahmad be man.
food give-INF-EZ Ahmad to I
'Ahmad's giving of food to me'

b. * qazǎ dǎdǎn-e man-e ahmad.

The infinitives can be further modified by adjectives or relatives. Examples of these are shown in the following:

(3.15) nasihat=kardan-e ziyad-e hasan
advice=giving-EZ very-EZ Hasan
'Giving too much advice to Hasan'
'Hasan's giving too much advice'

(3.16) a. ziyad nasihat=kardan-e Ali
very advice=giving-EZ Ali
'To give advice to Ali too much.'

b. Ali rǎ ziyad nasihat=kardan[* -e hasan]
Ali SPCF very advice=give-INF
'To give advice to Ali too much [* by hasan]'

c. ziyad Ali rǎ nasihat=kardan[* -e hasan]

The issue of modification of infinitives by adjectives and adverbs needs further investigation. In general infinitives in Persian can have different structures and there has been no satisfactory analysis to cover different examples of Persian infinitival clauses[7]. We will end this section by giving examples of different interpretations of non-finite clauses inside a clause.

[7] A contrastive study of Persian and Urdu infinitives might be useful. See Butt [1995]for some examples of infinitives in Urdu.

(3.17) u [did-an-e ahmad] ră goft.
he [meet-INF-EZ Ahmad] SPCF told
'He said (someone) to meet Ahmad.'
'He said (described) Ahmad's act of seeing.'

(3.18) u [qază xord-an-e ahmad] ră goft.
he [food ate-INF-EZ Ahmad] SPCF told
'He described Ahmad's eating.'
'He said Ahmad should eat food.'

In the previous examples we gave two interpretations for the Persian sentences. These interpretations are the closest translations of the above sentences into English.

In passing we should note that the function of *Ezafe* as a genitive marker in infinitives contrasts with its function in noun phrases. There, it was transformationally inserted as a mechanism for case-sharing of the head of the noun phrase and the Ezafe construct; while here in infinitives, it is a case marker for the arguments of infinitives. The interaction between these two roles needs further investigation[8].

3.3 Control Constructions

Now that we have explained the clausal arguments of Persian, we can consider complex cases of control constructions in Persian.

The fundamental mechanism of control is the co-indexation between the unexpressed subject of an embedded clause and its controller in a clause dominating it. Hashemipour in her dissertation considers a range of control phenomenon in Persian [Hashemipour, 1989]. She concentrates on control phenomenon in finite embedded clauses, and does not consider control in non-finite clauses of Persian. In Persian, unlike many other languages, control can occur in finite clauses.

As [Hashemipour, 1989] argues both *obligatory* and *non-obligatory* control are possible in Persian. The following examples illustrate the latter case. The lexical subject of the

[8]Note that (3.16-b) and (3.16-c) were the NP-dislocated version of (3.16-a).

embedded finite clause, present in (3.19), can be absent as in (3.20). However, (3.20) only admits an interpretation in which *Ali* is the understood subject of the complement clause.

(3.19) ali be Amir pišnahăd kard [ke sib ră hasan be-xor-ad].
Ali to Amir proposal did [that apple SPCF Hasan SUB-eat-3S]
'Ali proposed to Amir that Hasan eat the apple.'

(3.20) ali be Amir pišnahăd kard [ke – sib ră be-xor-ad].
Ali to Amir proposal did [that – apple SPCF SUB-eat-3S]
'Ali proposed to Amir to eat the apple.'

That is (3.20) does not allow an interpretation in which *Hasan* or some third person is the understood subject, as might be expected if the absent subject was construed as an ordinary pronominal.

Hashemipour considers the control phenomenon in *majbur kard-an* (i.e. to persuade) in (3.90) to be instances of obligatory control. In (3.90), the embedded subject position must be bound by a matrix nominal s(emantically)-selected by the matrix verb. This position cannot be filled by a lexical noun phrase and as a result (3.21) is ungrammatical. The obligatory control in (3.21) contrasts with the non-obligatory control in (3.19).

(3.21) * ali amir ră majbur kard [ke hassan sib ră be-xor-ad].
Ali Amir SPCF persuade did [that Hassan apple SPCF SUB-eat-3S]
'Ali persuaded Amir [for Hassan] to eat the apple.'

(3.22) ali amir ră majbur kard [ke – sib ră be-xor-ad].
Ali Amir SPCF persuade did [that – apple SPCF SUB-eat-3S]
'Ali persuaded Amir to eat the apple.'

In the following we show Hashemipour's classification for different kinds of control verbs:

The following examples are instances of control verbs in Persian:

to persuade:

dastur-dădan	'to order'
ejăze-dădan	'to allow'
esrăr-kardan	'to urge'
goftan	'to say'
pišnahăd-kardan	'to propose
sefareš-kardan	'to recommend
taqăză-kardan	'to request'

Table 3.1: Control by the Matrix Object of a Preposition

qol-dădan	to promise
say-kardan	to try
tunestan	to be able
xăstan	to want

Table 3.2: Subject Control

(3.23) ali man ră majbur (be in$_i$) kard [ke sib ră be-xor-am]$_i$.
Ali I SPCF persuade (to this$_i$) did [that apple SPCF SUB-eat-1S]$_i$
'Ali persuaded me to eat the apple.'

(3.24) ali man ră majbur be in$_i$ [ke sib ră be-xor-am]$_i$ kard.
Ali I SPCF persuade to this$_i$ [that apple SPCF SUB-eat-1S]$_i$ did-3S
'Ali persuaded me to eat the apple.'

(3.25) ali man ră majbur be [xord-an-e sib] kard.
Ali I SPCF persuade to [eat-INF-EZ apple] did-3S.
'Ali persuaded me to eat the apple.'

(3.26) ali man ră majbur kard be [xordan-e sib].
Ali I SPCF persuade did-3S to [eat-INF-EZ apple].
'Ali persuaded me to eat (not any other action) the apple.'

(3.27) ali man ră majbur be [sib xord-an] kard.
Ali I SPCF persuade to [apple eat-INF] did-3S.
'Ali persuaded me to eat apples.'

majbur kardan	to persuade, to cause
vădăr kardan	to compel

Table 3.3: Object Control

The above are instances are clear examples of object control, because the matrix object is bound with the subject of the embedded clause (whether finite or non-finite). It is not possible for the embedded clause to have another subject that is not bound with the matrix object (for the case of object-control in *persuade*.

possible for another NP Note that in (3.26) the extraposing of the non-finite clause is difficult and there must be some change of intonation for the sentence to be grammatical. In the last example *sib* (i.e. apple) comes before the non-finite verb and has an indefinite and generic meaning.

to promise

(3.28) ali be man (in_i ra) qol dăd [ke sib ră be-xor-ad]$_i$.
Ali to I (this$_i$ SPCF) promise gave [that apple SPCF SUB-eat-3S]$_i$
'Ali promised me to eat the apple.'

(3.29) * ali be man in_i ră [ke sib ră be-xor-ad]$_i$ qol dăd.
Ali to I this$_i$ SPCF [that apple SPCF SUB-eat-3S]$_i$ promise gave.
'Ali promised me to eat the apple.'

(3.30) ali be man [xord-an-e sib ra] qol dăd.
Ali to I [eat-INF-EZ apple SPCF] promise gave.
'Ali promised me to eat the apple.'

(3.31) ? ali be man qol dăd [xord-an-e sib ra].
Ali to I promise gave [eat-INF-EZ apple SPCF].
'Ali promised me to eat the apple.'

Notice that extraposing of the finite clausal argument is obligatory in this case - unlike example (3.24). As with *majbur kardan* (i.e. to persuade) we can also form an indefinite reading for apple by putting it before the verb in the non-finite clause.

to expect

(3.32) ali (in$_i$ ra) entezăr (az man) dărad [ke sib ră bexor-am]$_i$.
Ali (this$_i$ SPCF) expectation (from me) have [that apple SPCF SUB-eat-1S]$_i$.
'Ali expects me to eat the apple.'

(3.33) ? ali in$_i$ ră [ke sib ră bexor-am]$_i$ (az man) entezăr dărad.
Ali this$_i$ SPCF [that apple SPCF SUB-eat-1S]$_i$ (from i) expectation have.
'Ali expects me to eat the apple.'

(3.34) ali [xord-an-e sib ra] (az man) entezăr dărad.
Ali [eat-INF-EZ apple SPCF] (from i) expectation have.
'Ali expects me to eat the apple.'

to hope

(3.35) ali (be in$_i$) omidvar ast [ke sib ră be-xor-am]$_i$.
Ali (to this$_i$) hopeful is [that apple SPCF SUB-eat-1S]$_i$.
'Ali hopes that I eat the apple.'

(3.36) ali be in [ke sib ră be-xor-am] omidvar ast.
Ali to this [that apple SPCF SUB-eat-1S] hopeful is
'Ali hopes that I eat the apple.'

(3.37) a. ali be [sib xord-an-e hasan] omidvar ast.
Ali to [apple eat-INF-EZ Hasan] hopeful is
'Ali hopes that Hasan eat=apple.'

b. * ali be [hasan xord-an-e sib] omidvar ast.
Ali to [Hasan eat-INF-EZ apple] hopeful is
'Ali hopes that Hasan eat apples.'

(a) shows that the subject can only be expressed if the object comes immediately before the infinitive. The infinitive can only mark a bare noun to its immediate left and another to its right. The object can go to both positions. But if it appears in the left position we will get an indefinite/generic reading for the object of the clausal argument e.g. (a).

A subject can only appear in the right position. Note that it is also possible for the infinitive to be modified by an adverb. An example of this was shown in (3.15), repeated here as (3.38). This is an example of non-obligatory control in Persian that is proposed by Hashemipour for Persian finite clauses. If one does not express the subject, it will be automatically bound by the matrix controller. We have illustrated this for the non-finite example and the finite example is derived similarly.

Note that [Hashemipour, 1989] does not discuss non-finite control. The corresponding non-finite examples better illustrate the notion of control[9].

(3.38) nasihat=kardan-e ziyad-e ali
advice=giving-EZ very-EZ Ali
'Giving too much advice to Ali' 'Ali's giving too much advice'

In (3.39) if we leave out the subject of the non-finite clause, it will be interpreted as controlled by Ali. This is an example of control in Persian non-finite clauses.

(3.39) ali be [sib xord-an] omidvar ast.
Ali to [apple eat-INF] hopeful is
'Ali$_i$ hopes that he$_{i/*j}$ will eat apple.'

to try

(3.40) ali (? barăy-e in$_i$) say=kard [ke sib ră be-xar-ad]$_i$.
Ali (for-EZ this$_i$) try=did [that apple SPCF SUB-buy-3S]$_i$.
'Ali tried to buy the apple.'

(3.41) ali barăy-e in [ke sib ră be-xar-ad] say=kard.
Ali for-EZ this [that apple SPCF SUB-buy-3S] try=did.
'Ali tried to buy the apple.'

(3.42) ali barăy-e [xarid-an-e sib] say=kard.
Ali for-EZ [buy-INF-EZ apple] try=did.
'Ali tried to buy the apple.'

[9]In this regard the control phenomenon in Greek differs the same phenomenon in Persian. Control in Persian subjunctive finite clauses may be derived from non-finite ones, this needs further research.

As we have shown in the examples of this section, the control phenomenon in Persian appears in both finite and non-finite clauses. In other languages, control normally exists only in non-finite clauses.

3.4 Structure of Clausal Arguments

Based on the discussion on control phenomenon for tensed and untensed clauses of Persian we propose the following structure in Figure (3.1) for Persian tensed clausal arguments. In this structure if the clausal argument is extraposed, the NP (i.e. *in*) may be absent, otherwise it must be present.

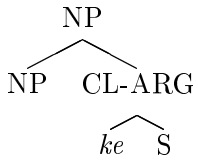


Figure 3.1: A Structure for Persian Tensed Embedded Clauses

We can represent the structure of the Persian untensed embedded structures as in Figure (3.2). In this diagram, the position corresponding to *in* in Figure (3.1) is empty.

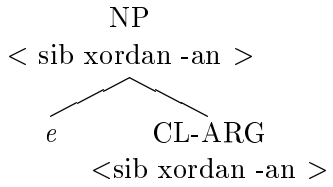


Figure 3.2: Structure for Untensed Clausal Arguments

Based on these two structures, we can represent both tensed and untensed clausal structures by the general structure of Figure (3.3). In this structure we treat tensed and untensed clauses of Persian in parallel to each other.

In this structure:

1. If the clausal argument is non-finite (not tensed) then the clausal argument is not

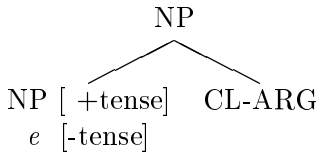


Figure 3.3: General Structure for Persian Clausal Arguments

dominated by a noun phrase (NP) (i.e. the place is empty or e).

2. If the clausal argument is finite, then extraposition is possible and the dominating NP argument is normally empty.

3. If the clausal argument is finite but it is not extraposed, then it must be dominated by an NP.

Note that as we explained in Persian, the non-finite clauses behave as NPs, but we do not imply that [+tense] is a feature of NP in the above figure only clausal arguments have [tense] feature.

The structure which we described is in line with the proposal of Moyne and Carden [Moyne and Carden, 1974] for clausal arguments in Persian:

- Sentential arguments originate in pre-verbal position in Persian.
- They are dominated by an NP.
- They are moved to the post-verbal position by an obligatory extraposition rule.

But as Soheili-Isfahani [1976] and others have shown, the extraposition is not always obligatory.

In (3.43) we see an example of subject complement.

(3.43) (in$_i$) be-nazar-mires-eh [ke ali sib ră xord-eh ast]$_i$.
(this$_i$) is-seeming [that Ali apple SPCF eaten is-3S]$_i$
'It seems that Ali has eaten the apple.'

Note that the extraposition is obligatory and (3.44) is not grammatical.

(3.44) * in [ke ali sib ră xord-eh ast] be-nazar-mires-ad.
this [that Ali apple SPCF eaten] is-3S is-seeming.
'It seems that Ali has eaten the apple.'

In the above examples *in* is a noun phrase which co-refers with the extraposed subject complement[10]. In fact *in* in Persian (i.e. this) can be considered as a kind of expletive like *it* in English, but in Persian it can be further modified by other nouns, and as we have shown, is not obligatory.

In (3.45) we see an example of object complement.

(3.45) ? [in haqiqat]$_i$ [ke irăq be Iran hamleh kard]$_i$ ră hame mi-dăn-and.
this fact [that Iraq to Iran invasion did] SPCF all CN-know-3S
'Every one knows this fact that Iraq invaded Iran.'

(3.46) [in haqiqat]$_i$ ră hame mi-dăn-and [ke irăq be iran hamleh kard]$_i$.
this fact SPCF all CN-know-3S [that Iraq to Iran invasion did].
'Everyone knows this fact that Iraq invaded Iran.'

Note that in the previous examples the non-extraposed (center-embedded) examples are not used, while the following type of center-embedded complement clause is often used.

(3.47) hame az [in haqiqat]$_i$ [ke irăq be Iran hamleh kard]$_i$ ăgăh hastan-and.
all from this fact [that Iraq to Iran invasion did] aware be-3S
'Everyone is aware that Iraq invaded Iran.'

(3.48) hame az [in haqiqat]$_i$ ăgăh hast-and [ke irăq be iran hamleh kard]$_i$.
all from this fact aware be-3S [that Iraq to Iran invasion did]
'Everyone is aware that Iraq invaded Iran.'

The examples show that the extraposition is obligatory for subject complements, while it is optional for the complements other than subject and object complements; that is complement clauses preceded by a preposition. Note that object complements are extraposed most of the time. Soheili-Isfahani [1976] states that the non-extraposed examples (center-embedded)

[10]This is an instance of noun complement structure in Persian.

are difficult to comprehend. According to Soheili Isfahani center-embedding reduces comprehensibility and this may be related to a limitation on the human capacity of temporary memory.

3.5 Relative Clauses

So far we have considered clausal arguments of Persian. Here we will discuss relative clauses of Persian. In Persian, NPs (whether marked by a preposition or not) can be further modified by relative clauses. These relative clauses normally come immediately after the NP which they modify :

(3.49) mard-i/*mard [ke did-i] sib ră xord.
man-REL [that saw-2S] apple SPCF ate-3S
'A/The man whom you saw ate the apple.'

(3.50) mard sib-i/*sib ră [ke did-i] xord.
man apple-REL SPCF [that saw-2S] ate-3S
'The man ate the apple you saw.'

(3.51) man be madrese-i [ke ali mi-rav-ad] raft-am.
I to school-REL [that Ali CN-go-3S] went-1S.
'I went to the school that Ali is going.'

These relative clauses are always marked by a clause marker *ke* that comes at the beginning of the relative clause. The modified noun phrase is usually marked by a relative marker *-i* at the end. In general *-i* is an indefinite marker which doesn't appear after definite nouns. More specifically this marker is a specific indefinite marker [Shariat, 1971] and as Peterson [1974] has discussed, it functions in a similar way to a/some in English [Soheili-Isfahani, 1989]. But here this suffix is required on the head of a restrictive relative clause, but not on the head of a non-restrictive relative clause [Comrie, 1981].

3.5.1 Restrictive/Non-restrictive Relative Clauses

Relative clauses of Persian are categorised into restrictive (or attributive) and non-restrictive (or descriptive) ones. Example (3.51) shows an example of a restrictive relative clause. The relative clause *ke Ali mi-rav-ad* serves to delimit the potential referents of *madrese* (i.e. school): the speaker assumes that the sentence *man be madrese raft-am* does not provide the hearer with sufficient information to identify the school (the hearer would probably have to ask "which school?"), so the additional information as a relative clause is added to indicate specifically which school is being talked about. An example of non-restrictive relative clauses of Persian is shown in (3.52).

(3.52) ali [ke midănest ăn ră] săket mănd.
Ali [that knew that SPCF] silent remained-3S
'Ali, who knew it remained silent.'

In this sentence, it is assumed by the speaker that the hearer can identify which man is being talked about, and that it is one particular, identifiable *Ali* that is being talked about, and the relative clause serves merely to give the hearer an added piece of information about an already identified entity, but not to identify that entity. (3.53) shows another instance of a non-restrictive relative clause [Comrie, 1981].

(3.53) moallef [ke nevisande-ye xub -ist] in sabk ră extiyăr=kardeh-ast.
author [that writer good is] this style SPCF has-chosen.
'The author, who is a good writer, has chosen this style.'

As we showed in Persian, in addition to the difference between restrictive and non-restrictive relative clauses in terms of semantic or pragmatic terms, there is a formal distinction (i.e. -i marker) between them. In addition to these there is also an intonational distinction between these two types of relative clauses. The interested reader can see [Soheili-Isfahani, 1989]. In the following we will discuss the binding of empty categories and pronouns in relative clauses.

3.5.2 Binding in Relative Clauses

In relative clauses there is always an empty category or a resumptive pronoun (or a clitic) which co-refers with the head noun of the relative clause. In Persian the obliqueness hierarchy

plays an important role in introducing a resumptive pronoun in place of the empty category [Soheili-Isfahani, 1989]. If the subject of the relative clause is an empty category and is unified with the head noun, a pronoun does not appear:

(3.54) sib-i$_i$ [ke e_i/(? ăn) injă bud] xordeh shod.
apple-REL$_i$ [that e_i/(? that) here was] eaten became.
'The apple that was here was eaten.'

If the direct object in the relative clause is an empty category unifying with the head noun, we can optionally replace the object empty category with a pronoun.

(3.55) sib-i$_i$ [ke man e_i/(ăn ra) xord-am] sabz bud.
apple-REL [that I e_i/(that SPCF) ate-1S] green was
'The apple I ate (it) was green.'

In other cases where the head noun should unify with an empty category which is dominated by a preposition or another noun then the empty category is obligatorily replaced by a pronoun.

(3.56) man ketab-i$_i$ ră [ke donbăl-e ăn$_i$ /* e_i bud-am] peydă=kard-am.
I book-REL SPCF [that after-EZ that$_i$ / e_i was-1S] found=do-1S
'I found the book that I was looking for.'

3.5.3 Extraposition of Relative Clauses

As in German, a relative clause can be extraposed to the end of the clause[11]. As we explained the relative marker *-i* can often be used to mark the NP which the extraposed clause modifies.

(3.57) mard-i sib ră xord [ke did-i].
man-REL apple SPCF ate-3S [that saw-2S]
'The man (whom) you saw ate the apple.'

(3.58) mard sib-i ră xord [ke did-i].
man apple-REL SPCF ate-3S [that saw-2S]
'The man ate the apple that you saw.'

[11] In Persian only restrictive relative clauses can be extraposed.

(3.59) sib-i ră [ke did-i] mard-i xord [ke injă bud].
apple-REL SPCF [that saw-2S] man-REL ate-3S [that here was]
'A/The man who was here ate the apple that you saw.'

(3.60) * sib-i ră mard-i xord [ke did-i] [ke injă bud].
apple-REL SPCF man-REL ate-3S [that saw-2S] [that here was]
'A/The man who was here ate the apple that you saw.'

(3.61) mard-i [ke ali did-(ash)] injă bud.
man-REL [that Ali saw-(him)] here was
'A/The man whom Ali saw was here.'

(3.62) mard-i [ke ali goft [ke did-ash]] injă bud.
man-REL [that Ali told [that saw-him]] here was
'A/The man whom Ali said he has seen him, was here.'

The interaction between extraposition of relative clauses and clausal arguments is an interesting issue in Persian which sheds light on the actual position of clausal arguments. If we assume that the clausal arguments are base generated post verbally then it shouldn't be possible for embedded clauses to appear between verbs and clausal arguments. But this is not the case, and an example of this is shown in (3.63).

(3.63) ali be mard-i$_j$ goft [ke injă bud]$_j$ [be-rav-ad xaneh]$_k$.
Ali to man-REL told [that here was] [SUB-go-3S home]
'Ali told to the man who was here to go home.'

(3.64) ali be mard-i$_j$ goft [be-rav-ad xaneh]$_k$ [ke injă bud]$_j$.
Ali to man-REL told [SUB-go-3S home] [that here was].
'Ali told to the man who was here to go home.'

These examples further support the proposal of extraposition of clausal arguments in Persian. In extraposition, an embedded clause is moved to a place after the right boundary of the embedding clause. If this position is already filled by another extraposed relative clause then it is not possible to extrapose other relative clauses. In other words there is only one position available for relative clauses in the post-verbal position in Persian.

3.6 Fronting and Scrambling

In Persian, there are examples of leftward movement from embedded clauses into main clauses. In this section, after reviewing some examples of this movement we will argue for two different types of movements.

Before going into the details of fronting and scrambling, we will first show instances of movement from embedded clauses in Persian.

3.6.1 Examples of Fronting in Clausal Arguments

The examples of embedded clauses – clausal arguments and relative clauses – which we presented in the previous sections don't have any instances of fronting in them. In fronting, a category from an embedded clause is moved to the domain of the clause which dominates it. In this section we will review examples of fronting for the sentences we saw earlier.

to expect

In (3.65) an example of fronting is shown. This sentence corresponds to the non-fronted example of (3.66). Note that *sib* does not belong to the subcategorisation frame of the matrix verb.

(3.65) ali *sib* *ră* entezăr (az man) dărad [ke __ bexor-am].
Ali *apple* *SPCF* expectation (from me) have [that __ eat-1S].
'Ali expects me to eat the apple.'

(3.66) ali entezăr (az man) dărad [ke sib ră bexor-am].
Ali expectation (from me) have [that apple SPCF eat-1S].
'Ali expects me to eat the apple.'

to promise

(3.67) corresponds to the non-fronted example of (3.68).

(3.67) ali be man *sib ră* qol dăd [ke __ be-xor-ad].
Ali to I *apple SPCF* promise gave [that __ SUB-eat-3S]
'Ali promised me to eat the apple.'

(3.68) ali be man qol dăd [ke sib ră be-xor-ad].
Ali to I promise gave [that apple SPCF SUB-eat-3S]
'Ali promised me to eat the apple.'

to hope

(3.69) corresponds to the non-fronted example of (3.70).

(3.69) ali *sib ră* omidvar ast [ke __ be-xor-am].
Ali *apple SPCF* hopeful is [that __ SUB-eat-1S]
'Ali hopes that I eat the apple.'

(3.70) ali omidvar ast [ke sib ră be-xor-am].
Ali hopeful is [that apple SPCF SUB-eat-1S]
'Ali hopes that I eat the apple.'

to try

(3.71) ali *sib ră* say=kard [ke __ be-xar-ad].
Ali *apple SPCF* try=did [that __ SUB-buy-3S].
'Ali tried to buy the apple.'

to think

(3.72) u *sib ră* fekr=kard [ke če-tor ahmad __ xord-eh ast].
He *apple SPCF* thought=do-3S [that what-way Ahmad __ eat-en is-3S]
'He wondered how Ahmad has eaten the apple.'

to say

The fronting phenomenon is not restricted to the direct object case and it is possible to front different kinds of categories. In the following, a few of them for the verb *goftan* (i.e. to tell/say) are shown.

(3.73) u *ahmad* *rā* goft [ke __ sib rā be-xor-ad].
he *Ahmad* *SPCF* told-3S [that __ apple SPCF SUB-eat-3S]
'He said that Ahmad eat the apple.'

(3.73) shows an instance of subject fronting.

(3.74) u *šir* *rā* goft [ke u (ăn ra) be-xor-ad].
he *lion/milk* *SPCF* told-3S [that he that SPCF SUB-eat-3S]
'He said (to s.o.) that he eat the milk/lion.'

(3.74) depicts an instance of object fronting.

(3.75) u *madrese* *rā* goft [ke ahmad (be ănja) be-rav-ad].
he *school* *SPCF* told-3S [that Ahmad (to there) SUB-go-3S]
'He said (to ?) that Ahmad go to school.'

(3.76) u *ahmad* *rā* fekr=kard [ke če-tor __ sib rā xord-eh ast].
he *Ahmad* *SPCF* thought=do-3S [that what-way __ apple SPCF eat-en is-3S]
'He wondered how Ahmad has eaten the apple.'

(3.75) is an instance of fronting where the fronted category is not subject or object. It is a direction or location.

In the above examples we showed that the fronted category is marked with *rā* and this is usually the case. In fact some have suggested that *rā* is a topic marker.

3.6.2 Is Fronting a Case of NP Left-Dislocation

NP-left dislocation is a possible way for an Ezafe (NP) construct to be extracted from inside a NP or PP in order to be preposed to the clause. The preposed NP leaves a resumptive pronoun *-sh* which is cliticised to its governor:

(3.77) diruz ketab-e hasan gom shod.
yesterday book-EZ Hasan lost became-3S
'Yesterday Hasan's book was lost.'

(3.78) diruz hasan ketab-esh gom sho-d.
yesterday Hasan book-CLITIC lost became-3S
'Hasan, yesterday his book was lost.'

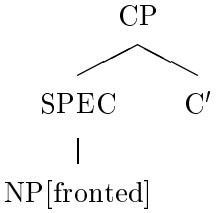
(3.79) hasan ră tup ră az-ash gereft-am.
Hasan SPCF ball SPCF from-CLITIC caught-1S
'Hasan, I caught the ball from him.'

Note that the left-dislocated noun phrase always co-refers with a clitic and conveys old information. This suggests that the phenomenon is a topicalisation process. There exists an analogous phenomenon in Arabic. Consider these examples:

(3.80) ali-un$_i$ dharab-tu akh-a-hu$_i$.
Ali-NOM beat-1S-MASC brother-ACC-his
'Ali, I beat his brother.'

(3.81) ali-un$_i$ akh-o-hu$_i$ dhahaba.
Ali-NOM brother-NOM-his went-3S-Mas.
'Ali, his brother went.'

The examples of NP left-dislocation can be represented by this structure, in which the left-dislocated NP goes to the SPEC position:



In Arabic, the left-dislocated noun phrase receives nominative case. Note that the nominative case marker in Arabic is also a topic marker and in the above examples topic-marks[12]

[12]The term used in Arabic for topic is *mobtadă* (i.e. fronted). There is a corresponding notion *khabar* for comment.

Ali. This may suggest that in Persian *ră* is also a topic marker, but *ră* cannot appear after a topicalised noun phrase that has been extracted from subject positions:

(3.82) a. ali, madrese-ash xarăb shod.
Ali, school-his demolished become-3S
'Ali, his school was demolished.'

b. * ali ră, madrese-ash xarăb shod.

(a) shows that *ră* does not appear after all topics and does not appear after nominals extracted from subjects[13] Note that *ră* is obligatory after noun phrases topicalised from non-subject phrases.

Consider the following example in which *gušt* has been moved from the embedded clause into the matrix clause.

(3.83) man *gušt (ră)* goft-am [ke __ na-xor-d].
I *meat SPCF* said-I [that __ NOT-eat-3S]
'The meat, I told him not to eat.'

An analogous phenomenon does exist in Arabic.

(3.84) qolto la-hu [an la ya'kola al-lahm-a].
told to-him that not eat the-meat-ACC.
'I told him not to eat the meat.'

(3.85) al-lahm-o$_i$, qolto la-hu [an la ya'kola-hu$_i$].
the-meat-NOM, told to-him that not eat-it
'The meat, I told him not to eat.'

Note that in the Arabic example (1) the topicalised noun phrase moves to the initial position and (2) it leaves a pronoun/clitic in its place inside the matrix clause. Neither of these is required for the examples of fronting we studied. In (3.86) *gušt* can appear anywhere in the matrix clause and it does not leave a pronoun in its initial position inside the embedded

[13]Such examples have motivated some to argue that *ră* is a secondary topic marker and as a result does not appear after primary topics (i.e. subjects).

clause. In fact in (3.86) the sentence does not sound grammatical when there is a pronoun inside the phrase co-referring with the moved element. In addition *ră* is not obligatory after the fronted noun phrase. These facts clearly distinguish fronted noun phrases from non-subject topicalised NPs. In fact some examples of the fronted constituents carry new information such as contrast which is against the assumption that they are topics.

(3.86) *gušt* man goft-am [ke (? an$_i$ ra) na-xor-d].
meat I told-1S [that it SPCF NOT-eat-3S]
'The meat, I told him not to eat it.'

In passing it should be noted that examples of local scrambling inside a matrix clause which we studied in the previous chapter, give the speaker the opportunity to choose the appropriate order of constituents in the context. The tendency is to put a topic phrase at the beginning of a sentence to link to the previous discourse.

(3.87) be madrese ki raft?
to school who went?
'who went to the school.'

3.6.3 Is Fronting Leftward Movement?

Having shown that the fronting examples are not instances of NP left-dislocation/topicalisation in Persian, the second possibility is for them to be instances of some other kind of leftward movement.

But if this is the case, we must answer the question why it is not possible to front an element from the clausal argument when *in* is present in the main clause:

(3.88) ? ali in$_i$ ră say=kard [ke sib ră be băzăr be-bar-ad]$_i$.
Ali this SPCF try=did [that apple SPCF to Bazar SUB-take-3S].
'Ali tried to take the apple to the bazaar.'

(3.89) * Ali *sib ră* in$_i$ ră say=kard [ke __ be băzăr be-bar-ad]$_i$.
Ali *apple SPCF* this SPCF try=did [that __ to Bazar SUB-take-3S]
'Ali tried to take the apple to the bazaar.'

Such proposals as Karimi [1990] that consider the movements as instances of leftward movement face problems and cannot explain the presence of *in* and the blocking of movement.

If fronting is an example of leftward movement, then how is it possible to have instances of verbs (such as 'gotan' to say) where we can front instances of noun phrases of the embedded clause easily; and why is it difficult to do fronting in verbs such as 'majbur=kardan' in (3.91) which have their own indirect objects?

(3.90) ali *sib ră* majbur kard [ke __ be-xor-am].
Ali *apple SPCF* persuade did [that __ SUB-eat-1S]
'Ali persuaded me to eat the apple.'

In (3.90) an example of fronting is shown. This sentence corresponds to the non-fronted example of (3.23). Here *sib* does not belong to the subcategorisation frame of the matrix verb.

Note that the following example, under normal intonation is not grammatical:

(3.91) * ali [ki ră] *[sib ră]* majbur kard [ke __ be-xor-ad]?
Ali who SPCF *apple SPCF* persuade did [that __ SUB-eat-3S]
'Whom did Ali persuade to eat the apple?'

In fact there are more complex examples of the so-called fronting phenomenon which we haven't mentioned, examples such as (3.92)-(3.93) where we have in addition to the fronted noun phrase instances of one or two prepositional phrases which are also scrambled into the matrix clause. These examples create a further problem for the left movement approach to fronting, because in most approaches there is a single position considered for this kind of fronting and fronting more than one element creates problems.

(3.92) ašăyer *[gosfandhă ră] [be kojă]* say=kard-and [__ __ be-bar-and]?
nomads *[sheep SPCF] [to where]* try=did-3P [__ __ SUB-take-3S]
'Where did the nomads try to take the sheep?'

(3.93) ašăyer [gosfandhă ră] [az yeylăq] [be qešlaq] say=kard-and [__ __ __ be-bar-and].
nomads [sheep SPCF] [from yeylăq] [to qešlaq] try=did-3P [__ __ __ SUB-take-3S].
'The nomads tried to take the sheep from summer pasture to winter pasture.'

We should also note that the case marking of the fronted noun phrase is not necessarily the same as in the embedded clause. In (3.73) repeated here as (3.94) the fronted subject of the embedded clause is marked by *ră* in the matrix clause. As we said *ră* does not appear with subject phrases.

(3.94) u *ahmad ră* goft [ke __ sib ră be-xor-ad].
he *Ahmad SPCF* told-3S [that __ apple SPCF SUB-eat-3S]
'He said that Ahmad eat the apple.'

These facts clearly show that fronting is not an example of leftward movement either.

3.6.4 Previous Formal Approaches to Fronting

In the following, we will review two formal approaches for representing some instances of fronting in Persian, and then we will propose our solution for accounting for examples of NP movement in Persian.

First we will review Karimi's proposal for *ră* and her proposed *Case Tendency Principle* for capturing fronting. Then we will discuss Yoon's proposal for representing examples of long distance NP movement from Persian subject complement clauses.

Fronting and Case Tendency Proposal

Karimi [1990] in a GB framework, proposes that *ră* in Persian is a specific oblique marker and obligatorily case marks a noun phrase if that noun phrase is specific and is oblique. She argues that a noun phrase is oblique if it is not in the minimal government-projection of a noun or an adjective or a preposition. In other words she considers a noun phrase oblique, if its case is not nominative [-NOM] (i.e. it is not subject) and it is not preceded by a preposition.

She further revises the case assignment principle for Persian. Under the revised version[14]:

- a. INF assigns [+NOM] case to the subject NP under agreement.
- b. V and Prep assign [-NOM] case to the object NP.
- c. The EZAFE particle transfers the case of the head noun to its complements.

[14]The previous version is augmented by the (c) case.

By this simple solution, she captures many instances of the function of *ră* in Persian in a principled way. We will elaborate on some of these[15] [Karimi, 1990].

Specific direct object marking

Consider this example:

(3.95) a. man in ketab ră dida-am.
I this book SPCF saw-1S
'I saw this book.'

b. * man in ketab dida-am.
I this book saw-1S
'I saw this book.'

For this sentence to be grammatical, *ră* must appear after *in ketab*. According to Karimi's proposal, since *in ketab* is specific and is the direct object of the sentence (i.e. [-NOM] and accusative marked) so it is both oblique and specific and must be case marked by *ră*.

Occurrence of *ră* with arguments that are not direct objects

ră co-occurs with noun phrases that are not direct objects:

(3.96) man ră beh-em mi-khand-e.
me SPCF to-me Cont-laugh-3S
'As for me, she laughs at me.'

According to Karimi's proposal, here *man* is specific and oblique and *ră* must appear after it. *man* is specific because it is a pronoun and all pronouns are specific[16]. *man* is oblique because it is co-indexed with *-em* and inherits the [-NOM] case of *-em*. Also it is not governed by a preposition.

Double occurrence of *ră* in a sentence

It is possible for *ră* to appear twice in a sentence. This is shown in (3.97):

[15] We will discuss in (3.97) the need for the c part.

[16] Note that *man* is also co-indexed with the clitic *-em* in this sentence and by co-indexation can also get the specificity of -em which is always a pronoun and as a result specific.

(3.97) mășin ră dăr-esh ră bast-am.
car SPCF door-its SPCF closed-1S
'As for the car, I closed its door.'

Here *măšin* is oblique for the same reason that we mentioned for *man* in the previous case, and it is also specific, because it is co-indexed with a pronoun. So, according to Karimi, it must be marked by *ră*. Similarly *dăr-esh* is specific and is the direct object of the verb (i.e. oblique) so it must also be marked with *ră*.

Karimi's proposal successfully represents examples where *ră* shouldn't appear. For example in the following example since *măšin* and *dăr-esh* are co-indexed, they have the same [+NOM] case. As a result *ră* shouldn't appear after either of them, although both are also specific.

(3.98) măšin, dăr-esh baz ast.
car door-its open is-3S
'As for the car, its door is open.'

According to the proposal of Karimi, specificity and obliqueness together are the necessary and sufficient conditions for *ră* marking. She tries to capture all possible functions of *ră*, but there are examples that her proposal does not capture very elegantly. Among these are examples of nominal time and place adverbs, after which under certain conditions *ră* might occur or not.

(3.99) emšab (ră) injă mi-xăb-im.
tonight SPCF here CONT-sleep-1P
'As for tonight, we will sleep here.'

It is not clear why emšab, that does not bear [+NOM], and can receive oblique case from the oblique case assigner verb[17], does not always get marked by *ră*, although it is not governed by any head either.

In her proposal, Karimi does not consider examples of di-transitive verbs and the case marking of the object and the object complement in these sentences:

[17]Karimi distinguishes between pure transitive verbs and oblique case assigners [Karimi, 1990]. She assumes that some transitive verbs can assign oblique case.

(3.100) mă bače ră ali seda=mikon-am.
we child SPCF Ali call=do-1P
'We call the child Ali.'

(3.100) shows an example of such a sentence. Note that the object complement *ali* is specific, but is not case marked by *ră*. In general, in di-transitives the object is obligatorily marked with *ră*, and the object complement obligatorily precedes the verb.

Karimi in her work does not elaborate much on examples of long distance topicalisation. But she gives examples that support the case marking of the fronted category inside its present clause.

(3.101) *gušt* behtar-e beg-i [__ na-xor-d].
meat better-is tell-2S [__ NOT-ate-3S]
'As for meat, it is better to tell him/her not to eat.'

In (3.101) she assumes that *gušt* is case marked by the verb *beg-i* and again she considers the verb of the sentence, an instance of a oblique assigner verb. For representing this and also the phenomenon of attraction in Persian relative clauses, she proposes the Case Tendency principle for Persian.

(3.102) The Case Tendency
The case of a non-argument NP tends to be determined by its position in the CP containing it, or the closest CP.

But what are the underlying formal principles for case tendency in Persian? Karimi does not discuss this.

Fronting and A-SPEC Proposal

Yoon [1992] discusses some interesting properties of finite raising in some languages and also discusses subject complement clauses and movement from them in Persian. He argues that movements (or raising in his terminology) from subject complement clauses are examples of A-movement and not A′-movement[18] because:

[18] In A-movement, a phrase is moved to an argument position like subject (i.e. A-position) that is associated with a grammatical function, while in A′-movement, the phrase is moved to a non-argument like adjunct position (i.e. A′-position). This is a simplified definition, for further details see Haegeman [1994].

- Idiom chunks can be raised. As seen in (3.103)

  (3.103) sar-e ali lăzem ni-st [ke kolăh gozasht-e be-šav-ad].
  Head-of Ali necessary NEG-be that hat put-PASS SUBJ-inch-3S
  'Ali is not necessary that (he) be ripped off.'

  Here *sar-e S.O. kolăh gozash* is an idiom chunk.

- The raised nominals can bind from the raised position as seen in Karimi's (:18), here repeated as (3.104)

  (3.104) ali barăy-aš lăzem ast [ke har ruz varzeš konad].
  Ali for-him necessary is that every day exercise do-3S
  'It is necessary for Ali to exercise every day.'

- Raised nominals can undergo further raising and passive.

Yoon considers examples where only one of the arguments is scrambled and argues that these kinds of arguments will move to the SPEC position and then to the subject position[19]. But as we will show in (3.130) it is possible to move/raise more than one argument. Hence his assumption of movement of these arguments to an A(rgument)-SPEC position and then to a subject position is not correct. For this he assumes that the SPEC of CP in Persian is an A-position[20].

3.6.5 Our Account of Fronting and Scrambling

In Sections 3.6.2 and 3.6.3, we showed counter evidence for the proposal that the sentential arguments appear canonically post-verbally. Therefore we assume that:

(1) The sentential arguments originate in pre-verbal position in Persian.

(2) They are dominated by an NP.

(3) Fronted constituent moves to SPEC of NP.

(4) CP is moved to the post-verbal position.

[19] He discusses more cases, but we have chosen some of them. The interested reader can see Yoon [1992].

[20] It seems that he assumes that in Persian complement clauses are not dominated by an NP and their canonical position is post-verbal. Earlier we showed counter-examples to this.

Basing our approach on these assumptions, we can easily justify the absence of movement into main clauses in cases where there is a noun phrase co-indexed with the clausal argument.

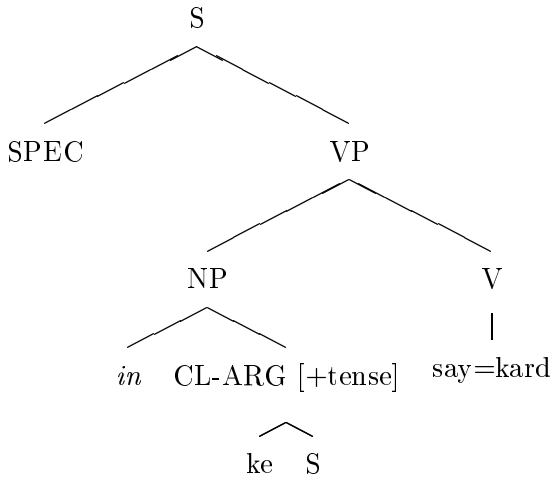
Structural Constraints on Long Distance Scrambling

In example (3.88) repeated as (3.105) there are two bounding nodes in the sentence that prevent the movement of the arguments: one is the tensed clause itself and the other is the dominating noun phrase *in*.

(3.105) ? Ali in_i ră say=kard [ke sib ră be băzăr be-bar-ad]$_i$.
Ali this SPCF try=did [that apple SPCF to Bazar SUB-take-3S].
'Ali tried to take the apple to Bazar.'

In other words for any movement to occur it should pass two bounding nodes, which is generally assumed not to be possible [Ross, 1967], [Chomsky, 1986]. This constraint of movement is known as the subjacency condition.

(3.106) **Subjacency condition**
Movement cannot cross more than one bounding node, where **bounding nodes** are IP and NP.



Put it in another way, when *in* is not present, then there is only one bounding node and subjacency do not prevent the movement of arguments[21] crossing only one bounding node.

[21] In earlier frameworks complex nounphrase constraint (CNPC) would not be violated.

Note that when *in* is not present, the extraposition of the embedded clause is obligatory. We assume that constituents from the clausal arguments may move into the matrix clause before the extraposition happens. After extraposition the clause becomes frozen and no constituent can move from it. This is also true for extraposed relative clauses.

Based on this we can now represent the possible kinds of movements and the constraints on them.

Examples of Long Distance Scrambling

(3.107) shows a simple case of long-distance scrambling in which the NP *gosfandhă* has been moved out of the embedded postverbal clause.

(3.107) ašăyer *[gosfandhă ră]* say=kard-and [__ az yeylăq be qešlaq be-bar-and]
nomads *sheep SPCF* try=did-3P [__ from yeylăq to qešlaq SUB-take-3S]
'The nomads tried to take the sheep from summer pasture to winter pasture.'

It is also possible to scramble more than a constituent out of an embedded clause; however such cases only seem to be fully acceptable if those constituents are PPs[22], as illustrated in (3.108).

(3.108) ašăyer *[az yeylăq] [be qešlaq]* say=kard-and [gosfandhă __ __ ră be-bar-and].
nomads *[from yeylăq] [to qešlaq]* try=did-3P [sheep SPCF __ __ SUB-take-3S].
'The nomads tried to take the sheep from summer pasture to winter pasture.'

Our claim is that the underlying form of (3.108) is the following structure.

(3.109) ašăyer [$_{NP}$ in ră [$_{CP}$ ke gosfandhă ră be qešlaq be-bar-and]] qol=dăd-and.
nomads [this (SPCF) [that sheep SPCF to qešlaq SUB-take-3S]] promise=gave-3P.
'The nomads promised to take the sheep to winter pasture.'

Note that when the embedded clause is in the canonical position, namely preverbally, it forms part of an NP, introduced by *in*. The argument for assigning the category of NP rests on the possibility of marking the whole constituent with the case marking particle *ră* as shown in (3.110)[23]. It is also possible in some cases for the embedded clause to extrapose and *in* remains in situ.

[22] It is also possible to scramble *gosfandhă ră* in (3.108).
[23] See a similar discussion for for relative clauses in Page 78.

(3.110) ašăyer [$_{NP}$in [$_{CP}$ ke gosfandhă ră be qešlaq be-bar-and]] ră qol=dăd-and.
nomads [this [that sheep SPCF to qešlaq SUB-take-3S]] SPCF promise=gave-3P.
'The nomads promised to take the sheep to winter pasture.'

No scrambling is possible out of an embedded clause in preverbal position, as illustrated in (3.111). This can be seen as a violation of subjacency as we explained in Page 63. With the presence of *in* (an NP and a bounding node) it is not possible to cross two bounding nodes.

(3.111) * ašăyer *gosfandhă ră* [$_{NP}$*in*(*ra*)[$_{CP}$ ke __ be qešlaq be-bar-and]] qol=dăd-and.

For the movement to happen, the clause should not be dominated by *in*. (3.112) is also ungrammatical, because the finite clause needs an NP like *in* to dominate it to be able to receive case (or it should be extraposed to be grammatical).

(3.112) * ašăyer [$_{CP}$ ke gosfandhă ră be qešlaq be-bar-and] qol=dăd-and.

For this to become grammatical, the clause should come after the verb as an adjunct CP to the IP node. As it is illustrated in Figure 3.4.

(3.113) ašăyer qol=dăd-and [$_{CP}$ ke gosfandhă ră be qešlaq be-bar-and].

As a result of the extraposition of the embedded clause, the sentence becomes grammatical. This is illustrated in (3.113). The embedded clause moves to an adjunct position and the moved phrase can be properly case marked locally or retain its case through its trace. Any movement out of the embedded clause must happen before the extraposition of the embedded clause. Note that *ră* only appears after specific objects.

(3.114) ašăyer *gosfandhă ră* qol=dăd-and [$_{CP}$ ke __ be qešlaq be-bar-and].

In (3.114), these operations have happened: (1) *in* as a bounding node is deleted, as a result the movements out of the embedded clause can happen, *gosfandhă* is moved out and the embedded clause is extraposed. Since *gosfandhă* is not locally case marked by a preposition, it must be case marked in the new clause and steal the case marking of the deleted *in* .

For the scrambling of the PP (which are always governed and case marked by a preposition) the analysis is simpler.

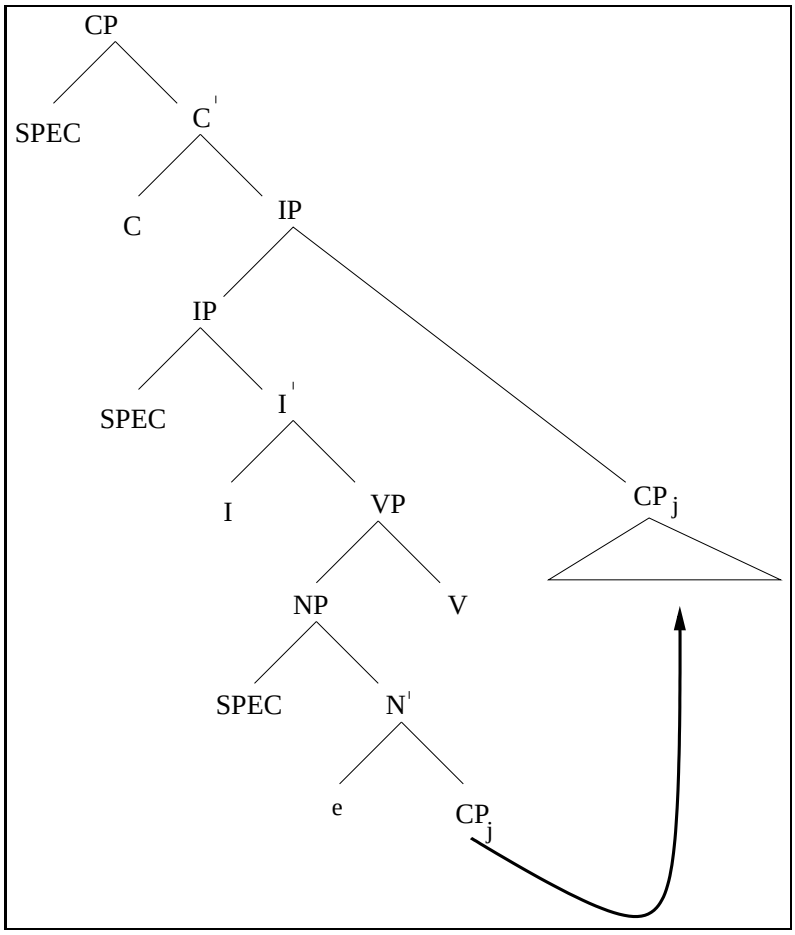


Figure 3.4: Structure After Extraposition of Clause

(3.115) * ašăyer *be qešlaq* [$_{CP}$ ke gosfandhă ră __ be-bar-and] qol=dăd-and.

(3.115) is ungrammatical because the clause must extrapose. The sentence will be grammatical if the clause is extraposed to an adjunct position. E.g. (3.116).

(3.116) ašăyer *be qešlaq* qol=dăd-and [$_{CP}$ ke gosfandhă ră __ be-bar-and].

In (3.116) again first *in* is removed which removes the barrier for the movement of the constituents from the embedded clause. Then the PP is moved out of the embedded clause into the main clause and finally the embedded clause is moved out for the sentence to be grammatical. The PP is locally case marked by the preposition.

(3.117) ašăyer *[be qešlaq] [gosfandhă ră]* qol=dăd-and [$_{CP}$ ke __ __ be-bar-and].

In (3.117) first *in* is removed which removes the barrier for the movement of the constituents from the embedded clause. Then the PP and the NP are moved out of the embedded clause into the main clause and finally the embedded clause is moved out for the sentence to be grammatical. The PP is locally case marked by the preposition, and the NP uses the case marking of the verb for the deleted *in*.

When *in* is present, there is no possibility of long distance scrambling, whether the clause is extraposed or not and the long distance movements are blocked before the extraposition and after it. (3.118) is an example of this. The NP cannot move out of the embedded clause as it was possible in (3.114).

(3.118) * ašăyer *[gosfandhă ră]* [$_{NP}$ in (ra) qol=dăd-and [$_{CP}$ ke __ be qešlaq be-bar-and].

In (3.119), the same is true. The presence of *in* prevents the possibility of movement of the PP out of the embedded clause.

(3.119) * ašăyer *[be qešlaq]* [$_{NP}$ in (ra) qol=dăd-and [$_{CP}$ ke gosfandhă ră __ be-bar-and].

The combination of the NP and PP movements in (3.119) and (3.118) also is ungrammatical for the same reasons.

In our analysis we assume that the scrambling of PPs are instances of adjunct attachment (A′ movement). As a result we can see one or more instances of PP long distance scrambling in Persian.

(3.120) ašăyer *[az yelaq] [be qešlaq] [gosfandhă ră]* qol=dăd-and [$_{CP}$ ke __ __ __ be-bar-and].

In (3.120) first *in* is removed which removes the barrier for the movement of the constituents from the embedded clause. Then the two PPs and the NP are moved out of the embedded clause into the main clause and finally the embedded clause is moved out for the sentence to be grammatical. The PPs are locally case marked by their prepositions, and the NP uses the case marking of the verb for the deleted *in*. This is why only one instance of NP can move. There is only one case to be assigned. In Section 6.4.1 we will elaborate on performance constraints that further constrain these possibilities.

Structure for Long Distance Scrambling

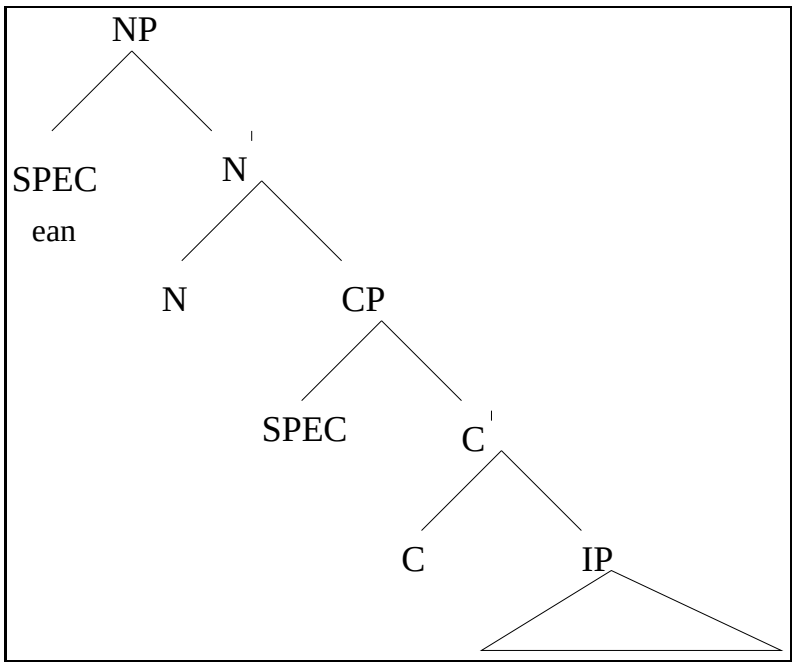


Figure 3.5: A Structure for Clausal arguments in Persian

Based on these observations we propose the structure shown in Figure 3.5 for representing clausal arguments in Persian.

For fronting we assume that the fronted category moves to the SPEC position of the clausal complement (i.e. SPEC of NP). This is shown in Figure 3.6.

Since there is only one SPEC position for each clausal argument, there is only one case of fronting. Like Karimi, we assume that these fronted arguments are case marked by the verb in their new domain. According to her analysis , the fronted categories are inside the domain of the verb and can be case marked by the verb because of *the case tendency principle*. This is shown in Figure 3.7.

Karimi claims that in Persian, the case of a non-argument NP tends to be determined by its position in the CP containing it, or the closest CP [Karimi, 1990]. But Karimi's proposal faces problems in representing sentences like (3.108) where two constituents are scrambled into the main clause. Scrambled constituents always retain their own case marking even in the new clause. In contrast to Karimi we assume that the principle at most can apply to

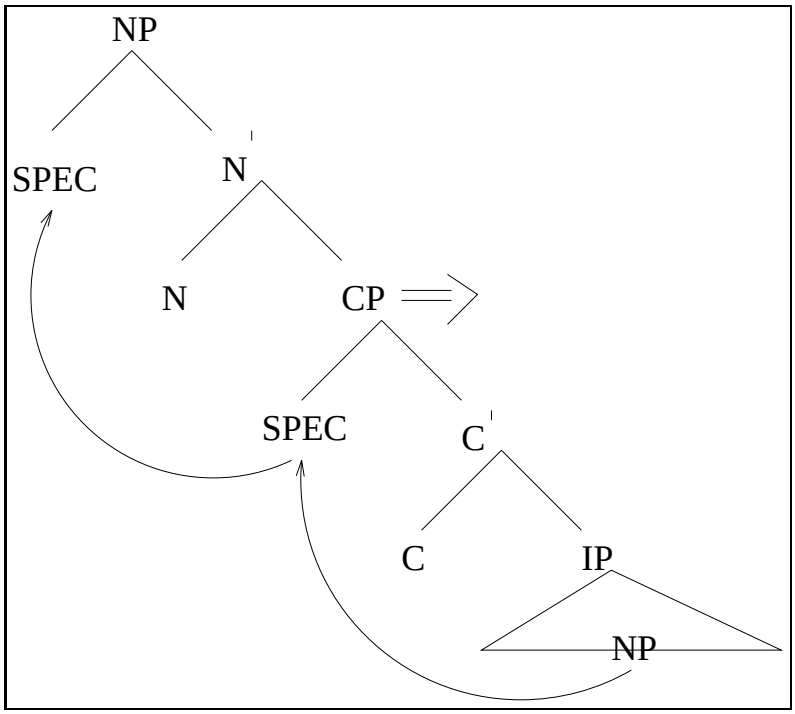


Figure 3.6: A Structure for NP Fronting in Persian

arguments in SPEC position, an A′-position. The position must be case marked in Persian.

Note that unlike Karimi, we don't need to assume that some verbs in Persian are oblique assigners [Karimi, 1990]. Because in our analysis, the oblique[24] case of the absent dominating NP (i.e. *in*) can be assigned to the SPEC of it. In contrast to Karimi, we argued that the clausal arguments originate in pre-verbal positions.

The fronted constituent which is in the SPEC position of the clausal argument, can undergo an NP left dislocation process. This is shown in (3.121).

(3.121) gorbe$_i$ ră man pa-sh$_i$ ră goft-am ke be-bin-id.
cat SPCF I foot-it SPCF told-1S that SUBJ-see-2P
'The cat, I told you to see its foot.'

This structure is shown in Figure 3.8.

And it can also move to higher level clauses:

[24] At the moment we are just concentrating on non-subject complement clauses. We will deal with subject complement clauses later.

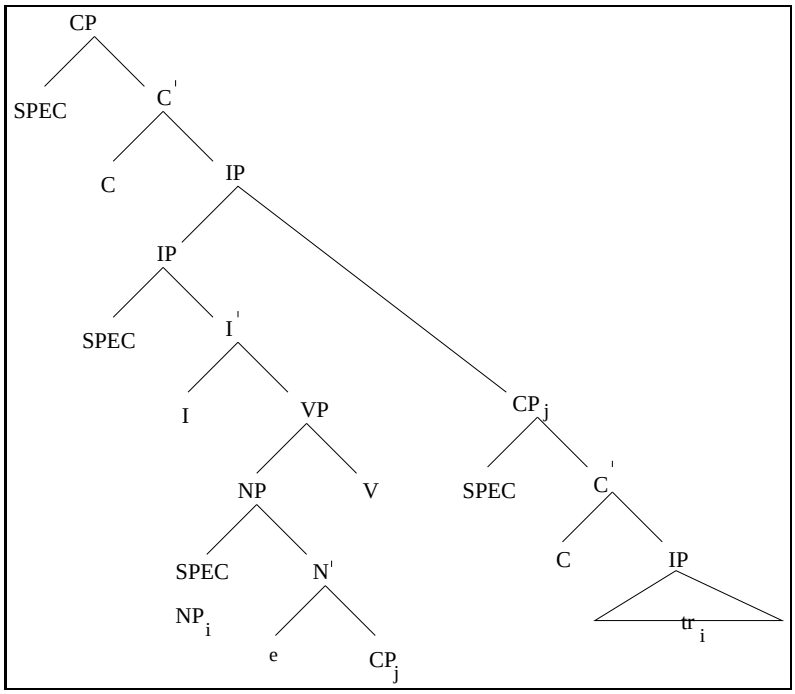


Figure 3.7: Extraposition and NP Fronting in Persian

(3.122) man sib ră goft-am [ke beg-e [ke __ na-xor-ad]].
I apple SPCF told-1S [that tell-3S [that __ NOT-eat-3S]]
'The apple, I said to someone to tell not to eat it.'

The so-called NP fronting phenomenon can be more complex, and we can have instances in which two categories are fronted, but into two different clauses:

(3.123) man *ali ră* goft-am [ke *sib ră* beg-e [ke __ __ na-xor-ad]].
I *Ali SPCF* told-1S [that *apple SPCF* tell-3S [that __ __ NOT-eat-3S]]
'I said to Ali to tell someone not to eat the Apple.'
'I said to someone that tell Ali not to eat the apple.'

In (3.123) the object of the most embedded clause can be fronted into the SPEC position of the higher clause. The subject of the most embedded clause can be controlled by the addressee of the clause one level higher (second translation), or not (first translation). The addressee of this clause is moved to the SPEC position of the main clause.

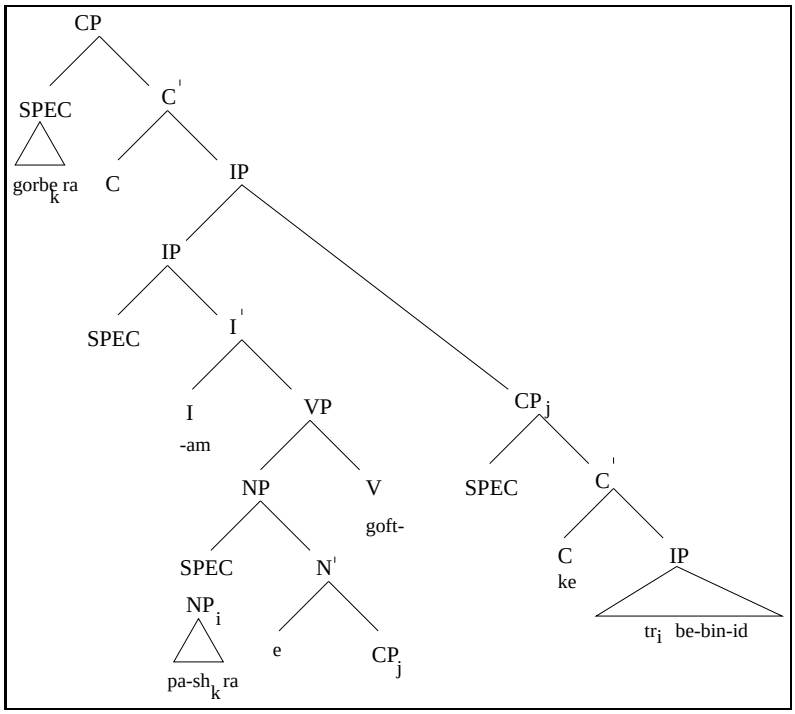


Figure 3.8: Example of Extraposition and NP Fronting in Persian

(3.124) man ali ră goft-am [ke hasan ră bege [ke na-xor-ad]].
I Ali SPCF told-1S [that Hasan SPCF tell-3S [that NOT-eat-3S]]
'I said to Ali to tell Hasan not to eat.'

Note that in (3.124) the sentence has only one interpretation because of the control phenomena. Note that what makes the second sentence have one interpretation is the result of semantic and world knowledge information[25] (I.e. can't eat Hasan).

To elaborate more, we propose that for non-subject clausal arguments we have the following constraints:

1. In the case of fronting, the fronted noun phrase is case marked inside the new clause, but it agrees with its trace in number (and person). A case of weak unbounded dependency.

2. In the case of scrambling , the scrambled noun phrase is not case marked inside the new

[25]Based on this approach a parser has been developed for capturing embedded clauses of Persian [Rezaei, 1993].

clause and it agrees with its trace both in number and case. A case of strong unbounded dependency.

3. Only one of the NPs of the extraposed clause can be fronted and move to the SPEC position of the complement clause in the preverbal position. These are marked by *ră* for non-subject complement clauses.

4. Other NPs of the extraposed clause which scramble into the matrix clause need to be case marked by a preposition.

Now we discuss the fronting in subject clausal arguments in verbs with a modal-like meaning such as *be-nazar resid-an* (seem). (3.125) shows an example of this in Persian where *sib* is being moved:

(3.125) sib ră be-nazar-mires-eh [ke ali xord-eh ast].
apple SPCF is-seeming [that Ali eaten is-3S]
'It seems that Ali has eaten the apple.'

We can extend our analysis for non-subject complement clauses to cover movement examples of subject complement clauses. Based on this we can argue why we cannot have *in* and movement at the same time. As we explained earlier, the presence of *in* acts as a barrier to movement.

(3.126) * *sib ro* in$_i$ be-nazar-mires-eh [ke ali xord-eh ast]$_i$.
apple SPCF this is-seeming [that Ali eaten is-3S]
'It seems that Ali has eaten the apple.'

We claim that our proposal can naturally be extended to cover instances of subject complements where there is no dominating NP. In our analysis we assumed that the verb can case mark the fronted arguments that go into the SPEC. But the SPEC position of subject clausal arguments is out of the domain of the verb of main clause and therefore cannot be case marked as oblique. Hence this position cannot be followed by *ră* in subject complement clauses. This justifies the ungrammaticality of (3.127) in which *Ali*, being the subject of the embedded clause, is followed by a marker of obliqueness.

(3.127) * *ali* *ră* be-nazar-mires-eh [ke sib ră xord-eh ast].
Ali SPCF is-seeming [that APPLE SPCF eaten is-3S]
'It seems that Ali has eaten the apple.'

In (3.127) Ali cannot receive oblique case from the verb. In fact according to our analysis in these cases the SPEC can only get the case of the subject complement which is not oblique. But the verb is always third person. This is further highlighted in the following example:

(3.128) tu be-nazar-mires-eh [ke sib ră xord-e -i].
You is-seeming-3S [that APPLE SPCF eaten is-2S]
'It seems that you have eaten the apple.'

(3.129) (in_i) be-nazar-mires-eh [ke tu sib ră xord-e -i]$_i$.
(this) is-seeming-3S [that you APPLE SPCF eaten is-2S]
'It seems that you have eaten the apple.'

Note that although the sentence in (3.128)is grammatical, but there is no agreement between *tu* and *be-nazar-mires-eh*. As a result the inflection cannot case mark *tu*. When *in* is present (e.g. in (3.129) there is no scrambling possible, whether the embedded clause is extraposed or not. The only possible answer is to consider all instances of this type of scrambling in *seem* as adjunct attachment. But the solution requires that we assume subjects and objects that are not governed by any preposition can also be moved by adjunction[26], since in Persian we have examples such as (3.130) where an object and a subject are moved from an embedded clause to a domain higher:

(3.130) ali *sib* *ră* be-nazar-mires-eh [ke __ xord-eh ast].
Ali *apple SPCF* is-seeming [that __ eaten is-3S]
'It seems that Ali has eaten the apple.'

It seems that in Persian, the modal-like verbs that have a subject complement behave differently when their complement clause is not dominated by an NP (i.e. *in*.)

Based on Yoon's arguments on movement of these arguments into an A-position and the fact that any number of arguments from the embedded clause can scramble and come before

[26]This wasn't possible for the non-subject case.

the modal-like verb, we conclude that these modal-like verbs, when their subject clausal arguments are not dominated by an NP (i.e. *in*), behave like modal verbs in Persian.

The only restriction on the movement is that the modal verb and the optional comp *ke* must precede the verb of the clause. Note that the modal-like verb and *ke* behave as a parenthetical constituent. This is also true for other modals of Persian:

(3.131) ali sib rǎ bǎ čangal bǎyad (ke) xord-eh bǎsh-ad.
Ali apple SPCF with fork must (that) eaten SUB-is-3S
'Ali must have eaten the apple with fork.'

Here *ke* functions as an optional stress marker[27]. Based on this we can represent sentences such as (3.132) where all the arguments come before the modal verb.

(3.132) ali sib rǎ bǎ čangal be-nazar-mires-eh (ke) xord-eh ast.
Ali apple SPCF with fork is-seeming that eaten is-3S
'It seems that Ali has eaten the apple.'

In the rest of this section we will consider fronting in other kinds of embedded clauses, such as non-finite and relative clauses.

The structure we outlined in Figure 3.1 is analogous to the structure of an NP which is modified by a relative clause. The difference is that in the latter the NP must be co-indexed with an empty category in the embedded clause (i.e. Cl-arg in that Figure). The former case is similar to the case of noun complement structure in Persian. In general, in the above structure, the tensed clause and the dominating NP act as barriers and therefore fronting cannot occur in relative clauses, tensed clausal arguments and noun complement structures:

(3.133) ali in rǎ say=kard [ke sib rǎ be bǎzǎr be-bar-ad].
Ali this SPCF try=did [that apple SPCF to Bazar SUB-take-3S].
'Ali tried to take the apple to the bazaar.'

(3.134) * Ali sib rǎ in rǎ say=kard [ke be bǎzǎr be-bar-ad].
Ali apple SPCF this SPCF try=did [that to Bazar SUB-take-3S].
'Ali tried to take the apple to the bazaar.'

[27] See [Nu-bahar, 1992] for different functions of *ke*.

But for the non-finite clauses the situation is different. They are neither tensed clauses and nor dominated by a NP, so the fronting from them is possible. In fact these clauses act as NPs; the same phenomenon of NP left-dislocation that we described earlier exists for them. An example of fronting for (3.25) is shown in (3.135).

(3.135) sib ra, [ali, man ră majbur be [xord-an-esh kard]].
Apple SPCF, [Ali I SPCF persuade to [eat-INF-it did]].
'The apple, Ali persuaded me to eat.'

As in the example of NP topicalisation we discussed earlier, the topicalised NP usually appears at the beginning of the sentence; hence, it must precede the object of the sentence.

(3.136) * ali man ră sib ră majbur be [xord-an-esh kard].
Ali I SPCF apple SPCF persuade to [eat-INF-it did].
'The apple, Ali persuaded me to eat.'

3.6.6 The Reverse Case of Fronting in Relative Clauses

In relative clauses, as we discussed earlier, there are no cases of fronting or scrambling. Here we will instead concentrate on the issue of case marking in constructions which involve relative clauses.

Comrie [1981] gives interesting examples of case marking of head noun phrases that are modified by relative clauses. The examples are:

(3.137) zan-i [ke did-id] injă-st.
woman-RES [that saw-2P] here-is
'The woman that you saw is here.'

(3.138) ăn zan-i ră [ke diruz amad] did-am.
that woman-RES SPCF [that yesterday came-3S] saw-1S
'I saw that woman who came yesterday.'

In (3.138), the head noun phrase of the relative clause can become *attracted* to the relative clause and lose its specific object marker *ră*. This is shown in (3.139).

(3.139) ăn zan-i [ke diruz amad] did-am.
that woman-RES [that yesterday came-3S] saw-1S
'I saw that woman who came yesterday.'

This is further highlighted in (3.140).

(3.140) [ăn zan-i ke diruz amad] (ra) did-am.
that woman-RES [that yesterday came-3S] (SPCF) saw-1S
'I saw that woman who came yesterday.'

Note that here the head noun phrase and the relative clause can be case marked with *ră*, which is here a specific accusative marker[28].

A phenomenon similar to this is present in Latin and Greek; it is called *Attraction*.

(3.141) illustrates another example of attraction:

(3.141) a. in sib-i ră [ke __ injă bud] xord-am.
this apple-REL SPCF [that __ here was-3S] ate-3S
'I ate the apple which was here.'

b. * in sib-i xord-am [ke __ injă bud].

c. in sib-i ră xord-am [ke __ injă bud].

d. * in sib-i [ke __ injă bud] xord-am.

e. in sib-i [ke __ injă bud] ră xord-am.

In (a) *in sib-i*, is specific and being the object, is marked accusative by the matrix verb, so *ră* must appear after it. This is the reason why the (b) sentence without *ră* is ungrammatical

[28]What is interesting is that when the relative clause is extraposed, then the presence of *ră* is obligatory, while when the specific head noun phrase is attracted, the presence of *ră* becomes optional:

ăn zan-i ră did-am ke diruz amad.
∗ ăn zan-i did-am ke diruz amad.

and (c) is grammatical. In (d) the matrix verb's accusative case is not assigned properly, so the sentence is ungrammatical. This is in contrast to (e) where the whole relative clause is marked by *ră* as accusative. Note that in (e) sib-i (the head noun of the relative clause) receives nominative case from the verb of the relative clause.

These examples show that in Persian there is a difference between an NP as a head of a relative clause and the whole relative clause construction, and they can separately receive case marking.

Note that attraction is not restricted to examples where the head noun is a direct object in the relative clause, but *ră* only appears after attracted noun phrases which are not subjects in the relative clause.[29]

(3.142) a. mard-i ke [sib ră xord-e bud] injă-st.
man-RES that [apple SPCF eat-en was] here-is
'The man who has eaten the apple is here.'

b. * [mard-i ră ke sib ră xord-e bud] injă-st.

(3.143) a. mard-i ke [sib ră be-esh dad-am] injă-st.
man-RES that [apple SPCF to-him gave-1S] here-is
'The man to whom I gave the apple is here.'

b. [mard-i ră ke sib ră be-esh dad-am] injă-st.

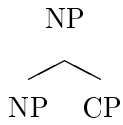
(3.144) a. mard-i ke [bač-esh ră did-am] injă-st.
man-RES that [child-him SPCF saw-1S] here-is
'The man, I saw whose child is here.'

b. [mard-i ră ke bač-esh ră did-am] injă-st.

But what is the structure of relative clauses to accommodate these examples of case marking, and how does the case tendency principle work for attraction in Persian?

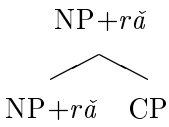
[29] In general when the head noun is governed by a preposition attraction does not apply. In other words the preposition case marking is very strong.

Samiian [1983] argues that the relative clause in Persian is a sister to the head noun phrase:



But her proposal falls short in giving account of attraction in Persian.

Karimi [1990] suggests another configuration as follows:



According to this configuration, *rǎ* may appear following the head noun of the relative clause or the complete relative noun phrase [Karimi, 1990]. Karimi further suggests that the principle of Case Tendency[30] is responsible for the different examples of attraction in Persian. But she gives no more details about the underlying principles of case tendency and attraction in Persian and does not formalize them further.

In order to capture attraction in relative clauses we propose the structure in Figure 3.9. The head noun, when it is located in its NP position, can be case marked from outside of the relative clause, especially when the relative clause is extraposed. When the head noun is located in an A′-position (i.e. SPEC) then the whole relative clause can be case marked and the head noun gets its case marking from its empty position inside the relative clause.

Note that in both cases the head noun projects an NP barrier and prevents any example of scrambling from inside of the relative clause into the matrix clause. That is, the two landing sites for the head noun of a relative clause (NP and SPEC) are unified. In other words, this will force always a projection of NP that acts as a barrier for extraction out of the relative clause.

When a noun phrase is attracted, it will be case marked locally from the relative clause. In this case, if the head noun is co-indexed with a non-subject A′-SPEC position and is specific then it will be case marked by specific oblique marker *rǎ*. Note that attraction is only possible in restrictive relative clauses. Afarli [1994] discusses a *promotion* analysis for

[30]According to the case tendency principle, the case of a non-argument NP tends to be determined by its position in the CP containing it, or the closest CP [Karimi, 1990].

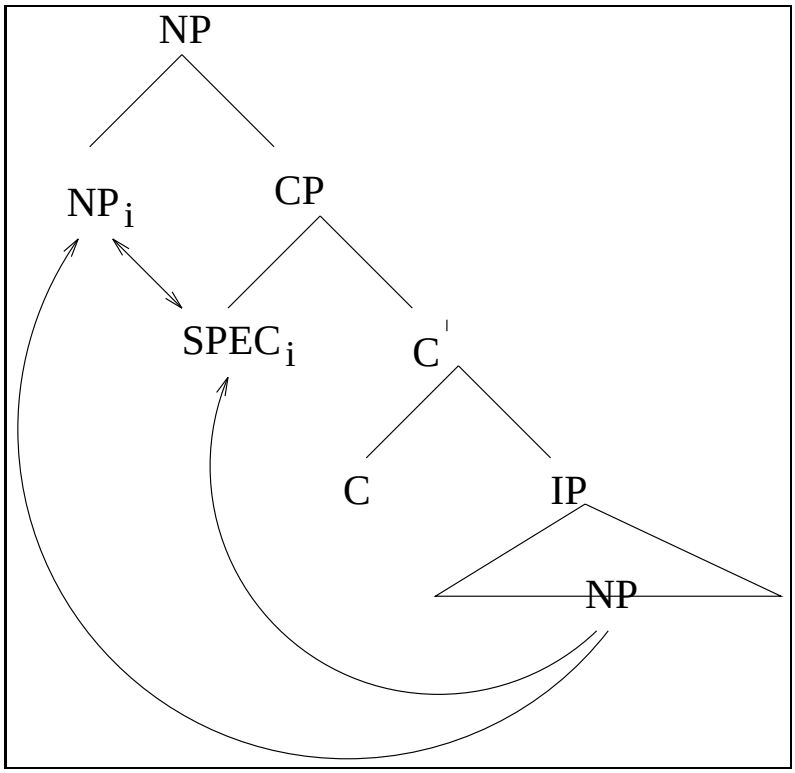


Figure 3.9: A Structure for Relative Clauses in Persian

restrictive relative clauses in Norwegian. Her approach is analogous to ours. She considers two separate structures for Norwegian restrictive relative clauses, with promotion and with no promotion. The former corresponds to the case with attraction in Persian and the latter corresponds to the traditional treatment of head nouns as separate constituents from relative clauses. Due to restricted time scale for our work, we do not go into the details of this.

The structure of relative clauses may be considered as a parallel to the structure Persian complement clauses that we studied. This is illustrated in Figure 3.10. But this needs further investigation.

In summary, we further formalized the Case Tendency Principle in Persian based on our proposed structure for Persian embedded clauses.

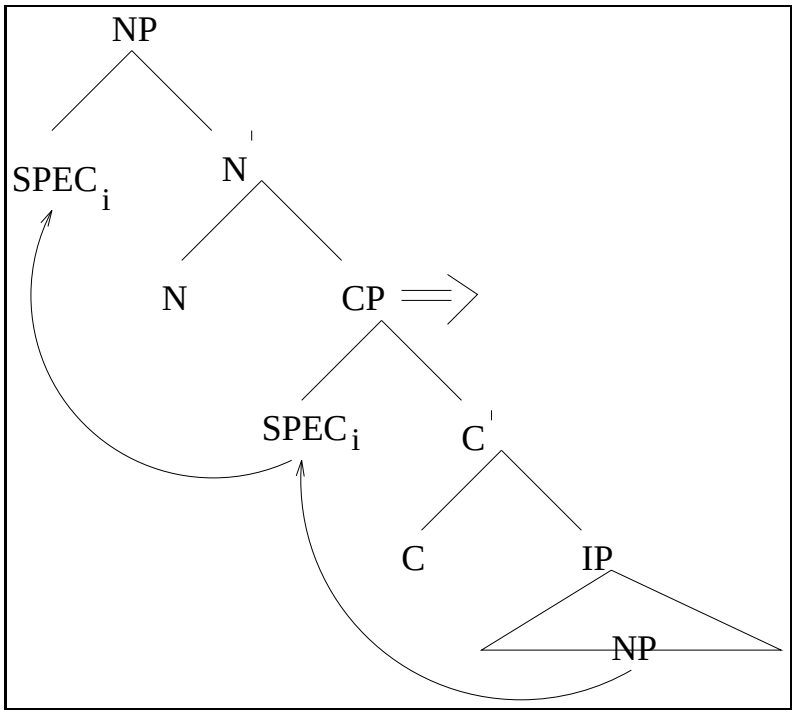


Figure 3.10: Relative Clauses as Complement clauses

3.7 Conclusion and Summary

In the previous sections we discussed embedded clauses of Persian and our analysis further supports the proposal[31] that:

1. Sentential arguments originate in pre-verbal position in Persian.
2. They are dominated by an NP.

These arguments are often moved to the post-verbal position by an obligatory extraposition. In our approach we captured the fronting of noun phrases. In this framework we assumed that the fronted category is in fact part of the extraposed clause and during the clause movement this extraposed category is left in its actual place. In other words our approach contrasts with the traditional approach to fronting which treats fronting as an exceptional leftward movement, while we do not treat it as a case of leftward movement. We

[31] [Karimi, 1989] contains a summary of previous proposals for representing sentential arguments in Persian, work such as Moyne and Carden [1974], Soheili-Isfahani [1976] and Dabir-Mogaddam [1982]

further proposed that this left-over category, if it is not already case marked by a preposition (i.e. scrambling), will receive oblique case from the verb of the matrix clause. In summary we argued for these constraints on embedded clauses:

- If the clausal argument is non-finite (not tensed) then the clausal argument is not dominated by a noun phrase (NP) (i.e. the place is empty or e).
- If the clausal argument is finite, then extraposition is possible:
  - If it is extraposed, then the dominating NP with the clausal argument is normally empty. In this case fronting and scrambling into the matrix clause is possible.
  - If it is extraposed, but the dominating NP is present then fronting and scrambling into the matrix clause is not possible.
  - If it is not extraposed, then it must be dominated by an NP. As in the previous case no fronting and scrambling into the matrix clause is possible.

In the case of movement from the finite non-subject embedded clauses we argued that:

1. In the case of fronting, the fronted noun phrase is case marked inside the new clause, but it agrees with its trace in number (and person). A case of weak unbounded dependency.
2. In the case of scrambling , the scrambled noun phrase is not case marked inside the new clause and it agrees with its trace both in number and case. A case of strong unbounded dependency.
3. Only one of the NPs of the extraposed clause can be fronted and moves to the SPEC position of the complement clause in preverbal position. These are marked by *rā* for non-subject complement clauses.
4. Other NPs of the extraposed clause which scramble into the matrix clause must be properly case marked.

We considered movement from embedded subject complements as examples of local scrambling where the modal-like verb behaves as a modal verb. The modal verbs in Persian, do not require agreement.

We also argued that in Persian tensed clauses and NPs act as barriers and as a result it is not possible to raise categories from inside clauses which are dominated by both of these. And finally we considered the case marking of relative clauses in Persian and we argued that the attraction phenomenon in Persian is a result of a promotion like phenomenon.

The proposed principle of case tendency in Persian [Karimi, 1990] was further suggested to be a result of the interaction of deeper principles of the universal grammar, but with different parameter settings for Persian.

Having studied the different constituents and structures in Persian syntax, in the second part of the thesis we will concentrate on computational aspects of a parsing system for Persian. In the next chapter we will have a brief review of the main parsing systems that have been developed for processing Persian. In our work we are primarily concerned with the treatment of word order and scrambling in Persian. This is one of the main areas which have been neglected by the previous approaches.

In our study we will not attempt to implement a GB based parser as in Fong [1997]. The GB theory and the principles-and-parameters framework are under revision and the recent Minimalist Program (Chomsky [1995]) had not been stabilized at the time of our research. But notions of competition and resource sensitivity analogous to those discussed in Chapter 6 are also discussed in the Minimalist Program (MP) literature. In Chapter 6 our focus will be on scrambling constraints and performance based word order principles. The study of these constraints and their interaction with the performance system have been largely neglected in principles-and-parameters framework and the Minimalist Program.

The next part of the thesis will give a complementary perspective to the word order constraints in Persian that we have discussed so far. For the grammar of Persian, we will look at the hypothesis that the parsing architecture (the performance system) imposes performance constraints on the competence grammar. This hypothesis will be spelled out by introducing two resource limitation principles for parsing scrambling data in Persian. These constraints will be discussed in Chapter 6.

Chapter 4

Survey: Processing Persian

A general assumption of a theory for grammatical analysis has been the existence of two distinct components: a grammar and a processing device. The grammar defines a set of strings which comprises all and only the sentences of the language under question. The processing device applies the rules of grammar to produce or analyse the grammatical strings. Depending on whether we concentrate on producing or analysing grammatical strings we will have generators or parsers.

In this chapter we will concentrate on the second component and will review some of the previous approaches to systems which have been developed for processing Persian.

This chapter is a general introduction to parsing Persian and we will build on the previous parsers to reach a more efficient framework in Chapter 6 for processing scrambling examples in Persian.

4.1 Rule Based Parsing

One of the first and most comprehensive parsers (analysers) for parsing sentences of Persian is the PERSIS system [Sanamrad and Matsumoto, 1985]. PERSIS is based on a grammar model implemented using more than 850 syntactic production rules. In constructing PERSIS, two descriptive grammars of Persian were used: [Lambton, 1953] and [Khanlari, 1965]. Some typical syntactic production rules are illustrated in Table 4.1.

The order of the arrow in a production rule is the reverse of the arrow in a CF rule. So the following rule from Table 4.1:

PS rule	
N hă → N	(plural)
N e/ye N → N	(Genitive Case)
N e/ye ADJ → N	(Adjectives)
N i ke SNT → N	(Relativised Sentence)
ADV ADV → ADV	(PPs in a Sentence)
VRB → PRD	(Predicate/verb)
ADV PRD → PRD	
DOBJ PRD → PRD	(Direct object of the verb)
N PRD → SNT	(Subject of the verb)
PRD → SNT	(pro-drop Subject)

Table 4.1: Production rules in PERSIS

N hă → N (plural)

will be written as below in a CFG notation.

N → N hă

In PERSIS the production rules are augmented by a set of attribute and feature values. In PERSIS, each word or phrase has an additional list of attributes. These attributes convey additional syntactic and semantic information about that word or phrase. These attributes and their values (i.e. attribute prototypes in PERSIS) are used with the grammar rules and the parser checks the consistency of attributes (e.g. of a noun group and its modifier). Note that PERSIS was the first analyser of Persian, but unfortunately, later researchers on parsing Persian have been unaware of PERSIS.

There are 17 different attribute prototypes in the system, which represent the meaning encoded in different structures. These correspond to the following:

(i) Noun (Phrases) (ii) Adjective (Phrases) (iii) Adverbs (iv) Verbs (v) Predicates (vi) Pronouns (vii) Unit (viii) Interjections (ix) Numbers (x) Time (xi) Day (xii) Week (xiii) Month (xiv) Year (xv) Time Periods (xvi) Preposition (-al Phrases) (xvii) Text

1) Noun, DOBJect, VOCative	2) VeRB, PRDicate, SNTence, ...
Normal/Interrogative	Indicative/Interrogative/Imperative
Person	Positive/Negative
Quantity	Attributes of INF (e.g. in/transitivity)
Abstract/Physical/Proper	Attributes of subject N
Counter Class	Attributes of ADJective (for some verbs)
Human/Animal/Place/Time/Cond., State/Inanimate	Attributes of object N
Attributes of Apposition N	Attributes of ADVerb
Attributes of N in Genitive Case	Simple/Comp.: Reason/Cond./Time,Else
Attributes of ADJ	Attributes of SNT - when compound
Attributes of Relativised SNT	Tense

Table 4.2: 2 Attribute Prototypes of PERSIS

Two examples of attribute prototypes are shown in Table 4.2.

In addition to these, the system uses about 100 words, particles, case-markers, suffixes, prefixes which are used in syntactic patterns and act as "functional operators" in determining grammatical structures [Sanamrad and Matsumoto, 1985]. These are stored in the dictionary of PERSIS.

So far PERSIS is the most sophisticated and large coverage analyser for Persian texts that has been implemented. But since it has been developed for the analysis of written Persian, it assumes that the verb of the sentence appears at the end. Nevertheless it considers the flexible order of adverbials, as long as they appear before the verb. It seems that the grammar assumes that objects come before all adverbials. PERSIS also doesn't consider examples of long distance scrambling which is used in spoken Persian. It also doesn't consider linguistic issues like control.

PERSIS parsing engine is a depth first mechanism, which produces the first parse for an input sentence. But it can be extended to produce all parses.

PERSIS produces an embedded dependency network corresponding to the input sentence. The dependency network is similar to the representations used in Conceptual Dependency (CD) theory of Schank [1975] and Gershman [1982].

In passing we should mention another analyser for Persian [Rais-Ghasem, 1991] which produces CD representation for an input sentence. Rais-Ghasem has built a semantic parser for Persian which looks like other interlingua based systems that were built based on the work

of Schank and his colleagues [Schank, 1975]. He considers only simple clauses with no instance of embedded clauses, but with almost no restriction on the order of clausal arguments[1].

4.2 An Extension to ATN for parsing Persian

Rezaei has attempted to build an ATN system for parsing (and generating) simple sentences of Persian with examples of local scrambling [Rezaei, 1992]. The main linguistic sources used for his work were the systematic grammar of Persian [Bateni, 1970] and [al Dini, 1987]. The implemented parser is based on Kashket's principle based parser [Kashket, 1986]. Following Kashket, Rezaei built a two-stage parser. The two stages are:

- Chunking: corresponding to PS level in Kashket's parser.
- Subcategorisation: corresponding to SS level in Kashket's parser.

In the first stage, the maximal projections such as NP, PP (here treated as adverbial), and the verb corresponding to the input string are identified by an ATN network.

In contrast to Kashket's PS level, Rezaei assumes the existence of hierarchical information at this level and as a result he captures noun phrase coordination and prepositional phrase coordination in the chunking stage. This is shown in Figure 4.1.

In this ATN network, JUMP arcs are represented by #. The non-terminals are represented by capital letters, and terminal symbols such as *va* (i.e. and), and *ră* the specific object marker in Persian are represented by small letters.

In the second stage, according to the information from the first stage and the subcategorisation information of the verb, the subject and object and predicate nominal of the sentence are identified. In the system, verbs can be either transitive, intransitive or stative (or linking verb). At this stage no fixed constituent order is assumed and by a general loop the necessary arguments of the verb are identified. In this stage the grammatical functions of the constituents are determined and they are attached to the verb.

At this point, the disambiguation between subject and object is the major problem, which is resolved by a procedure based on the work of Karimi on specificity and word-order [Karimi,

[1] [Fahimi and Shamsfard, 1995] is a project based on [Rais-Ghasem, 1991] which extends the coverage of Rais-Ghasem system.

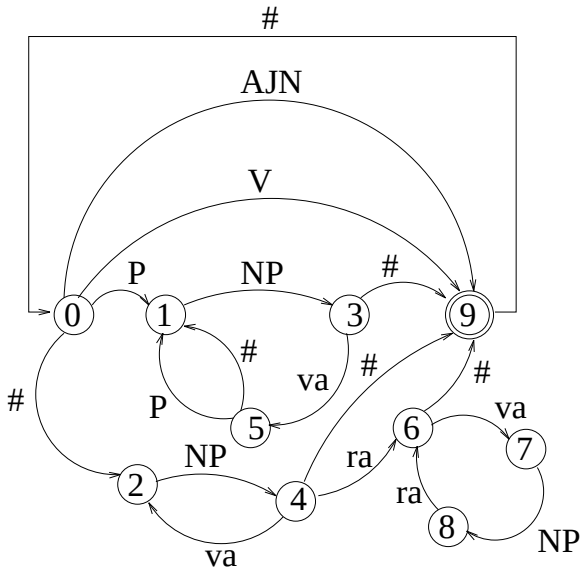


Figure 4.1: S Network of the ATN.

Semantic	specific, non-specific, pronoun
Structural	Cat(egory)
Syntactic	Subject, Object, Adverbial, predicate-nominal(mosnad)

Table 4.3: Features in ATN Analysis of Persian

1989]. In the system, specific and non-specific arguments are distinguished from each other. In general the parser uses the information that is shown in Table 4.3. The major drawback of this procedural representation is that it is hard to understand and modify. In addition since the parser has been mainly designed for capturing scrambling inside a clause, the parser doesn't deal with embedded clauses, and it is difficult to extend it to do this.

4.3 ID/LP parser for Persian

[Rezaei, 1993] is another work on building a parser for parsing examples of Persian clauses with scrambling. The parser is a declarative extension to the ATN parser of the previous section and it is also a modified version of the ID/LP framework. For capturing local scrambling, it employs a concept of domain which is an extension to Reape's proposal for word order

domain[2] [Reape, 1996].

Like the ATN parser, the ID/LP parser works in two stages. In the first stage, by employing phrase structure rules, the input words of the sentence are grouped into chunks such as NP, PP and V. In the second stage of the parser, the clauses are formed by employing Immediate Dominance (ID) rules.

Such rules are:

(4.1)
(id-1) $\text{VP}_{Subcat} \rightarrow \text{VP}_{[X \cup Subcat]}$, X [-subj]
(id-2) $\text{Clause}_{Subcat} \rightarrow \text{VP}_{[X \cup Subcat]}$, X [+subj]
(id-3) $\text{Clause}_{[chain/subj=EC]} \rightarrow$ VP
(id-4) $\text{Clause}_{[chain/nonsubj=EC]} \rightarrow$ Clause
(id-5) $\text{Clause-minor}_{Chain} \rightarrow$ ke , Clause_{Chain}
(id-6) $\text{NP} \rightarrow \text{NP}_i$, $\text{clause-minor}_{ec(-,i,-)}$
(id-7) $\text{Clause}_{Subcat} \rightarrow \text{clause}_{Subcat}$, clause-minor

In the notation used for representing the rules, [X∪Rest] represents a set (or bag) which 'X' is one of its members, and 'Rest' is the rest of its members. X [-subj] represents an argument which is not the subject. In these rules, a chain is used for passing information about missing noun phrases and gaps.

By a set of filters which apply locally, the LP constraints are imposed on the ID rules. For example the filter for the rules that join a subject argument and a VP are illustrated next:

- If either of the subject or object is specific then there is no ambiguity for determination of subject and object. Specific object marker *ră* disambiguates the subject and object. (CASE 1)
- If neither of the subject nor the object is specific then only the subject of the clause is allowed to appear after the verb. (CASE 2)
- Otherwise the default unmarked order of the clause (i.e. SOV) holds. That is, subject precedes object. (CASE 3)

[2]We will discuss Reape's word order domains in Section 5.1.1.

The filters use control domains - similar to Reape's word order domains - for checking adjacency of chunks/constituents. The major drawback of this parsing system is that the parser is very inefficient compared to the previous parsers for Persian, and although it can handle embedded clauses in a concise way, it is not capable of parsing all examples of long distance scrambling over these clauses. But the system parses examples of long distance scrambling and imposes some constraints on the set of possible examples with local scrambling. Unlike the ATN parser, the grammar representation is expressive[3].

4.4 Summary

In this chapter we have reviewed some of the recent processing systems for parsing Persian. Table 4.4 summarises the major features of these systems. In the table we have also included two recent pieces of work for processing Persian. [Riazati, 1997] is a two level morphological system for Persian with a limited syntactic parser. SHIRAZ is an on-going Persian-English machine translation project in the US which will be completed by September 1999 [Cowie et al., 1997]. There is another major work on processing Persian in Iran [Fahimi and Shamsfard, 1995] which we haven't included in this chapter because of unavailability of resources.[4]

PERSIS uses a bottom-up parser while [Rezaei, 1992] uses a two-stage parser. At the first stage, the lexicon is searched as a bottom up parser, and at the second stage the ATN parses in a top down fashion. [Rezaei, 1993] uses a similar two-stage parser, but the parser works bottom up. The first stage uses CFG rules while the second stage uses ID/LP rules.

From the linguistic aspect, PERSIS covers more than the other parsers for Persian. PERSIS has concentrated on simple clauses with very detailed examples, but it fails to deal with complex examples of scrambling in Persian clauses and does not discuss linguistic notions such as control. [Rezaei, 1992] only tries to parse simple clauses but it concentrates on extending the ATN for dealing with local scrambling. The ATN network also represents coordination in NPs and PPs. [Rezaei, 1993] captures more complex examples of relative clauses, extraposed clauses and complex clauses with instances of long distance scrambling. This is the first implemented parser that deals with long distance scrambling in Persian, but the extension to

[3]There are other systems that we haven't considered. One of them is Kuznick [1988] which claims to be able to parse Persian and English sentences, but the parser cannot parse SOV sentences which is the major constituent structure of Persian.

[4][Fahimi and Shamsfard, 1995] is a general introduction and does not give the details of the parsing system.

	PERSIS	RaiesGhasem	Rezaei-1	Rezaei-2	Riazati	SHIRAZ
Approach	Production Rule	Conceptual Dependency	ATN	ID/LP	KIMMO 2 Level	CORRELI
Parser	Bottom-Up	Procedural	Bottom-Up Top Down	Bottom-Up	Bottom-Up	Bottom-Up?
Tokenisation	NO	NO	NO	NO	NO	YES
Morphology	NO	NO	NO	NO	YES	YES
Explicit Ezafe	YES	YES	YES	YES	YES	NO
Coordination	YES	NO	YES	NO	NO	?
Local Scrambling	V-final	unrestricted	YES	YES	Limited	V-final
Complement Clauses	YES	NO	NO	YES	NO	NO
Relative Clauses	YES	NO	NO	YES	NO	YES
Long Dis. Scrambling	NO	NO	NO	Fronting	NO	NO
Control	NO	NO	NO	NO	NO	NO
Multiple Parses	NO	NO	NO	YES	YES	YES

Table 4.4: Parsing Systems for Persian

ID/LP and the existence of optional pro-drop in Persian makes the system very inefficient. Note that the data for SHIRAZ MT project are speculative[5]. Among the systems, only SHIRAZ assumes no explicit *ezafe* in the Persian texts, we will come back to this issue in the final chapter.

In the remaining chapters we will elaborate on a parsing system which we have implemented for the efficient analysis of examples in Persian with local and long distance scrambling. In the next chapter we will first have a closer look at possible alternatives which have been proposed in different formalisms, to deal with scrambling.

[5]These are based on the period I was working on the project.

Chapter 5

Free Word Order and Discontinuous Constituency

For the last decade free word order languages have posed one of the most challenging problems facing natural language parsers. In the literature there are a number of reports on parsing languages such as Finnish, Warlpiri, German, Dutch and Persian. All these languages have one thing in common: the possibility of word order variation in their sentences is less restricted than in English. But what are the characteristics of free word order languages?

Latin is one of the languages in which many permutations of the words of a sentence yield another grammatical sentence, but with almost identical meaning. We say another grammatical sentence because there are always some intonational and pragmatic differences between the two sentences and these two sentences are not generally interchangeable. In other words their meaning is slightly different.

Roughly speaking in this respect Latin can be viewed as an absolute notion for a free word order language and other languages are somehow between this extreme case and English, which can be viewed as a highly fixed word order language.We can say that in free word order languages the word order primarily determines pragmatic information, while in less free word order languages such as English it also conveys structural and syntactic information.

Throughout this thesis, the term *word order* is used in its traditional linguistic meaning, referring to the linear order of constituents. Thus, no distinction is made between free word order and free constituent order, and in this respect we are following Uszkoreit [1987]. Languages like Finnish, German and Persian are considered to be free constituent order lan-

Morphological	Case, Number, Aspect, Quantity
Phonological	Emphasis
Semantic	Positive, Aspect, Quantity
Structural	Cat(egory), Pattern, Branching
Syntactic	Subject, Object, Adverb
Pragmatic	Topic, Contrast, New

Table 5.1: Features in Karttunen's Analysis

guages.

For representing free word order languages, traditional approaches are not very appropriate and they need to be extended or modified in order to be able to deal with phenomena such as local scrambling (movement of constituents inside a clause boundary) and long distance scrambling (movement of constituents across clause boundaries). But what makes a grammar adequate for describing a free word order language? And what makes a parsing algorithm adequate for processing a free word order language?

5.1 Approaches to Free Word Order

Karttunen and Kay [1985] is one of the earliest unification based systems for analysis of Finnish word order. Karttunen and Kay employ FUG (Functional Unification Grammar) in which each grammatical phrase of a language has only one functional representation or description (FD). In other words there is no phrase structure rule in the grammar, and the dominance hierarchy of mother and daughter nodes is also represented inside FD's (i.e. similar to a lexicalist approach).

In Karttunen and Kay's approach each FD can have a set of possible features, ranging from phonological to semantic properties. Table 5.1 illustrates some of the features which they employ [Karttunen and Kay, 1985].

As shown, Karttunen and Kay have considered a broad and general set of features for analysing free word phenomena in Finnish including pragmatic and semantic properties. In fact for parsing free word order languages it is necessary to focus on semantic and pragmatic properties, because the word order in languages with a flexible word order does not provide the necessary information for identifying grammatical relations and other mechanisms need

to be employed. In this respect their work can be considered as a good starting point for working on free word order languages.

To capture free word order phenomena we must focus on pragmatic and other linguistic features (e.g. specificity) and non-linguistic features. Features should also be considered for representing the order of constituents in the input string, i.e. features for precedence information.

[Karttunen and Kay, 1985] does not give a specific parsing model or an efficient technique for parsing and it only gives an outline of Finnish syntax in the framework of functional unification grammar which is discussed in more detail in another paper by Kay in the same book [Kay, 1985]. The main result of the work is the demonstration that the complexity of surface ordering in Finnish arises from the interplay of a small number of simple word order principles that involve syntactic functions and discourse functions.

In the following sections we will consider some formalisms and systems which have been designed for representing the grammar of free word order languages. We will elaborate on the specific problems that the grammar of free word order languages will create for traditional approaches.

5.1.1 ID/LP

In many approaches to free word order, the grammar is divided into two components: the immediate dominance (ID) and linear precedence (LP) rules. These rules can be considered as extensions to general phrase structure rules. In this section, I discuss the use of the ID/LP notation in GPSG and HPSG.

GPSG

Most of the work on computational linguistics in the past has been relied on traditional phrase structure rules. In this kind of system each rule specifies two distinct relations:

- Linear Precedence relations among daughter categories (i.e. right hand side categories in a rule).
- Immediate Dominance relations between the mother category (i.e. the left hand side category in a rule) and each of its daughters (i.e. right hand side categories).

PS rules	ID rules	LP rules
VP $\rightarrow$ V NP	VP $\rightarrow$ V, NP	VP $\prec$ PP
VP $\rightarrow$ NP V	VP $\rightarrow$ NP, VP	NP $\prec$ VP
VP $\rightarrow$ NP VP	VP $\rightarrow$ NP, VP, PP	NP $\prec$ PP
VP $\rightarrow$ NP VP PP	NP $\rightarrow$ NP, PP	
NP $\rightarrow$ NP PP		

Table 5.2: A Comparison of PS Rules and ID/LP Rules

In contrast to this view, in Generalised Phrase Structure Grammar (GPSG) [Gazdar et al., 1985] these two relations are specified by two different kinds of rules:

- Immediate Dominance(ID) rules
- Linear Precedence (LP) rules

Immediate Dominance rules in GPSG *only* specify immediate dominance relations between mother and daughter categories of a rule and do not specify the order of the right hand side elements (i.e. daughters). In other words the right hand side elements are unordered.

Ordering relations in GPSG are specified by Linear Precedence relations. Each LP rule only specifies an ordering relation between two categories in the right hand side of the same rule and it is of the form $\alpha \prec \beta$. This rule means that if α and β ever appear together in the right hand side of an ID rule, then α should precede β. Thus LP rules are notationally detached from ID rules and apply independently. The LP rules filter the strings that are permitted by the ID rules. As a result it is not possible to define ordering constraints for two categories which are not in the right hand side of a rule (i.e. word order is derived from the surface constituent structure).

A comparison of ID/LP rules and phrase structure (PS) rules is illustrated in Table 5.2, where '$\prec$' shows precedence relation for ID rules.

Following research in Generalised Phrase Structure Grammar (GPSG), Uszkoreit addresses free word order phenomena in German [Uszkoreit, 1987]. The grammatical framework chosen by him is a modification of the Immediate Dominance/Linear Precedence (ID/LP) version of GPSG. Uszkoreit redefines LP rules in order to allow potentially conflicting ordering principles to be present in the LP rule set.

Based on this framework Uszkoreit discusses word order and constituent structure in German. In his work for capturing ordering principles in German, he employs pragmatic

(1)	AGENT	$\prec$	THEME
(2)	AGENT	$\prec$	GOAL
(3)	GOAL	$\prec$	THEME
(4)	-FOCUS	$\prec$	+FOCUS
(5)	+PPRN	$\prec$	-PPRN

Table 5.3: LP Rules in Uszkoreit's Analysis

features such as **focus** and **theme**. His proposed LP constraints are illustrated in Table 5.3. (here PPRN stands for personal pronoun) [Uszkoreit, 1985].

In standard GPSG notation it is not possible to have conflicting ordering principles because the LP rules apply conjunctively (i.e. a local tree admitted by an ID rule has to satisfy all LP rules at once). In contrast Uszkoreit introduces disjunctive LP rules which can be violated as long as at least one of the rules holds true. For example in Table 5.3 a rule can violate precedence constraints 2, 3, 4, 5 if it satisfies 1.

Uszkoreit [1987] gives examples of long distance scrambling in German, where constituents from embedded clauses are moved up to the matrix clause.

(5.1) Dann hatte er {den Bestohlenen}$_1$ {die gleichen B *ucher*}$_2$ versucht e$_1$ e$_2$ zu Schleuder
then had he the theft-victims the same books tried $-$ $-$ to dumping
preisen zur *uckzuverkaufen*.
prices back-to-sell
'then he tried to sell the same books to the theft victims again at dumping prices.'

But his system needs further extension and research to deal with these examples of free word order where we have instances of cross-serial dependency.

In passing we should refer to JPSG [Gunji, 1987], another extension of GPSG, for Japanese. In Gunji's approach for capturing local scrambling, the subcategorisation list of a verb is represented as an unordered set. However the grammar cannot capture long distance scrambling.

In general, standard GPSG cannot handle multiple number of long distance scrambling since the SLASH mechanism can only handle one instance of long extraction.

By adding *liberation* rules, Zwicky [1986] extends the ID/LP formalism. Liberation rules are used to flatten the constituent structure. For example by collapsing two ID rules S $\rightarrow$ NP, VP and VP $\rightarrow$ NP, V into one ID rule S $\rightarrow$ NP, NP, V he captures local scrambling. This is achieved by eliminating or liberating two constituents of the VP node. Similarly the

examples of long distance scrambling can be handled by liberating the embedded S node. It is not clear whether all the constraints and restrictions on movement can be represented by this extension to ID/LP.

HPSG

Reape [1996] tries to capture possible word order variations in Germanic languages. Reape introduces the notion of word order *domain* for phrasal (or non-lexical) categories. In general the word order domain of a phrase consists of the word order domains of its children. In its elementary form the word order domain of a phrasal category contains its immediate lexical children.

Working with an HPSG framework, Reape employs a concept similar to GPSG LP rules to specify order inside a domain, and uses the same linear precedence binary relation (i.e. $\prec$) of GPSG. However his LP constraints are defined as well-formedness conditions on word order domains, rather than well-formedness conditions on local trees (i.e. right hand side categories of a rule).

Reape assumes that when two word order domains are merged together, the original internal order of each domain is preserved in the new word order domain. However it is possible for the elements of the two domains to be interleaved in the new word order domain. For example, let the word order domain of a category be equal to <NP[DAT] V_1> and the word order domain of another category be <NP[ACC] V_2> and assume the LP constraints:

(5.2) NP[DAT] $\prec$ NP[ACC]
NP $\prec$ V

If we want to merge the word order domains of these two categories, the result can only be one of the following word order domains:

(5.3) < NP[DAT] NP[ACC] V_1 V_2 >
< NP[DAT] NP[ACC] V_2 V_1 >

Notice that the first LP constraint has no effect on each of the unjoined domains, but it requires that the NP[DAT] precedes the NP[ACC] in the result. By employing the concept

of word order domain and using a shuffle operator for merging domains [Reape, 1996], Reape examines under the HPSG grammar formalism the word order variation in Dutch and German. The shuffle operator is computationally expensive to implement.

Reape also introduces a feature [unioned: +/-] to show whether two word order domains are allowed to be collapsed into one. [unioned: -] prevents the merging of two word order domains into each other and imposes island behavior in scrambling.

5.1.2 CG for Free Word Order Languages

Another framework which has been extended for capturing the grammar of free word languages is Categorial Grammar [Ajdukiewicz, 1935], [Bar-Hillel, 1953]. CG and its various extensions try to capture the function-argument relations in language and preserve a parallel and compositional syntax and semantics. In contrast to the constituent oriented approach in rule based systems, in CG the grammatical entities are of two types: Functions (functors) and basic elements (categories). Functions have one or more arguments, and the application rules allow functions to combine with their arguments. In this section we consider CUG and CCG.

CUG of Kartunnen

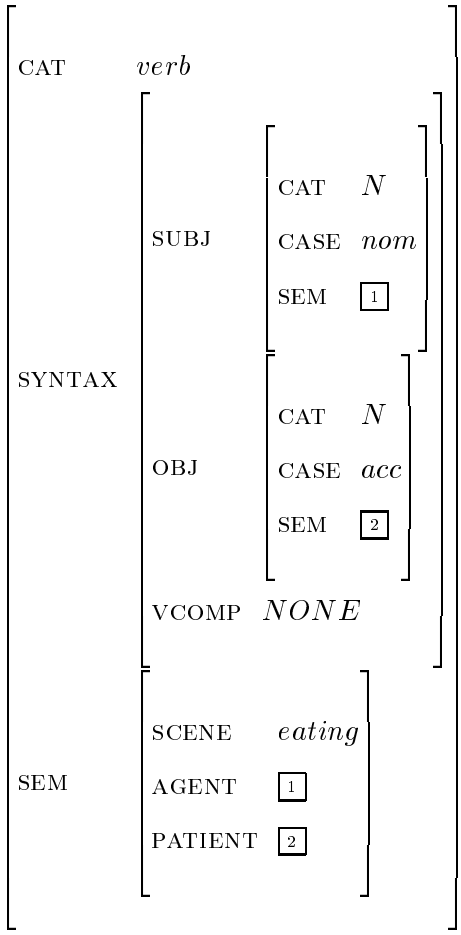
Using Categorial Unification Grammar, Karttunen [1989] analyses Finnish . In CUG [Uszkoreit, 1986] free word order is handled by treating noun phrases as functors that apply to the verbal basic elements. (5.4) shows the set of features (e.g. Nominative, Noun) for Ali.

(5.4) $Ali =$

$$\begin{bmatrix} \text{ARGUMENT } \boxed{1} & \begin{bmatrix} \text{CAT} & verb \\ \text{SYNTAX} & \begin{bmatrix} \text{SUBJ} & \begin{bmatrix} \text{CAT} & N \\ \text{CASE} & nom \\ \text{SEM} & Ali' \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{LEFT} & [] \\ \text{RIGHT} & [] \\ \text{RESULT} & \boxed{1} \end{bmatrix}$$

In Karttunen's analysis, the matrix verb is not a function, but a basic element with a set of features, e.g. SYNTAX. A verb's arguments combine with it in any order (left or right) and the linear order of the arguments is not specified in the SYNTAX feature. In this way local scrambling can be captured. An example of a verb is shown in (5.5). *Xord* (eating) is a Persian verb that needs a subject and object.

(5.5) $xord =$



The application rule in CUG allows a noun functor *Ali* to be applied to the verb *xord* as its argument. If the two unify the result is a verb with *Ali* as its subject. The result is a verbal argument that can become an argument for another noun functor. The NP can combine with a verb either to the right or left of itself.

Karttunen handles long distance scrambling by using *functional uncertainty*[1]. This is specified in the category definition of the NPs. If we replace the feature value of [SYNTAX SUBJ] with [[SYNTAX VCOMP]* SYNTAX SUBJ] then the NP can be the subject of a verb which is embedded indefinitely many times inside the verb category. This notation will be explained later in Section 5.1.4. In this way some examples of long distance scrambling are achieved. But as Hoffman [1995] shows, the formalism cannot capture some examples of long distance scrambling and is not general enough. We will discuss Hoffman's extension to CCG in the next section.

[1]We will elaborate on Functional Uncertainty in Section 5.1.4.

CCG and extensions

Another extension to CG is Combinatory Categorial Grammar (CCG) [Steedman, 1987]. CCG has been developed to handle coordination and long distance dependencies without the use of movement rules and traces. Unlike CUG, in CCG the verbs are the functors and categories such as NP are basic category.[2] By means of a small set of combinatory rules, functions and their arguments are combined together. Among other operations, one can name *function composition*. The composition combinator combines two function categories together and the arguments of one are added to the end of the argument categories of another functor.

Hoffman [1992] presents a grammar of Turkish in CCG. But CCG has its limitation in capturing examples of long distance scrambling in Turkish. Hoffman [1995] argues that CCG should be extended for this purpose and describes various versions of CCGs and their limitations in capturing free word order phenomena.

> "Although the use of type-raised categories without variables, like the ones above, can handle local scrambling and long distance scrambling with one embedded clause, it cannot handle all word order variations in sentences with an arbitrary number of embedded clauses. "(Hoffman 1993, 22).

She also argues against encoding word order in the subcategorisation frame of the verbs and she proposes that the strict order of NP arguments in a verbal category (e.g. $S\backslash N_{nom}\backslash N_{acc}$) be relaxed. For this she extends CCG by allowing multi-sets of argument types, rather than just argument type. The relative order of categories inside these multi-sets can remain unspecified (e.g. $S|\{NP_{nom}, NP_{acc}\}$). In her multi-set extension to CCG (i.e. {}-CCG) she also extends the definition of function composition. In {}-CCG, when two functions are combined then the union of their argument sets is the argument set of the new function. Unlike CCG here the order is not relevant. An example of this is shown in the following:

[2]Note that in CCG it is possible in the lexicon for nouns to be type raised into functions. Steedman [1985] mentions that in languages with case marking, the case-markers may type-raise nouns into categories with grammatical relations.

(5.6)

Kitabi	[Fatma	[okudugumu]	saniyor]	benim.
book-acc	Fatma	read-gerund-acc	thinks	I-gen
N_{acc}	N_{nom}	$S_{ger-acc}\|\{N_{gen},N_{acc}\}$	$S\|\{N_{nom},S_{ger-acc}\}$	N_{gen}
		$S\|\{N_{nom},\ N_{gen},N_{acc}\}$		
		$S\|\{N_{gen},N_{acc}\}$		
		$S\|\{N_{gen}\}$		
		S		

'As for the book, Fatma thinks that I read it.'

In this example[3] the two verbs are adjacent to each other and by function composition they can be combined. The different stages of composition of functions and the combination of functions and arguments is depicted.

Hoffman argues that {}-CCG can derive a string of any number of scrambled NPs followed by a string of verbs. Here V_i subcategorises for NP_i.

$(NP_1...NP_m)_{scrambled}V_m...V_1$

This will over-generate for examples of Turkish. Consider another permutation of the sentence in (5.6).

(5.7)

*	Kitabi	[benim	Fatma	okudugumu]	saniyor.
	book-acc	I-gen	Fatma	read-gerund-acc	thinks
	N_{acc}	N_{gen}	N_{nom}	$S_{ger-acc}\|\{N_{gen},N_{acc}\}$	$S\|\{N_{nom},S_{ger-acc}\}$
				$S\|\{N_{nom},\ N_{gen},N_{acc}\}$	
				$S\|\{N_{gen},N_{acc}\}$	
				$S\|\{N_{acc}\}$	
				S	

'As for the book, Fatma thinks that I read it.'

Here *Fatma* from the matrix clause has moved into the embedded clause. This is not grammatical in Turkish, but as we have shown the {}-CCG recognises it. In general, solutions based on function composition will face this kind of problem. Rambow and Joshi [1994] (p. 50) refers to a similar problem for FO-TAG. They argue that an integrity constraint is required

[3]Note that we haven't shown the combinatory rules used in the examples.

that let elements exit from a constituent, but prohibits other elements from entering the constituent.

Hoffman also integrates a level of information structure (IS) – discourse functions: Topic, Comment and Focus – into {}-CCG. This level parallel to the syntactic level further puts restrictions on possible examples of scrambling in the system. It is doubtful that our previous counter-example could be ruled out by this level of IS, because the restriction belongs to the syntactic level of grammar.

Another problem with {}-CCG is that it only allows long distance extraction for the arguments of the verb and not for the adjuncts. This problem might be resolved if we consider those adjuncts as arguments of the verb. See Hoffman [1995](p. 47) for further discussion.

5.1.3 Extensions to TAG for Scrambling

Tree Adjoining Grammar (TAG) [Joshi et al., 1975] is a tree rewriting formalism that extends the domain of locality of context-free rules. TAGs are mildly context-sensitive grammars [Joshi et al., 1991] which consist of a set of elementary trees with two other operations, namely substitution and adjunction for deriving larger trees. These two operations replace a non-terminal node in a tree with another tree. The operations are depicted in Figure 5.1.

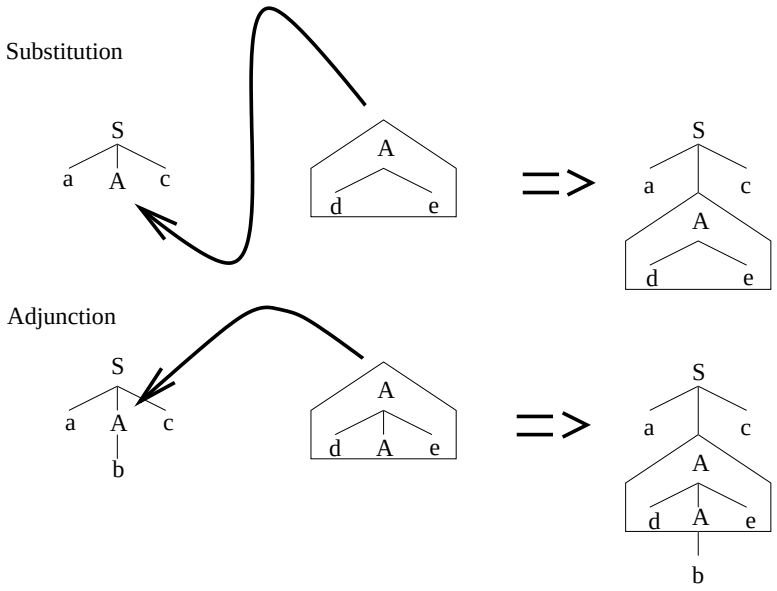


Figure 5.1: Substitution and Adjunction in TAGs

The substitution operation rewrites a node on the frontier of a tree, while the adjunction operation inserts an auxiliary tree into the middle of another.

By these two operations on elementary trees, TAG provides a framework which separates recursion and unbounded dependencies from the local dependencies (such as subcategorisation and wh-dependency). Becker et al. [1991] shows that if we want to enforce the constraint that a predicate and all its arguments occur in the same elementary tree (i.e. co-occurrence constraints) then TAGs cannot handle examples of long distance scrambling.

Different approaches have been proposed for extending TAGs to handle long distance scrambling. Rambow and Joshi [1994] reviews some of them.

Free-Order TAG (FO-TAG) [Becker et al., 1991] is an extended ID/LP version of TAG. In this framework, the elementary trees only indicate the dominance relations and do not specify the linear order among the head and its arguments. By a set of separate LP rules, the correct linear precedence is enforced. These LP rules can be specified for nodes occurring in the same elementary tree. There is also an integrity constraint which marks trees as islands. These islands disallow extraction of nodes from marked trees and act like barriers. The main problem with FO-TAG is the fact that leftward movement of NPs out of extraposed clauses is restricted.

Multi-Component TAG (MC-TAG) [Weir, 1988] is another extension to TAG. Unlike TAGs which consist of a set of *elementary trees*, MC-TAGs consist of a set of *sets of elementary trees*. There is also a difference in the adjoining operation. In TAGs we adjoin *an auxiliary tree* to another elementary tree, while in MC-TAGs we adjoin *all trees from an auxiliary set* simultaneously.

V-TAG is an extension of MC-TAG and can capture different examples of (long distance) scrambling. The introduction of set of elementary trees and sets of *sets of elementary trees* into TAG has introduced an additional complexity which has been avoided in other formalisms. Further integration of pragmatic information into V-TAG and consideration of performance[4] in parallel to competence is to be investigated. There remains the open question of whether for capturing free word order phenomenon we need such complex machinery. In the next section we will look at a simpler mechanism for this.

[4]See Super TAG for a framework for adding processing constraints to TAGs.

5.1.4 Lexical Functional Grammar

In Lexical Functional Grammar (LFG) [Bresnan and Kaplan, 1982], surface word order is encoded by C(onstituent)-structure. C-structure encodes dominance and precedence relations inside a constituent. C-structure is not used to encode grammatical relations. Instead the grammatical relations are encoded in the F(unctional)-structure. This mechanism for encoding grammatical relations is different from the mechanisms used in theories such as GPSG, HPSG, and CG. In LFG the grammatical relations are primitives (e.g. object, subject) rather than defined by position of arguments in a SUBCAT list. This way of encoding grammatical relations with F-structures in which the order of the arguments is not important provides a better solution for capturing scrambling of arguments inside a clause boundary (local scrambling). This is more suitable for capturing the grammar of free word order languages. The original formulation of LFG had problems in capturing long distance scrambling (movement of arguments across clause boundaries) and used C-structures to state generalisations about long distance dependencies.

[Kaplan and Zaenen, 1989] argues that long distance scrambling obeys functional rather than phrase-structure constraints. They propose an F-structure approach for representing long distance dependencies. They don't use mechanisms such as slash or gapping and their solution is based on the formal device of *functional uncertainty* for characterising systematic uncertainties in functional assignments. We will elaborate on this with an example:

A constituent in a clause might be the object (**OBJ**) of the clause in which it is located or if it is topicalised, it might be the object of the immediate embedded complement clause (**COMP OBJ**) or the object of any embedded complement clause (**COMP ... OBJ**). We don't know in advance which of these possibilities might be admissible and this depends on information that may be available arbitrarily far away in the string. Instead of formulating this infinite uncertainty by an explicit disjunctive enumeration, LFG uses a formal specification that characterises the family of all possible equations as a *regular expression* over the vocabulary of grammatical function names. For the above example the equation will be (**COMP* OBJ**). Here * is the Kleene star. This mechanism captures uncertainty using underspecification. The use of regular expressions for specifying the mechanism make it more general. It can potentially represent two levels of constraints on the uncertainty equation:

- Conditions on the potential functions at the end of the uncertainty path (the "bottom"

	Mechanism for LDS
V-TAG	D-Link
CCG	Function Composition
LFG	Functional Uncertainty
GPSG/HPSG	Slash Percolation+Liberation

Table 5.4: Formalisms: Long Distance Scrambling(LDS) and Probabilities

object in the previous example). In (**COMP* (GF - OBJ)**) the bottom can be any grammatical function(GF) except object (OBJ).

- Conditions on the potential functions in the middle of the uncertainty path (the **COMP*** in the previous example). For more examples see King [1993].

King [1993] is a recent work which discusses the syntactic representation of discourse functions of Russian in LFG. Russian is traditionally considered a free word order language.

To sum up, in the previous sections we gave examples of formalisms such as GPSG and HPSG (using ID/LP) and CG for representing the grammar of free word order languages.

Table 5.4 summarises the mechanisms for dealing with Long Distance Scrambling in some of the recent versions of the formalisms that we discussed.

We argued that the CCG mechanism for LDS is not capable to represent all constraints on scrambling, while the V-TAG mechanism for LDS is too complex. In the following we will concentrate on grammatical relations in these formalisms.

5.2 Discussion: Encoding of Grammatical Relations

The grammatical relations in theories such as GPSG, HPSG and CG were specified by associating grammatical relations with positions in the SUBCAT attribute associated with each predicate. The list represented by the value of the SUBCAT attribute encodes the unsaturated arguments of that category and the order of the list is important. In the literature this means of encoding is referred to as *hierarchical encoding* [Johnson, 1988]. The strict order of the arguments in the SUBCAT list creates problems for representing free word order languages and we saw that JPSG and Multiset-CG for Japanese and Turkish have relaxed the strict order of the arguments in the SUBCAT list.

Another type of encoding that we saw was in LFG. Johnson [1988] calls the way of encoding

in LFG, *direct encoding* and contrasts the two means of hierarchical and direct encoding. The main advantage of hierarchical encoding is the simplicity of the approach for representing the arguments of the predicate. The arguments are explicitly represented and the strict order of SUBCAT list is used in the parsing and attachment of the arguments. By contrast in direct encoding the arguments that a predicate can have should be specified with a extra mechanism. Either we need to use diacritic features such as transitive, intransitive or we should employ constraints that show the existence of an argument such as subject and object.

For free word order languages the direct encoding is more natural, while the hierarchical one needs to be extended to deal with free word order languages (JPSG and Multi-set CG). The hierarchical encoding faces another problem in representing verb final languages. Since all the information about subcategorisation is represented in the predicate, it is not possible to parse these languages with such grammars in an incremental and natural way, for example in applications like real-time parsing and translation of spoken language. No argument attachment can be done by the parser before the verb of the sentence is encountered. There have been works such as Konieczny and Hemforth [1994] for incremental parsing for HPSG, but it is argued in the literature that strictly head-driven models (such as HPSG) make wrong predictions for the on-line processing of certain verb-final clauses[5] [Bader and Lasser, 1994]. The incremental HPSG solution of Konieczny and Hemforth [1994] doesn't consider examples of long distance scrambling and it is very difficult to develop a fully incremental extension to HPSG for head final languages, as shown by Gungordu [1997][6].

In contrast, in direct encoding methods such as LFG, the use of grammatical relations for this and the notion of underspecification in functional uncertainty might be able to tackle the problem of argument attachment in a more natural way (especially for arguments which are long distance scrambled).

What will be the implications of these for Persian, a free constituent order language with SOV as a major word order? As we explain in this thesis Persian allows at the same time the extraposition of embedded clauses and long distance scrambling of constituents from some embedded clauses.

While the application of head driven approaches for Persian (as a verb final language)

[5] In Section6.2.1 we clarify the extent of psycholinguistic validity of our parser.

[6] Gungordu proposes a more powerful unification mechanism to be investigated for this purpose.

is very unnatural, it is not obvious whether employing functional uncertainty for immediate attachment of arguments which come before the verb is computationally advantageous.

Kaplan and Maxwell [1988] give an algorithm for functional uncertainty. Based on their results they conclude that it is advantageous to postpone functional uncertainty longer (than is absolutely necessary) to reduce the number of parses and increase the efficiency of the system:

> In particular, we found that if the uncertainties are postponed until predicates (semantic form values for PRED attributes) are assigned to the F-structure they belong to, the number of cases that must be explored is dramatically reduced.

Put it in another way, the introduction of functional uncertainty, at an early stage, adds to the number of parses which are generated and later discarded. In order to reduce this, one solution is to postpone the application of functional uncertainty to when the verb in the sentence is found and the subcategorisation information of the verb helps to reduce the number of uncertainties.

Having the functional uncertainty framework in mind, the major question is whether these findings are true for parsing different examples of local and long distance scrambling in Persian? Is their finding specific to their algorithm and language or linguistic theory?

An alternative approach is to have disjunctions instead of functional uncertainty equations and add possibility measures to these disjuncts. [Uszkoreit, 1991] is one of the works which discusses strategies for adding a control layer on top of declarative grammars for ordering the sequence of conjuncts and disjuncts. This extra control information adds performance models to the competence models without sacrificing their declarative nature. It suggests that in disjunctions, the disjuncts that have the highest probability of success should be processed first, whereas in conjunctions the reverse is true. For ordering the possible alternatives different static and dynamic measures can be taken into account.

An intermediate solution is to have a mixture of functional uncertainty equations and set of disjunctions which are augmented with possibility/probability measures. These can be ordered according to some contextual principles which are language specific. Although in our parser we have not considered the standard notion of probabilities, but we have used a notion of graded grammaticality that needs the introduction of these measures. In real world

applications, one cannot also ignore probabilities and a parsing system or formalism need to be powerful enough to be extended for this purpose.

In the next chapter we will have a closer look at an implemented system for representing examples of scrambling of Persian, and we will further elaborate on these issues. The parsing system is designed to analyse specific examples of local and long distance scrambling in Persian, nevertheless the system offers some parsing and linguistic generalisations which would be useful for processing other examples of scrambling in flexible word order languages.

Chapter 6

Parallelism and Parsing: A Competitive Parser

6.1 Introduction

In the previous chapter we reviewed some formalisms and systems which have been designed for representing the grammar of free word order languages. Each formalism tried to capture some examples of local scrambling or long distance scrambling. Some of the formalisms considered the role of discourse in scrambling and the fact that under a specific intonation, one word order may be more acceptable. Another issue which has not been thoroughly investigated is the notion of *acceptability* itself and implementing this imperfect notion for scrambling cases. Recently, notions such as probability, optimality, possibility, plausibility, acceptability and graded grammaticality have been incorporated into the linguistic theories. Despite the fact that scrambling and word order introduce a degree of acceptability and graded grammaticality, the necessary acceptability or plausibility notions have not been added to the scrambling rules.

In our study, we extend the word order rules by introducing a stochastic version of them. In this work, we use acceptability and plausibility interchangeably to refer to all these notions. We have only considered a limited subset of these and future work is needed to incorporate all these notions.

We have developed a framework with the aim that in the future we can add different aspects of *graded grammaticality*, ranging from fine-grained graded unification [Kim, 1994] to

more recent notions in syntax, such as Optimality Theory [Smolensky and Stevenson, 1997].

Modeling graded grammaticality has been neglected in much of the past work, despite the fact that Chomsky has attempted at various times (e.g. Chomsky [1964]) to incorporate it into a model of linguistic competence. However, graded grammaticality is now a challenge for any formalism and theory that wants to account for the representation and processing of natural languages. Is graded grammaticality part of competence, performance or both?

Graded grammaticality and its interaction with word order constraints have been studied from another perspective in *performance* models for languages Hawkins [1990], Kirby [1999]. But the main problem is that not much significant theoretical work has been done to incorporate graded grammaticality in a unified model of competence and syntax. The lack of methods for gathering data and formal models of graded grammaticality has also complicated the problem.

Keller and Alexopoulou [1999] shows that grammatical judgments are replicated by different speakers of a language. Their work deals with grammaticality judgments, their elicitation and their use as evidence in linguistic theory. It is now possible to obtain and elicit these judgments by psycholinguistic tests. Computational methods are needed to take these grammaticality judgments into account.

One of the pioneering works on graded grammaticality and word order is [Uszkoreit, 1985]. He proposes a framework for representing the flexible word order of German by introducing complex LP constraints that take into account different levels of grammaticality of examples in German with scrambling. How graded grammaticality is implemented by complex LP constraints is not further discussed in the work. To fill this gap in research in processing graded grammaticality and its effect on word order, we have implemented a parser that we will elaborate on in this chapter.

In our implementation, we are looking for a more economical representation and an alternative to complex LP constraints which is suitable for processing Persian and can take into account grammatical gradedness [Rezaei, 2000]. The study contributes to a better understanding of the problem of parsing and representing free word order languages.

What makes the study more interesting is that there are different levels of gradedness and ambiguity in the grammar of Persian that interact together. As we explained in Chapter 2 the subject and object in a sentence can be missing (i.e. pro-drop property) and subject

and object marking is ambiguous in some cases. The notion of *specificity* which is a graded notion in Persian plays an important role in the disambiguation between subject and object in Persian[1]. The gradedness of specificity has also been investigated in [Kluender, 1992], [Keller, 1996].

So modeling graded grammaticality becomes essential and this interacts with the word order rules. As we will discuss in this chapter, the interaction between graded grammaticality and scrambling in Persian can become more complex, especially when one tries to deal with the interaction between control and scrambling.

In our work we do not consider the ambiguity arising from using Persian script (e.g. lack of *ezafe*[2]), but some of the results of the work on modeling graded grammaticality can also be used to restrict that kind of ambiguity too. In our work, we also do not discuss the experimental methods for deriving linguistic acceptability using experimental methods in psycholinguistics (e.g. Keller and Alexopoulou [1999]). Our focus will be on designing a computational architecture that incorporates these acceptability measures and future experimental work is needed to derive these values.

In a computational framework, one can model graded grammaticality as a form of competition among a set of alternatives with different degrees of grammaticality. In a competition framework, the result depends on the entities that are taking part and violation of the principles of the grammar reduces the graded amount of grammaticality (i.e. acceptability) for each. Competition in a grammar can arise for acquiring the highest degree of grammaticality among a set of plausible interpretations, but competition can also arise for limited linguistic *resources*. What are these resources and are there specific principles in languages that put further restrictions for acquiring these resources? We will answer these questions in the specific domain of modeling Persian and the scrambling in its word order.

For the grammar of Persian, we apply recent competition-based approaches in such a way that the possible grammatical functions which could be assigned to a constituent compete with each other, while the scrambling constraints and their plausibility restrict the possibilities. But what additional machinery is required to represent such constraints without sacrificing the efficiency of the processing system? What kinds of frequency data relevant to scrambling

[1] We discussed specificity and gradedness in Page 10 and Chapter 2.
[2] See Chapter 2.

do human beings keep track of?

The relevance of competition and scrambling is not restricted to local scrambling and for long distance scrambling cases the possible word orders can also compete with each other. This is especially true for languages such as Persian and Japanese in which pro-drop can occur extensively and where subjects and other constituents in a sentence can be empty. For each constituent, the parser should take into account that the constituent can be attached locally or non-locally and this adds more inefficiency in terms of space and time for a parsing system. Having a competition framework in mind to some extent solves this problem, but as long as one doesn't have a set of criteria for restricting the possible alternatives in each step of processing, the mechanism is doomed to failure.

In this chapter we will look at these issues and by introducing competition and parallelism at the same time, we avoid some of the problems of backtracking and the inefficiency that it causes. We will further investigate linguistic limitations which one can impose on the processing architecture to restrict some of the possible alternatives. For this purpose we turn to recent proposals for adding resource limitation strategies to the processing [Johnson, 1996].

Over the last few years a different conceptualisation concerning 'resource sensitivity' has emerged in several disciplines connected to the study of language. This idea has been explored within categorial grammar in [Carpenter, 1996] and [Morill, 1994]. More recently Johnson [1997a] and Johnson [1997b] introduce a resource-based conceptualisation of LFG. In [Johnson, 1996] the approach is illustrated with a view of characterising constructions in terms of 'plugging'. A set of objects are constructed and some of these objects need to combine with other objects to become saturated, and rules determine what can be 'plugged into' what.

Phenomena such as argument attachment in natural languages are inherently resource based and most linguistic theories use some mechanism of resource sensitivity for argument attachment. We will consider competition for these grammatical resources.

The parsing model that has been implemented in this thesis is a parallel and concurrent extension of the parsing models that we studied in Section 4.2 and Section 4.3 of Chapter 4. It is another two-stage model, but the implemented parser is a parallel pipeline of two stages.

We will first investigate the application of techniques in parallel processing and parsing for this purpose. Then we will explain the rules, the different types of constraints for local and long distance scrambling and the details of the system. Finally we will discuss some

of the major design issues for implementing the competition strategies and will contrast our approach with more recent work in this area.

We will delay the more formal motivations and the dynamic infrastructure of the system to future work (in the next chapter) after the general approach is illustrated in this chapter.

6.2 Parallelism, Parsing and Linguistic Representation

6.2.1 Parallelism: An Introduction

There has been a growing interest in using parallel processing techniques for implementation of programs to simulate the intelligent activities of human beings.

The recent success of powerful chess machines like Deep Blue in defeating Kasparov, the world chess champion, doesn't lie in the fact that these programs simulate the behavior of an intelligent chess player. Their success lies in using massively parallel programs to defeat the highly efficient pruning and prioritizing mechanism of human brains.

In processing natural languages, humans are incredibly powerful in bringing all kinds of information — phonetic, semantic, pragmatic, syntactic constraints as well as knowledge of the world and the situation — to prune the huge search space of possibilities and disambiguate a sentence or utterance. The more constraints are added to the picture the better a human parser disambiguates an utterance.

Using parallelism and competitive methods can be seen as an artificial counterpart to this efficient natural mechanism for processing languages. Here, our goal is not to present justifications from psycholinguistic research for using parallelism, rather to use parallelism to help us in processing languages by machines which lack that efficient and natural mechanism. Nevertheless some of the techniques that we use in a parallel competitive framework might be useful in constructing psycholinguistic models.

But at what level of representation should parallelism be used and at what level of detail should we introduce parallelism in order to avoid unnecessary complexities? In other words how can we employ parallelism to be a help and not a burden in language processing?

6.2.2 Parallelism in Processing Languages

Many different models for parallelism have been proposed for language processing and in this section we will refer to a limited set of them. We will specifically look at parallelism at the knowledge level (macro-level) and parallelism inside the grammatical levels (micro-level).

One dimension for introducing parallelism is at the knowledge level where different knowledge sources for phonology, morphology, lexicon, syntax, semantics and pragmatics can interact with each other. During parsing, a system based on this task-oriented framework is capable of using any type of knowledge and the processing is not restricted to a sequence of non-interacting modules, as suggested in Figure 6.1.

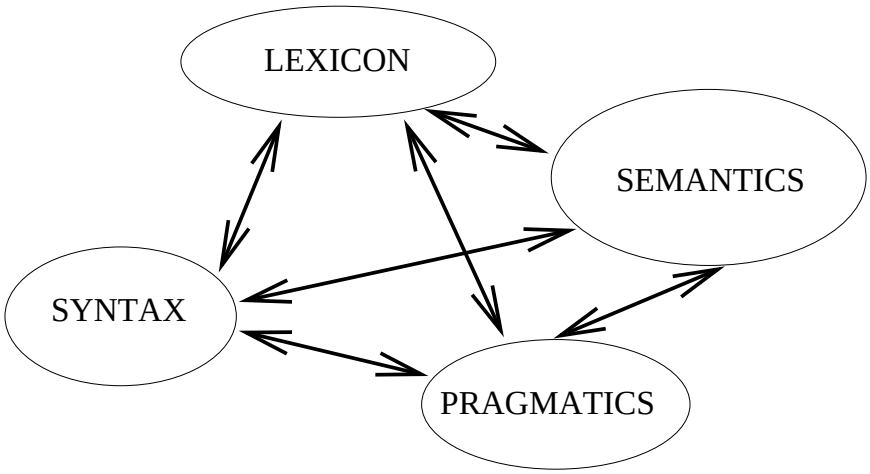


Figure 6.1: Parallelism as Interaction

A proposal for such model can be found in [Winograd, 1972]. This kind of interactive model of language processing, referred to as *heterarchical*, may become very complex and the need for specifying the communication and interaction between any two knowledge sources has motivated approaches in which a common module or data structure have been used for handling interaction between modules.

In Blackboard models (Figure 6.2) the multiple knowledge sources can progress in parallel and the commonly accessible architecture for the blackboard provides the means for cooperation between the parallel modules. All the communication and interaction for communication of intermediate results are routed and handled by the central and global blackboard. The main example of a blackboard system for language processing is the HEARSAY-II speech

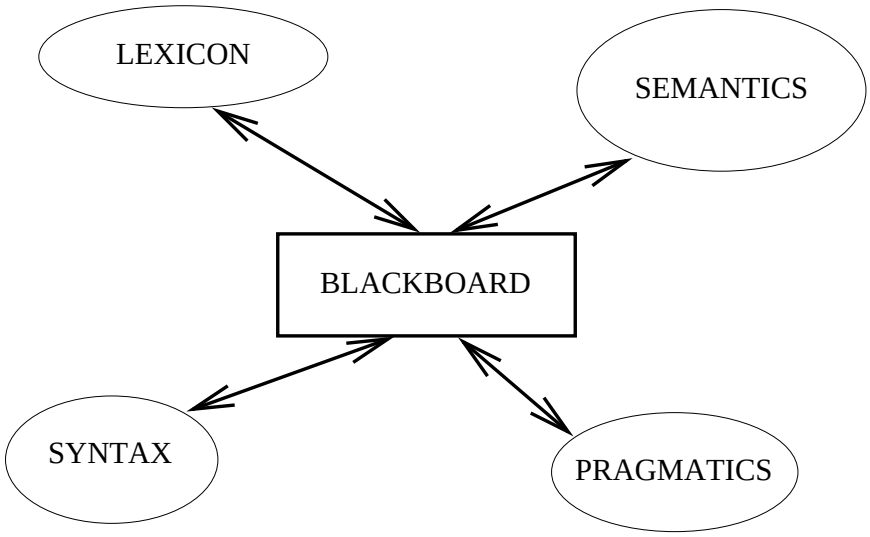


Figure 6.2: Blackboard Model

understanding system [Erman et al., 1980].

The blackboard models were developed to reduce the complexity of the heterarchical parallel models, but the potential parallelism provided by each of these knowledge sources looks rather small. This is because the knowledge sources are dependent on each other and should wait for each other and at any time each knowledge source can only see a portion of the blackboard, otherwise the system performance would degrade too much. A more recent example of a blackboard system is ANGEL [Bisiani and Forin, 1989] which make use of parallelism and pipelining to recognise speech.

A simpler approach to parallelism is to run a sequence of tasks in parallel as a pipeline or cascade of stages (see Figure 6.3).

Different subtasks can run in parallel, but the information flow in such a pipeline is serial and it is from one module to another. If all the stages in a pipeline are run in parallel and the communication cost/time is negligible between the stages, then the maximum speed of the pipeline could not be increased more than the speed of the slowest stage in the pipe (plus a constant delay time for the first line to appear in the output of the tokeniser). If we have a pipeline of stages that each performs a process on an input word or item, then for processing T words, the first word will take time equal to the sum of all stages. The subsequent ones emerge after the intervals of Max(P) where P is the time for each process/stage in the pipeline

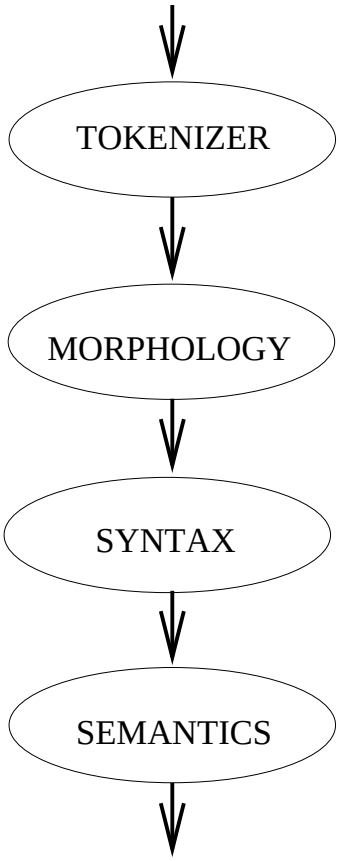


Figure 6.3: Pipeline Model

to complete. Max(P) is a function that returns the time for the process that is the slowest (i.e. the process that requires more time to complete its work, compared to the other processes in the pipeline). Hence creating smaller units can potentially increase the speed, if at the same time communication time/cost can be decreased.

These pipeline models can be extended to have feedback from a stage backward to the input of an earlier stage in the pipeline. A good example for pipeline parallelism is cascaded ATN models proposed by Woods [1980], Christaller and Metzing [1983].

So far we have concentrated on parallelism at macro-level. But one can also introduce parallelism at a finer granularity (i.e. micro-level) and introduce parallelism at different levels of grammatical representation, such as inside syntax and semantics.

[Huang and Guthrie, 1985] is an example of a model which mixes the two kinds of parallelism at the knowledge macro-level and the grammatical micro-level. In their model (Figure

6.4), two syntactic and four semantic processes interact. The two syntactic processes are used for constructing S(entence) and NP in parallel. The semantic processes are used for tasks such as finding meaningful adjective-noun (AN) word sense pairs, subject-verb (SV) word sense pairs and verb-object word sense pairs (VO). These processes will constrain the structures build by the NP and S processes.

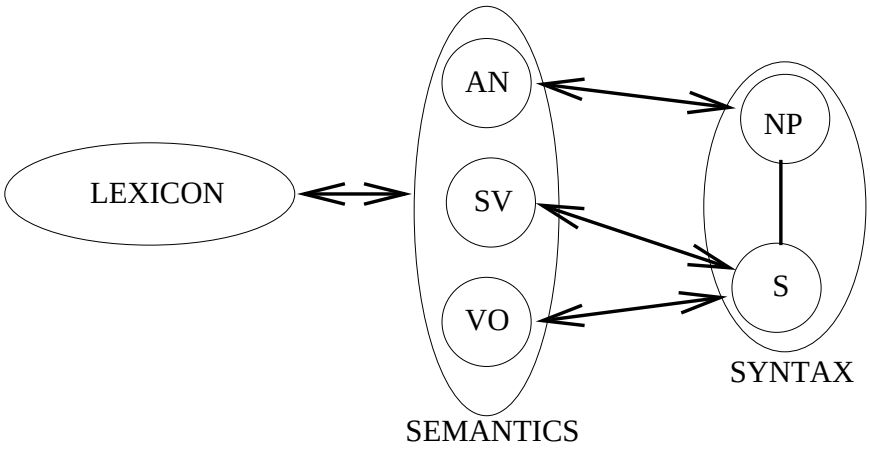


Figure 6.4: An Example of Semantic and Syntactic Parallelism

This micro-level parallelism has also been applied to linguistic theories and frameworks. For example, in GB, the modules can run concurrently and communicate with each other, e.g. Figure 6.5.

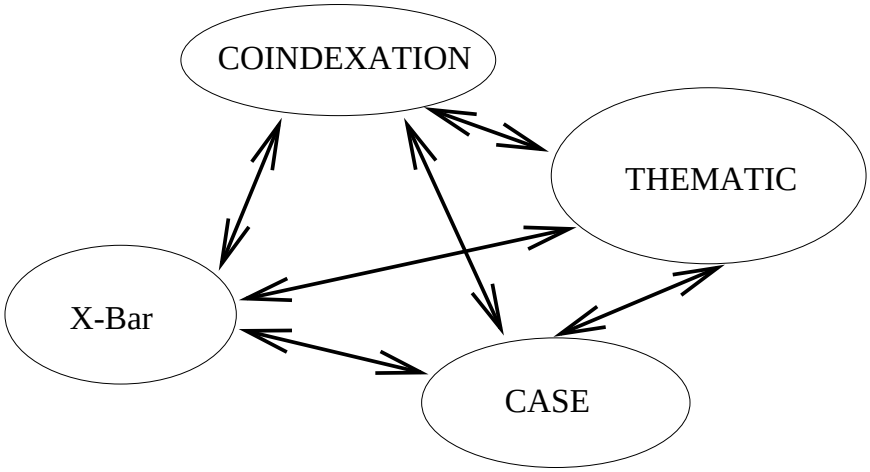


Figure 6.5: Parallel GB

Examples of this level of parallelism have been implemented in [Kuhn, 1990] and [Crocker, 1992]. The introduction of parallelism does not necessarily introduce higher speeds and in some cases the unwanted complexity of dealing with parallelism restricts the use of parallelism at micro-levels.

Parallelism has also been introduced for grammatical rules. The set of grammatical rules can be viewed as a network of agents or objects working concurrently. Each occurrence of a terminal or non-terminal symbol in the grammar rules corresponds with an agent with modest processing power and internal memory. The agents communicate with one another by passing subtrees of possible parse trees [Yonezawa and Ohasawa, 1988]. Chart parsers can be considered as serial implementation of such approaches. Parallel implementations of chart parsers such as [Trehan and Wilk, 1988] [Thompson, 1991] illustrate this approach.

Finally, a fine-grained notion of parallelism is introduced in connectionist or neural network (sub-symbolic) approaches. Language processing in this approach is coded into spreading of activation and converging of activation towards a pattern that represents the meaning of the sentence; ([Sharkey and Reily, 1992], [Jain and Waibel, 1991] and [Stevenson, 1994]).

In the following section we will have a closer look at this issue and some examples of agent (process) based approaches.

6.2.3 Parallelism: What Granularity?

In our model we have mainly concentrated on approaches which do not require complex coordination techniques such as Blackboards. Here, our goal is not to present justifications from psycholinguistic research for using parallelism, and instead we focus on approaches which improve the efficiency of the parsing system. In the following we will look closer at different levels of parallelism inside syntax. In some approaches words are considered as processes, while in others finer-grained objects such as features or more coarse-grained objects such as phrases are considered as the appropriate level of parallelism.

[Trehan and Wilk, 1988] is one of the approaches which attempts to introduce parallelism in parsing. Trehan *et al* have implemented a chart parser in a parallel environment. For this purpose they treat incomplete phrases as active processes which are looking for inactive processes (i.e. completed phrases on the left-hand side of the rules or words). For example in 6.1:

(6.1) $VP_2 \rightarrow VP_1$. NP
$VP_2 \rightarrow VP_1$ NP.

The first shows a VP_1 process which is an incomplete edge and is looking for an NP to become completed. After attachment of an NP to VP_1, a VP_2 will be generated as a completed process. This is illustrated by the second rule. Trehan *et al* use Context Free Grammar (CFG) rules for expressing the relationship between processes. In their approach the phrases are treated as processes, but the channels of communications between processes are not part of the linguistic theory and the existence of channels in the implementation is an implementational issue and is specific to the parser architecture. The parallel implementing of a chart parser in this way does not improve the speed of the system very much, and since the system is implemented in Parlog, it is hard to extend the rules with feature structures.

Trehan *et al* uses a notion of parallelism based on the actor model of computation [Agha and Hewitt, 1987]. This model combines object-oriented methodology with concurrency and distribution. The model assumes that a collection of independent objects (actors) communicate via asynchronous message passing. In this model a process can be thought of as an object with a state that can be changed by the process. For changing the state of an object a message can be sent to that object and an object may send messages to other objects. Objects can create instances of themselves or different objects.

ParseTalk [Broker et al., 1994] is a recent parser designed for analysing texts and based on the actor model. In ParseTalk each word evokes a process, and hence a sentence evokes a set of communicating processes. Each process is connected with its neighbors through channels and may communicate with them. The system parses a sentence by establishing a dependency tree incrementally and attachment of the words to the tree is achieved by message passing.

ParseTalk uses a dependency oriented framework as its grammar, which is fully lexicalised. But the distinction between procedural and declarative knowledge is not very clear and the system falls short in dealing with word order constraints properly. While ParseTalk claims that it does not use any rules, but it argues that it uses ID/LP format for dealing with word order which is very confusing. In an ID/LP notation, one separates the dominance and

precedence relations. Unlike CFG rules, in ID (Immediate Dominance) rules, the elements in the right hand sides of the rule don't specify precedence relations and the order is specified by separate LP (Linear Precedence) principles [Gazdar et al., 1985]. ParseTalk does not discuss how it can handle examples of long distance scrambling and its constraints.

[Fujinami, 1996] is another recent process based approach to language analysis. Fujinami proposes another actor based model. He represents objects of situation semantics in π-calculus. By using channels of π-calculus, he models different levels of grammar, from feature structures to phrase structure. He does not commit himself to any specific syntactic theory, but he uses constituents such as NP which are created as processes in his model. One of the major aspects of his work is that he tries to represent feature structures as process structures and each feature value in a feature structure is represented as a process in parallel with other feature value processes in the feature structure. In a channel notation, he manages to tackle the problem of shared structures inside a feature structure, but his formalism is not general enough and does not allow unification of DAGs. It is not possible to unify two DAGs if the result of unification adds new feature value pairs to the result. A general criticism to representing feature value pairs as parallel processes is that for unification of two feature structures (DAGs) in parallel, we need a level of synchronisation and restriction of parallelism (e.g. by locks, monitor [Hoare, 1973]) to ensure that the unification of two process-feature structures yields the same result as the unification of two normal feature structures. This added complexity makes the introduction of parallelism at the level of feature values very unlikely.

To sum up, we have looked at three different approaches for introducing parallelism in a grammatical framework. In Trehan *et al*, the level of granularity was phrase level (for active processes) and the system used processes that communicated through two general channels. The channels didn't correspond to the grammatical entities. In contrast in ParseTalk the granularity was at word level and the word processes communicated with their neighbors. Again the communication channels didn't correspond to any notion in grammatical theory. Finally Fujinami introduced a finer level of granularity and features were considered as processes and could communicate with each other. In addition, Fujinami regards (grammatical) relations as processes.

We argued that employing processes for representing feature values in feature structures introduces unwanted complexity to the framework and hence a coarser level of granularity should be considered (e.g. word level). Unlike ParseTalk in our framework we assume constituents (e.g. NP) for clustering words into process structures. In the next section we will look at the details of the parsing system and will further elaborate on the interaction between communication and competition in the parsing domain.

6.3 A Pipeline Parser

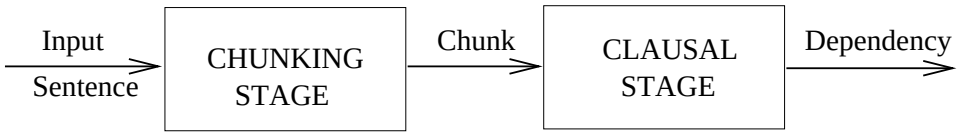


Figure 6.6: Parser Modules

Based on the grammar of Persian and previous experience in parsing Persian by PATR-II [Rezaei, 1993] and [Rezaei and Crocker, 1995] we have implemented a two level parsing system, illustrated in (6.2).

(6.2) **Main Body:**

PAR (run in parallel)

a. Parse-chunk(Pipe) to read a word and output a chunk on the pipe.

b. Parse-clause(Pipe) to read a chunk from the pipe and output dependencies.

Description:

Pipe is the Linda communication pipeline linking the two modules.

The first level of the parser, which is a variant to the PATR-II system, groups the words of the sentence into chunks: NP, PP, V and Comp using context free phrase structure rules. As soon as a chunk is found (in a) it is passed to the second level of parser (in b). The two stages are run in parallel. Abney [1996] uses a similar notion of pipeline parsing. He refers to the first stage as *chunk level* and to the second stage as the level of *simplex clauses*. Abney uses a finite-state cascade and his system uses finite state models for grammatical representation

at both stages. Instead of finite-state models we have used an extension to CFG rules in the first stage and regular grammars for the second stage. CFGs are more flexible and powerful in representing constituents with levels of recursion. We have also introduced a look ahead for these rules at the first stage.

For representing scrambling we have extended the regular grammar rules for clauses with a special path set that keeps record of possible interpretations for the arguments of the clause. This is in contrast to our previous approach [Rezaei, 1993] where we used bottom up parsing with extended ID/LP notation. This path set is used to represent competition for grammatical functions and backtracking is avoided. It is updated incrementally.

For example if the first constituent can be attached to the clause as SUBJect and OBJect, and if the next constituent can be attached as both SUBJect and OBJect, then the path set will include all possible combinations of: [SUBJ.SUBJ, OBJ.SUBJ, SUBJ.OBJ, OBJ.OBJ]. Some of these possibilities are restricted by the use of *word order constraints*. In this example SUBJ.OBJ is referred as a *path*. Each path in the path set has an activation or possibility value attached to it which shows the plausibility of that particular path relative to the others. The value corresponding to each path in the path set is calculated based on word order constraints and the numeric values considered for each word order constraint.

In other words in our framework, the word order constraints are defined locally to a clause and not for rules, and they specify the precedence relations between two grammatical functions. The precedence relations are probabilistic and each possible word order has a probability measure attached to it.

The word order constraints are of two types: hard and soft. The hard constraints cannot be violated, while the soft ones can be violated. The violation of a hard constraint makes the corresponding path inactive, while the violation of a soft constraint reduces the level of activity of that specific path. For simplicity we assume that the activity level is the same as a probability number.

In the following we will explain the details of the system and we will elaborate on hard and soft constraints that put restrictions on these alternatives (paths).

6.3.1 First Stage

For parsing the phrase structure rules of the grammar we have used a Prolog implementation of the standard version of PATR-II[3]. The extension is illustrated in (6.3) which is a recursive call to parse-chunk. Parse-chunk consists of a set of alternatives.

(6.3) **Main Body:**

parse-chunk(pipe)

If 3 top elements of the stack match the RHS of a rule:

i. Replace them with a new edge (i.e. LHS) in the stack.

ii. Parse-chunk the remaining sentence with the new stack.

Else If 2 top elements of the stack match with the RHS of a rule:

i. Replace them with a new edge (i.e. LHS) in the stack.

ii. Parse-chunk the remaining sentence with the new stack.

Else If the top element of the stack matches with the RHS of a rule:

i. Replace it with a new edge (i.e. LHS) in the stack.

ii. Parse-chunk the remaining sentence with the new stack.

Otherwise If a complete chunk can be formed from top of stack:

i. Remove the top of stack and output the new chunk to the Pipe.

ii. Parse-chunk the remaining sentence with the new stack.

If a new word can be shifted to the stack then shift and continue parsing.

Else (end of sentence detected) terminate.

Description:

The parser either matches the top of the stack with the Right Hand Side (RHS) of a rule or it reads a new word or it terminates. The stack is initialised with the input words. We have extended the PATR algorithm with a part to output a chunk, when the chunk is formed.

We will first review a simple example of parsing with numbers and further details. The input sentence is *ali seab xord*.

[3]See [Gazdar and Mellish, 1989] for further discussion on PATR.

(6.4) ali seab xord.
Ali apple ate
'Ali ate an apple.'

1. Dictionary look up:
   Input: `[ali, seab, xord].`

   Output:
   `Noun(ali,3,80), Noun(seab,3,20), Verb(xord,3,`$< Obj, Subj >$`).`

2. phrase chunking: (bottom-up)
   Input: `Noun(ali,3,80), Noun(seab,3,20), Verb(xord,3,`$< Obj, Subj >$`).`
   Output:
   `NP(dp(ali),3, ▷ obj:20 ◁, ▷ subj:80 ◁)`
   `NP(dp(seab),3, ▷ obj:80 ◁, ▷ subj:20 ◁)`
   `verb(verb(xord),3,` $< Obj, Subj >$`, 100)`

Noun(ali,3,80) gives this information about *Ali* that is 3rd person singular (3) and has a specificity of 80. ▷ gram-func:activation ◁ such as ▷ obj:20 ◁ shows a pair of grammatical function and activation value. Each constituent (chunk) may have one or more number of these pairs. The number indicates the plausibility of that grammatical function for the constituent. The verb entry also shows that the verb has an object and subject and is third person singular (3). We have used an activation value of 100 to raise the activation of clauses that have verb, compared to those which lack one and are not completed. Note that we have assumed no ambiguity for the verb and hence the activation value here reflects the notion of possibility of this interpretation.

The difference between an NP and DP is that NP is a fulfilled noun phrase (marked with a preposition or postposition or a null-marker[4]). At this stage we specify for each marked NP the possible grammatical functions that it can accept. The numbers after the grammatical

[4]In other words a phrase boundary is detected.

functions correspond to the possibility of that alternative. These numbers are derived from the specificity value of a noun and the presence or absence of *ra* after the constituent. For example in the above *Ali* is a proper noun and, as discussed in Section 2.2, it is specific. Since it is not marked by the *ra* specificity object marker, its object value is low (20%) and its subject value is high (80%). For NPs which are not marked with *ra* we have considered subjecthood equal to the specificity value and objecthood = 100 – specificity-value. We have used a numeric value for specificity because specificity of a phrase varies over a non-discrete range. In the absence of a corpus for deriving the probabilities of words and their co-occurrence we have used this notion to initialise the activation value, because we mainly use it for subject-object disambiguation which relies on specificity.

In contrast *seab* 'apple' is not a proper noun; and as it is not marked by *ra*, it can be either subject or object. For objects like *seab* the subjecthood value of 20% and objecthood of 80% have been considered. This is because the corresponding specificity value for *seab* is 20. Note that one can consider different numbers, but the choice of numbers and their relation with specificity and object marking by *ra* should be taken into account. We discussed this in Section 2.4.3.

In languages with a more fixed word order, such as English, syntactic parameters are more relevant, while for Persian and other free word order languages, the combination of semantic, syntactic, pragmatic parameters should be considered from the beginning. We discussed this issue earlier in Chapter 5.

The Phrase Structure Component

The constituents with internal rigid word order have been implemented by the use of phrase structure rules of PATR-II (i.e. $\rightarrow$ rules). This includes noun phrases, prepositional phrases (and verbs).

DP in our grammar is a noun phrase that is not 'marked' yet. Its structure is shown next:

In the above {...} shows zero or more number, and (...) shows optionality. There are three possibilities for marking a DP :

- Marking a DP by a preposition to get a prepositional phrase:

  PP $\rightarrow$ Prep DP

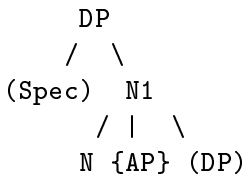


Figure 6.7: Structure of DP

- Marking a DP by a null marker[5] to get a complete noun phrase:

  NP $\rightarrow$ DP

- Marking a DP by specificity marker *ra* to get a complete noun phrase:

  NP $\rightarrow$ DP ra

By marking DPs we also assign possible grammatical functions (such as Obj, Subj) that NPs can accept. We have considered numerical values for specificity of NPs in Persian. As we explained, the combination of specificity and *ra* specifies the possibility that an NP be object or subject. Specificity was explained in Chapter 2. A summary of the phrase structure rules of the grammar is shown next. We discussed the structure of constituents in Persian in Section 2.3. For further details see Samiian [1983].

[5]The absence of ra or ezafe is considered as a null marker. This is implemented by the special look-ahead mechanism in our deterministic model for chunking.

(ps-1) ADJP → ezafe ADJ [ ]

(ps-2) ADJP → ezafe ADJ ADJP [ ]

(ps-3) DP → pronoun [ ]

(ps-4) DP → N1 [ˆ ezafe]

(ps-5) DP → SPEC N1 [ˆ ezafe]

(ps-6) N1 → N [ ]

(ps-7) N1 → N1 ezafe DP [ ]

(ps-8) N1 → N ADJP [ ]

(ps-9) PP → PREP DP [ˆ ezafe]

(ps-10) SPEC → Det [ ]

(ps-11) NP → DP [ˆ ra,˜ ezafe]

(ps-10) NP → DP ra [ ]

(ps-11) V → V [ ]

Our goal in designing the phrase structure (PS) component of the parser was to parse the input string into chunks and pass these chunks to the next level of parsing. By using the parallelism concept of Linda [Carriero and Gelernter, 1989], the interface between the two stages is implemented.

Linda is based on tuple space model of parallel programming. Processes can communicate with each other by sending or receiving messages as tuples through a shared tuple-space. In this model a few tuple-space operations are added to a base language (e.g. Prolog) to yield a parallel programming dialect. Due to sharing a single tuple space, the approach is not very efficient for cases where different processes want to communicate with each other. But in the case of a pipeline coordination (one producer-consumer pair), it is one of the simplest approaches. In our model the chunks are transmitted as Linda tuples between the two stages.

To restrict the creation of unwanted chunks, we have also added a look ahead item to the CFG rules. This makes the domain of the grammar that we have considered deterministic. For example in the PP rule we look ahead for one item, if the item is ezafe the PP will not be generated until the next item is not ezafe. This is shown next:

PP → Prep DP [ˆ ezafe]

The look-ahead list can contain zero, one or two items. Unlike conventional use of look-

ahead where the parser expects to see the look ahead item as the next item to be parsed, in our system, we have used an opposite notion of look-ahead where the parser make sure that the look-ahead item *does not* appear as the next item.

Note that the employed pipeline parallelism is useful especially when the chunking is not deterministic, which is another extension that we have not considered, but as we explained earlier the overlapping of the chunking and next module increases speed even for the deterministic chunking.

6.3.2 Parsing Stage II

At this stage, the constituents of a clause are assembled. This stage is run in parallel with the first stage and as a chunk is produced in the first stage, the attachment of it to the clause will be started. In other words, the second stage processes the chunks incrementally. The grammatical knowledge at this stage is represented procedurally and the parser gets the incoming chunks (from previous stage) and adds them to the clause that it is currently processing. The parser builds the main clause first and then the embedded clause and in this restricted sense, the parser works top-down.

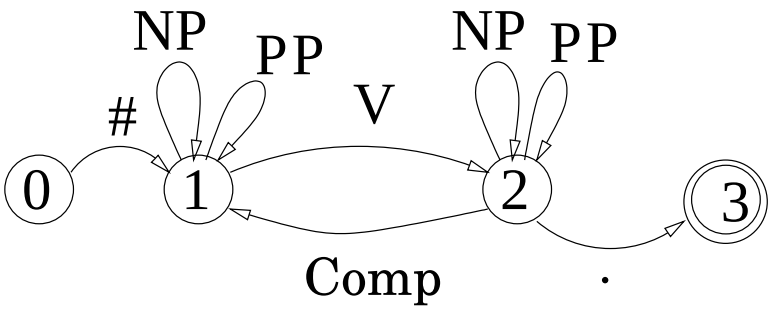


Figure 6.8: Second Stage

Depending on the incoming chunk, there are four different cases. The finite state model of this module is shown in Figure 6.8. The detail is further illustrated declaratively in Table 6.1 and the algorithm is shown in (6.5).

In the first three cases (in Table 6.1) the chunk will be added to the present clause and the parser continues with reading the next chunk and adding it to the present clause. In

PS rules
(ps-12) Clause → PP Clause
(ps-13) Clause → NP Clause
(ps-14) Clause → V Clause
(ps-15) Clause(Export) → Comp Clause(Import=Export)
(ps-16) Clause ([]=Export)→ '.'

Table 6.1: The Procedural Rules in the Second Stage

the fourth case, the parser spawns a new clause and initialises the variables of the clause with chunks that can be exported into it. In this way the parser represents long distance scrambling and control. When the parser reaches the end of the sentence, the work of the parser is completed. This will be illustrated in Section 6.4.1. The procedure implemented is as in 6.6.

(6.5) **Main Body:**

parse-clause(Pipe)

a. Initialise a new Clause

b. Input a Chunk from the pipe.

c. Do while Chunk not end of sentence.

i. If Chunk is a complementiser: attach-new-clause(Clause).

ii. Else If Chunk is a phrase: attach(Chunk,Clause).

iii. Input a new Chunk from the pipe.

d. Call Commit(Clause,Export)[6].

Description:

Depending on the input chunk we will have three cases: (1) an embedded clause is formed. (2) a phrase is attached. (3) end of sentence causes the termination of the program.

Note that this does not necessary mean that processing of a main clause is never resumed once attention shifts to a subordinate clause. Because when the work of an embedded clause

[6]Commit is called to generate dependencies and is described in (6.11). At the end of sentence Export must be empty.

is finished, the control will be passed to the level that has spawned it, until it reaches the main clause. Because of this, it is possible to extend the parser to deal with embedded relative clauses. It is also possible to do some checking inside a clause, once we exit from an embedded clause inside it.

(6.6) **Main Body: Attach (version 1)**

attach(Chunk,Clause)

a. If Chunk is NP:

i. Add grammatical functions of the Chunk to the Path-set of Clause.

ii. Use Apply-filter to impose word order constraints on the new Path-set.

iii. Update Export-set (and Import-set) for long distance scrambling.

b. If Chunk is PP:

i. Add the grammatical function of the Chunk to the Clause.

ii. Block ungrammatical parses that violate the word order Principles.

iii. Update Export-set (and Import-set) for long distance scrambling.

c. If Chunk is Verb:

i. Add the subcategorisation frame of the Chunk as subcat-resources of the Clause.

Description:

The program adds new chunks and makes sure (by Apply-filter) that the ungrammatical paths are removed from the Path-set. It works in two stages of GENerating all possible paths from combining each possible grammatical function of the chunk with the Path-set, and then shrinking the Path-set to a smaller one which respects the word-order constraints and performance principles (such as RLP and RBP[7]). If the chunk is a verb, then the subcategorisation frame of the verb is added to the clause as the expected resources of the clause. The Export-set and Import-set are used for long distance scrambling (LDS) and will be explained later in the LDS section.

For representing local scrambling, we have used the notion of the path set. This notion

[7]We will elaborate on these principles later in Section 6.4.1.

allows one to have competing alternatives of plausible word orders and rank them according to some constraints.

Implemented	Constraint	Explanation
Yes	precede(obj,subj, 0.90)	Subjects normally precede objects
Yes	precede(v,obj, 0.20)	Objects in most cases precede verbs
Yes	precede(v,subj, 0.20)	Subjects in most cases precede verbs
No	precede(obj,topic, 0)	Topics always precede objects
No	precede(obj2,obj, 0)	Object always precede object2
No	precede(obj2,subj, 0)	Subjects always precede object2

Table 6.2: Precedence Constraints in the Second Stage

The word order constraints that we have considered are listed in Table 6.2. The word order constraints are designed to reduce the activity of those alternatives which deviate from the canonical word order. A zero in the precedence constraint imposes a hard constraint to filter out illegal word orders[8]. A non-zero value imposes a soft constraint to reduce the activation value for non-canonical word orders.

We will first discuss examples of local scrambling and then we will deal with long distance scrambling and control in embedded clauses.

6.4 Parsing Local Scrambling

The constituent rules in this stage are simple CFG rules. A clause can be generated as a result of combination of a clause and a constituent, or it can introduce a new embedded clause, or by reaching the end of sentence, a clause can be terminated.

These automata do not specify the precedence relations between the constituents and a separate Linear Precedence component imposes the precedence constraints. This is done incrementally and as a constituent is added to a clause, all the possible word order constraints are applied between it and the constituents which are already part of the clause. Note that we haven't considered any immediate dominance (ID) component and the binary precedence relations are not imposed on sisters of an ID rule.

[8]We introduced the notion of filtering in the algorithm (6.6).

6.4.1 Examples of Parsing in the second stage

The system parses a sentence by initialising a clause and attaches the incoming chunks to this clause.

For (6.4), repeated in (6.7), the first chunk is *Ali*. As a result of incremental attachment at this stage we will have:

(6.7) ali seab xord.
Ali apple ate
'Ali ate an apple.'

np([0,1],▷ obj:20 ◁,▷ subj:80 ◁)

Rule: Clause → NP Clause

Candidates:

▷ [0,1]:obj ◁ 44.72
▷ [0,1]:subj ◁ 89.44

We have kept the indexes for each constituent. For example [0,1] shows that this constituent starts at point 0 and ends at point 1 in the input string. We use these indexes in generating the output dependencies for the parser. The parser generates these after it reaches the end of the clause (not sentence) which it is parsing.

The candidates also show the competing paths in the path set for each clause. At the beginning when the clause is initiated this path set is empty and after parsing the first constituent, the candidates (or path set) will be initiated. It is at this stage that the activation values for each path will be calculated. We have assumed 100 as the initial number for an empty clause and when it is combined separately with 20 and 80, the results will be 44.72 and 89.44.

$\sqrt{20 \times 100} = 44.27$ and $\sqrt{80 \times 100} = 89.44$

100 is the maximum value of activation and the activation value can range from 0 to 100. We will give the justification for using square root function later in Section 6.4.2.

The second chunk is *seab* and as a result of multiplication, we will have four grammatical-function pairs as potential candidates in the path set: subj.obj, obj.subj, obj.obj, subj.subj.

np([1,2],▷ obj:80 ◁,▷ subj:20 ◁)

Rule: Clause → NP Clause

Candidates:

- ▷ [0,1]:obj ◁▷ [1,2]:subj ◁ 28.37
- ▷ [0,1]:subj ◁▷ [1,2]:obj ◁ 84.58

At this stage only two of the four possible alternatives can pass the filters. Since no sentence can have two objects or two subjects, the activation values of those sequences which have two subjects or two objects are reduced to zero and only two will survive. Note that $\sqrt{89.44 \times 80} = 84.58$ because of combining a subject with activation value of 89.4 with an object with activation value of 80. The other alternative is the result of combination of an object of activation value of 44.72 with a subject of activation value of 20. Note that because of violating the default word order of *subject precedes object* the result should also be reduced by the violation factor 0.90 (see Table 6.2 for precedence rules and values). Hence we will get $\sqrt{(42.7 \times .90) \times 20} = 28.37$.

verb-comp([2,3],$< Obj, Subj >$,100)

Rule: Clause → V-comp Clause

Candidates:

- ▷ [0,1]:obj ◁▷ [1,2]:subj ◁ 28.37
- ▷ [0,1]:subj ◁▷ [1,2]:obj ◁ 84.58

When the verb is added with the activation of 100, those subjects which don't agree with verb will be deleted. Since both of the subjects agree with the verb, both alternatives will survive[9]. Finally for the attachment of the arguments to the verb, the path with the highest activation (acceptability) will be chosen and the arguments are bound to the verb. Since no word-order constraint has been violated the activation value will be $91.97 = \sqrt{84.58 \times 100}$.

Note that with the same constituents and a different order, the constraints will interact to yield a different measure of acceptability. For (6.8) the acceptability measure is 89.58. This is because the example with canonical word order is considered more correct.

[9] To avoid confusion, we have not shown the agreement features in the examples.

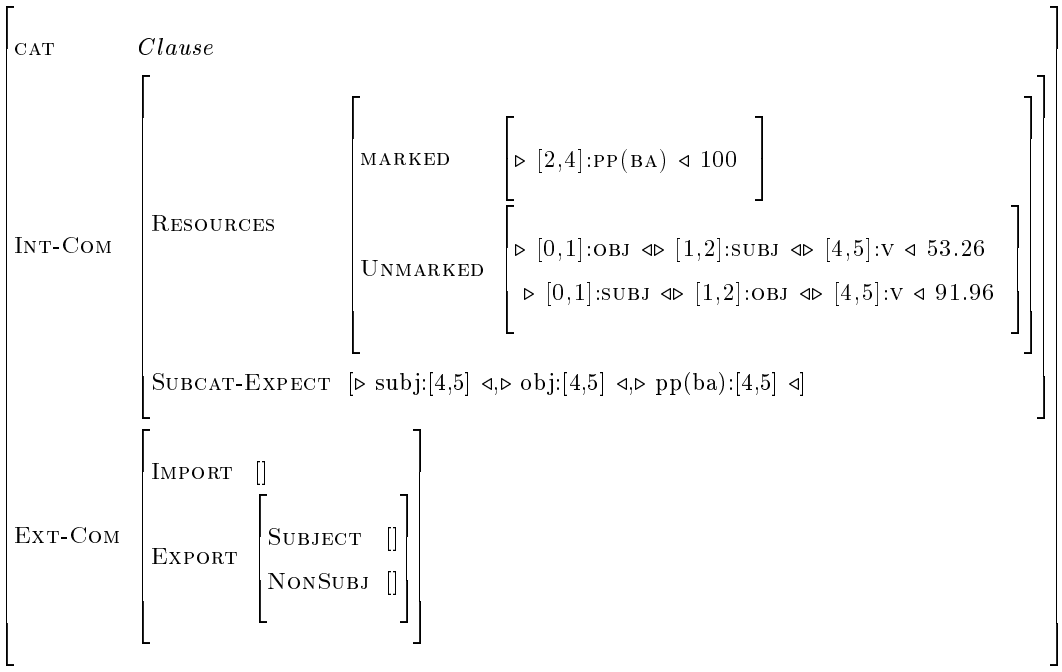
(6.8) seab ali xord.
apple Ali ate
'Ali ate an apple.'

In this example, the object precedes the subject and hence violates the canonical word order. As a result the activation will be multiplied by 0.90 (see Table 6.2 for precedence rules). A simple matrix for deriving the acceptability measure can be calculated by multiplying the values for the constraints which were violated. One can calculate all violations and yield a final violation measure and multiply the end result with this number. Instead we have multiplied each violation as soon as it is found. This incremental approach ensures that the alternatives which are reduced to zero are not further extended.

Finally in Persian, PPs can scramble freely and in parsing them we do not add them to the competition (unmarked) set, because they contribute the same to all the competing paths. Instead of adding them to all paths we factor them out and store them in another marked structure because their contribution to all parallel paths is similar. This is illustrated in (6.10).

(6.9) ali seab ba changal xord.
Ali apple with fork ate-3S
'Ali ate an apple with fork.'

(6.10)



(6.10) shows the different structures for the sentence before the generation of the dependencies. INT-COM (INTernal COMmunication) is considered for the local dependencies, while EXT-COM (EXTernal COMmunication) is considered for long distance dependencies.

In (6.10), the path set which captures the non-PP competitions corresponds to INT-COM/RESOURCES/UNMARKED and INT-COM/RESOURCES/MARKED structure (i.e. the mark set) holds the PPs. The subcategorisation expectations of the verb are also added to a separate structure in our model (i.e. SUBCAT-EXPECT) and when the end of the clause is reached the resources and the expectations are matched with each other and the dependency links are generated. In our model we choose the path with the highest activation (i.e. Best-path in (6.11)) and discard the other ones.

The EXT-COM has two substructures, the IMPORT and EXPORT for passing LDS constituents. The subject one is used for capturing control.

Note also the difference in the order of items for subcat resources and the normal resources (associated with NPs and PPs). In subcat we have subj:[4,5] where [4,5] corresponds to the location of the verb in the sentence, while in the unmarked resources we have [0,1]:subj (note the difference in the order of grammatical relation and the brackets in the two.). When the parser reaches the end of a clause, the highest active path in unmarked will be selected and the matching subj resource and subcat subj resource[10] are joined.

As a result of joining these two a dependency link with value [0,1]:subj:[4,5] will be generated. This depicts a transaction or communication across a subject communication link; in this transaction [0,1] is the producer and [4,5] the receiver. Similarly two other transactions for obj and pp(ba) links will be generated by this distributed approach to communication resources. The algorithm for generating dependencies is shown in (6.11).

In this algorithm, the resources associated with NPs (i.e. in path-set) will be first matched with the subcat resources, then the resources associated with the PPs which are stored in mark-set and finally the resources kept in import-set are matched with subcat resources. This priority in claiming subcat resources also ensures that control has priority over long distance scrambling.

[10] In our model we consider grammatical relations as pairs of links that attach NPs and Verbs. As we explain and illustrate in the next chapter, these links can be considered as communication resources between two linguistic processes.

(6.11) **Generate Dependencies: (version 1)**

Commit(Clause, Export)

a. Get subcat-resources of the clause.

b. Find the Best-path in the Path-set of the Clause.

c. Do while Best-path list not empty

If head of Best-path matches a subcat-resource

i. Generate the dependency

ii. Delete the resource from subcat-resource

iii. If the resource is marked as +control[11] then copy it as subject in export.

Else add the head to the export.

Remove head of Best-path.

d. Get mark-set of the clause (containing the PPs in the clause).

e. Do while mark-set list not empty

If head of mark-set matches a subcat-resource

i. Generate the dependency.

ii. Delete the resource from subcat-resource.

iii. If the resource is control then copy it as subject in export.

Else add the head to the export.

Remove head of mark-set.

f. Do while import-set not empty

If head of import-set matches a subcat-resource

i. generate the dependency.

ii. Delete the resource from subcat-resource.

iii. If the resource is control then copy it as subject in export.

Else add the head to the export.

Remove head of import-set.

[11]The control cases are marked in the lexicon for each subcat resource (on verbs) by an extra +control feature.

Description:

The program generates the necessary dependencies when a complementiser or the end of sentence marks the end of a clause. commit(Clause,Export) generates the dependencies and produces the exported constituents to be passed to the next clause (or checked to be empty for the end of sentence). Control is imposed by initialising the embedded clause with an exported subject.

If some of the resources in marked or unmarked parts could not be unified by a corresponding element in the subcategorisation frame of the verb, then these resources are moved into the embedded clauses. This is because of long distance scrambling in Persian in which some resources might belong to other embedded clauses. It is also possible that some of the expectations of the verb might not be satisfied due to the nature of pro-drop in Persian for the arguments of a verb[12]. The feature EXT(ernal)-COM(munication) is introduced to hold these cases. As we will show later in (6.17) EXT-COM has two features "import" and "export".

To sum up, in parsing local scrambling, as the unmarked arguments (subject, object) are added incrementally to the clause, the parser creates a parallel set of the plausible paths. The paths are restricted by some constraints. The constraints on local scrambling can be divided into hard and soft constraints. The hard constraints are strict precedence relations and verb-subject agreement which could block a path by reducing its activation value to zero. The other constraints which only reduce or increase the activation values to a non-zero value are soft constraints. The accumulative result of these values contribute to the possibility (activation) of a solution. The most active solution or path (i.e. among unmarked paths) will be chosen. Depending on the function which we use we will have different results. The constraints can be summarised as:

- Word order restrictions. These were illustrated in 6.2 and are used to penalize possible alternatives which deviate from the canonical word order. They also block alternatives which violate obligatory word order rules.

- Verb subject agreement. In Persian a subject must agree with the verb of the clause.

[12]The number of fulfilled expectations can contribute to the activation positively, but we have not considered it in our implementation.

There is no instance of *split ergativity* in Persian, and tense-dependent agreement has not been considered.

- One example of each resource in the sentence or Resource Limitation Principle (RLP).

(6.12) **Resource Limitation Principle**

No two NPs can exist in a clause with the same grammatical functions.

In the rest of this section we will elaborate on RLP[13] . Consider example (6.13).

(6.13) ali *seab* be man qol=dad [ke __ be ali bedahad].
Ali *apple* to me promise=gave-3S [that __ to Ali gave-3S]
'Ali promised me to give the apple to Ali.'

In Persian it is not possible for two grammatical resources with the same grammatical functions to appear in the same clause. Hence (6.13) is ungrammatical.

(6.14) * amir *seab* be man *be ali* qol=dad [ke __ __ bedahad].
Amir *apple* to me *to Ali* promise=gave-3S [that __ __ gave-3S]
'Amir promised me to give the apple to Ali.'

This is despite the fact that from a competence point of view, as we discussed in Section 3.6, this sentence should be grammatical. But in Persian it is not possible to have two NPs in a clause with the same grammatical function. In other free word order languages such as German, such an example and the existence of two dative NPs does not create a problem.

This performance constraint that we call Resource Limitation Principle (RLP) is not restricted in Persian to datives and no clause can exist in which two phrases (resources) have the same grammatical function. RLP has been implemented in our system as a general constraint that a resource cannot precede another with the same grammatical function (as is implemented in our system). We have used an extension to the blocking word order restrictions. For example the constraint that 'no subject can precede another subject' implements the existence of at most one 'subject' marked[14] resource in a clause. Note that resources are only exported (not copied) if they don't have a matching subcat resource.

[13] See another performance constraint RBP in (6.21).
[14] Future work is needed to specify this constraint based on case marking and not grammatical functions.

One can use RLP to differentiate and classify the flexibility of the word order and scrambling in free constituent order languages.

6.4.2 The Choice of the Function

The previous numeric constraints should be added together by a function to to yield a number representing the grammaticality/acceptability of an alternative. The choice of the function for combining two activation values is important. We have considered three functions for this purpose:

1. Arithmetic mean: $f_1(a,b) = (a+b)/2$
2. Multiplication: $f_2(a,b) = a \times b$
3. Geometric mean $f_3(a,b) = \sqrt{a \times b}$

These equations can be contrasted with each other by using the following *skeleton axioms* used also for fuzzy intersection in Fuzzy logic [Klir and Folger, 1988]. In Fuzzy Logic, a formal apparatus for partial membership in a set has been introduced, this is in contrast to the standard notion of *crisp* set where an object can either belong to a set or not. The activation values for each path in the path set that we introduced earlier, uses a similar notion of partial membership as in fuzzy logic[15].

Axiom 1. $f(1,1) = 1$; $f(0,1) = f(1,0) = f(0,0) = 0$, f behaves as the classical intersection with crisp sets (*boundary conditions*).

Axiom 2. $f(a,b) = f(b,a)$; that is, f is *commutative*.

Axiom 3. If $a \leq a'$ and $b \leq b'$, then $f(a,b) \leq f(a',b')$; that is, f is *monotonic*.

Axiom 4. $f(f(a,b),c) = f(a,f(b,c))$, that is, f is *associative*.

[15]The word order constraints that we introduced also use fuzzy relations and terms. Note that we use percentage in our notation. So 100 in our notation is equivalent to 1 here.

	$f_1 = (a+b)/2$	$f_2 = a \times b$	$f_3 = \sqrt{a \times b}$
Boundary	NO	YES	YES
Commutative	YES	YES	YES
Monotonic	YES	YES	YES
Associative	NO	YES	NO
Continuous	YES	YES	YES
Idempotent	YES	NO	YES

Table 6.3: Comparison of Functions.

Axiom 5. f is a *continuous function*. This axiom prevents a situation in which a very small increase in either a or b produces a large change in $f(a, b)$.

Axiom 6. $f(a, a) = a$; that is, f is *idempotent*.

All the three functions listed above are continuous, monotonic and communicative (axioms 2, 3, 5). f_1 satisfies axiom 6, but it does not satisfy the remaining axioms 1 and 4 and is not useful for representing blocking of constraints.

f_2 satisfies all the axioms except Axiom 6. In our model we have chosen f_3 which is not associative, but satisfies the first axiom and the last axiom. It is because the geometric mean of two numbers is a number between the two and even in cases when one of the numbers is 1, the geometric mean gives a better value compared to multiplication which returns the other number. The axioms in fuzzy sets are a good starting point for exploring the axioms that a linguistic fuzzy function should respect. In the fuzzy literature, there are a set of functions [Yager, 1980] which satisfy all the constraints and in the future research it is worthwhile to investigate their effect on activation values.

In the next section we will examine examples of long distance scrambling that we discussed in 3.

6.5 Parsing Long Distance Scrambling

In our model we have considered examples of long distance scrambling for prepositional phrases in Persian. The grammatical resources in a clause might not be expected by the verb and these resources can be exported into embedded clauses. This creates examples of long distance scrambling. In (6.15) *ba changal* 'with fork' should be attached to the embedded

clause.

(6.15) ali *ba changal* goft [ke seab __ bekhor-am].
Ali *with fork* said-3S that apple __ SUB-ate-1S
'Ali told (me) to eat the apple with fork.'

The following algorithm gives more details of parsing embedded clauses.

(6.16) **Main Body: (version 1)**
attach-new-clause(Clause)

a. Derive the dependencies in Clause by Commit(Clause,Export).
b. Initialise The Import-set of a New-Clause with the Export-set of Clause.
c. Input a Chunk from the pipe.
d. Do while Chunk not end of sentence.
   i. If Chunk is a complementiser: attach-new-clause(New-Clause).
   ii. else If Chunk is a phrase: attach(Chunk,New-Clause).
   iii. Input a new Chunk from the pipe.
d. Call Commit(Clause,Export)[16].

Description:
This will generate an embedded clause. The Import-Set and Export-Set correspond to the gap-nogap pair in the gap threading technique. But the word order constraints and control constraints can update the value of the Export-set. The value of Export-set will be determined by Commit which determines the final dependencies for the clause by choosing the optimal alternative and combines the resources of the clause (i.e. subcategorisation frame of the verb) with the possible grammatical functions. After that it can finalize the value for Export-Set. When the parser terminates, the parser makes sure that the Export-Set is empty.

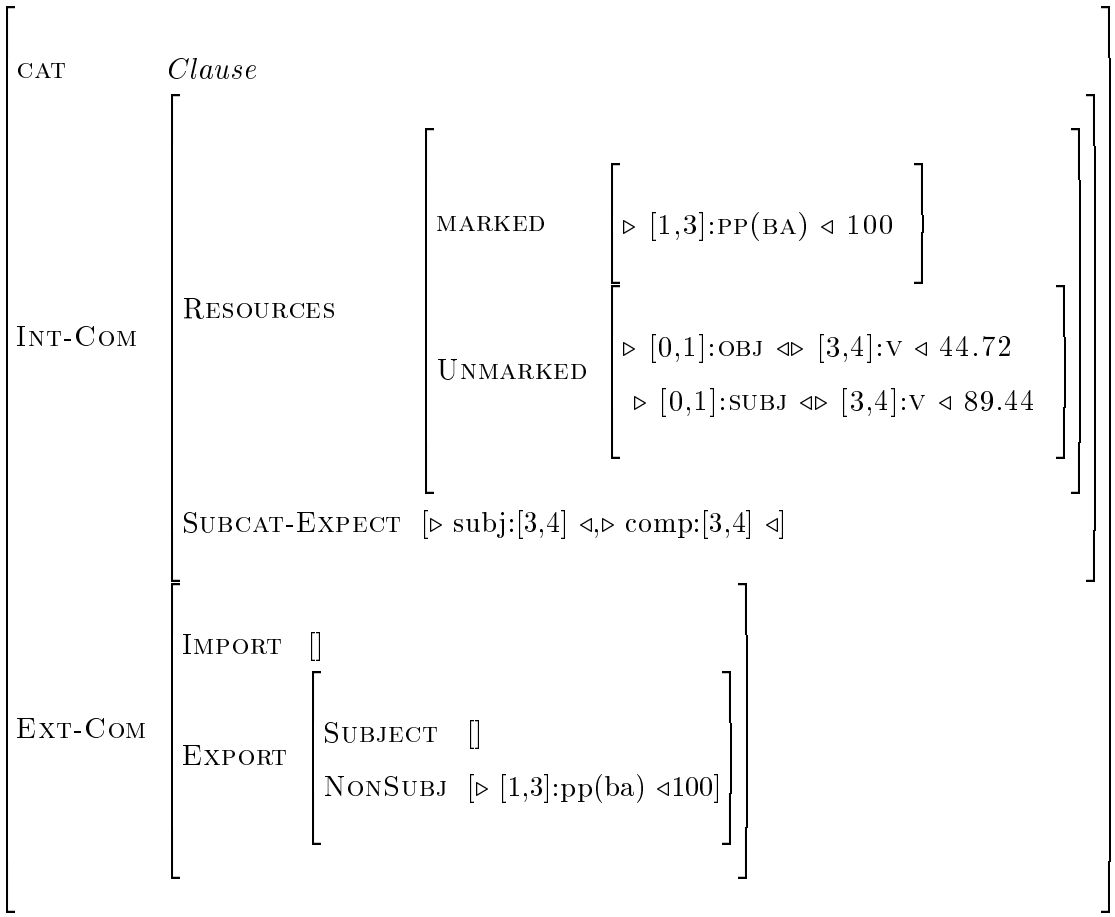
6.5.1 Long Distance Scrambling as Resource Passing

To accommodate long distance scrambling, we have added to each clause an *export* structure which contains the resources which are not matched by the expectations of the verb. (6.17)

[16]Commit is called to generate dependencies and is described in (6.11). At the end of sentence Export must be empty.

shows the corresponding graph for (6.15). In this example pp(ba) is not matched with the verb subcat resources and is added to the *export* structure.

(6.17)



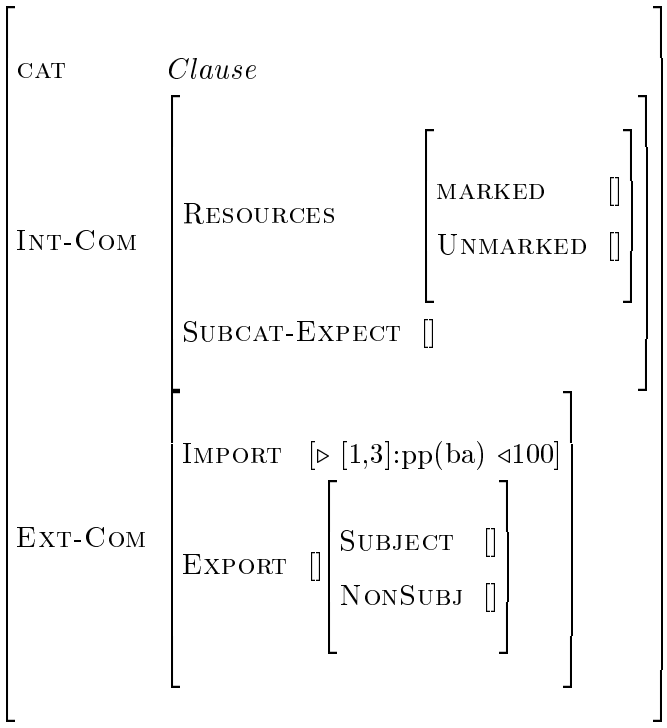
Upon creation of a new embedded clause these resources will be passed into the new clause and will be placed in an *import* structure in the new clause. The import structure for the main clause is initialised to null and for the most embedded clause the export structure must be empty for the sentence to be grammatical. This is similar to gap-threading, with this difference that the export sets can be extended with activation values in order to model competition and underspecification. But modeling such competition requires a corpus and we do not have such a corpus. In this section we will concentrate on long-distance scrambling (LDS) for non-subjects[17]. LDS for prepositional phrases are more common than the non-PP examples of LDS. These examples of LDS also have less interaction with discourse phenomena of Persian that we have not implemented. Examples of *rā* marked LDS interact with the

[17] In other work we have studied relative clauses and the long distance scrambling of objects [Rezaei, 1993] that we do not consider here.

notion of secondary topicalisation in addition to specific object marking that we discussed in chapters 2 and 3.

In our model after choosing the path with the highest activation and joining the resources with corresponding items in subcat resources (in Subcat-Expect), the unmatched resource(s) which have no matching corresponding resource in the subcat structure will be added to the export structure and then passed into the embedded clause as import structure. The creation of embedded clause for parsing (6.15) and initialisation of it is illustrated in (6.18).

(6.18)



As we have illustrated, after creating the embedded clause the export value of the main clause will be assigned to the import structure of the embedded clause. In the process of attachment of phrases to the embedded clause, the subcat resources will be matched with the normal resources of the embedded clause and after there is no other possibility for joining, the imported elements will be tried[18].

If the imported resource is not matched or there are extra unmatched resources inside the new clause, again all the unmatched resources will be exported into the next embedded clause and this continues until the end of sentence is reached. For (6.15), the pp(ba), which is imported into the embedded clause, will join with the corresponding subcat-resource of the

[18]See the algorithm for generate dependency in (6.11)

embedded verb and a dependency will be created.

If an instance of a resource is already present in a clause, then that resource (e.g. pp(ba)) will create a barrier in front of the progress of an imported resource competing for the same subcat resource. The conflict between the two will block the parse. In a robust parsing environment this may contribute to reducing the activation of the clause and not blocking the parse. As a result of this principle the following example is ungrammatical.

(6.19) * ali *ba changal* goft [ke seab ba kard bekhoram].
Ali *with fork* said-3S that apple with knife ate-1S
'Ali told (me) to eat the apple with fork, with knife.'

Here *ba changal* will be imported into the new clause, but when *ba kard* is attached to the unmarked set of the embedded clause a violation will happen. This is implemented by checking that new additions to the marked set are not already present in the import set[19]. The previous algorithm 6.6 for attaching chunks should be extended for this purpose. The complete version for attachment of PPs is shown in (6.20).

(6.20) **Main Body: Attach (Final Vesion)**
attach(Chunk,Clause)

a. If Chunk is NP:
   i. Add grammatical functions of the Chunk to the Path-set of Clause.
   ii. Use Apply-filter to impose word order constraints on the new Path-set.
   iii. Update Export-set (and Import-set) for long distance scrambling.

b. If Chunk is PP:
   i. If the gram. function of the Chunk is already present in the mark-set block.
   ii. Else If the gram. function of the Chunk is already present in the Import-set then block.
   iii. Otherwise Add the gram. function of the Chunk to the Clause.
   iv. Update Export-set (and Import-set) for long distance scrambling.

c. If Chunk is Verb:

[19] This is how we have implemented the *Resource Barrier Principle* (RBP) constraint.

i. Add the subcategorisation frame of the Chunk as subcat-resources of the Clause.

Description:

The program adds new chunks and makes sure (by Apply-filter) that the ungrammatical paths are removed from the Path-set. It works in two stages of GENerating all possible paths from combining each possible grammatical function of the chunk with the Path-set, and then shrinking the Path-set to a smaller one which respects the word-order constraints and performance principles (such as RLP and RBP). If the chunk is a verb, then the subcategorisation frame of the verb is added to the clause as the expected resources of the clause. The Export-set and Import-set are used for long distance scrambling (LDS) and will be explained later in the LDS section.

In this algorithm (b.i) implements RLP for attaching PPs and (b.ii) implements RBP for attaching PPs.

Our approach for representing long distance scrambling differs from GPSG, LFG and GB. For representing unbounded dependencies, some versions of GPSG allow empty categories [Gazdar et al., 1985]. In a highly pro-drop language such Persian (and Japanese) which allows different constituents of the clause to be empty, this will cause problems and many instances of empty categories will be generated. HPSG and some versions of GPSG do not use empty categories. More recently, in psycholinguistic research, the existence of empty categories for unbounded dependencies has been questioned.

LFG uses the mechanism of functional uncertainty for representing long distance scrambling (LDS). The use of functional uncertainty provides a powerful mechanism to deal with long distance scrambling. Functional uncertainty also allows the constituents to have multiple grammatical functions associated to them and the different principles of LFG make sure that each constituent will have only a unique grammatical function. In contrast, in our approach we do not use functional uncertainty and the grammatical functions are made explicit and compete against each other. The addition of competition to LDS in our model is straightforward and one can use an extension to path set for this purpose. But in LFG it is not clear how one can add competition to the underspecified grammatical functions. It has been suggested that for addition of such competition to LFG, one needs to make all grammatical relations

explicit [Bresnan, 1996]. This is in conflict with the underspecification notion of functional uncertainty.

Finally GB uses transformations and traces for this purpose. In our model we don't use gaps, traces or functional uncertainty and by introducing the notion of *external communication*, instances of grammatical resources are imported from/exported into clauses.

In the GB literature there are constraints such as *Subjacency* or *Complex NP Constraint*(CNPC) on movement of elements out of certain clauses and phrases. We discussed examples of these constraints for Persian in Chapter 3. In our implemented model there are also additional constraints that restrict the possible instances of external communication between clauses. For example the existence of a grammatical resource inside a clause acts as a barrier in front of importing that resource from higher clauses into the clause or into clauses dominated by the present clause. We call this constraint the *Resource Barrier Principle*[20] (RBP) which constrains examples of long distance scrambling. This constraint blocks the progress of examples of LDS which are not grammatical in Persian.

(6.21) **Resource Barrier Principle**

If a resource exists in a clause, it acts as a barrier in front of the resources with the same grammatical functions which want to scramble into lower level embedded clauses from higher ones.

Consider the sentence in (6.22) which has two different meanings depending on whether *be ali* is attached to the main verb or to the embedded verb.

(6.22) man [be ali] qol=dadam [ke seab bedaham].
I [to Ali] promise=gave-3S that apple gave-1S
'I promised to Ali to give apples to someone.'
'I promised to give apples to Ali.'

When another clause with a dative is added in the middle of the two clauses in(6.22), as illustrated in (6.23) then one of the interpretations are automatically blocked.

(6.23) man be ali qol=dadam [ke be hasan begam [ke seab bedaham].
I to Ali promise=gave-1S [that to Hasan tell-1S that apple gave-1S

[20]See another performance constraint RLP in (6.12).

'I promised to Ali to tell Hasan to give apples to someone.'

'* I promised to tell Hasan to give apples to Ali.'

This is because in Persian a Resource Barrier Principle (RBP) exists which blocks the second interpretation[21]. RBP is not specific to dative resources (phrases) in Persian and it applies to all resources. This ensures that the examples of *cross dependency* for the same resources in Persian are ungrammatical. RBP as a performance constraint restricts long distance scrambling in Persian and it allows only *serial dependencies* for the resources with the same grammatical functions.

Like RLP one might use RBP to differentiate and classify the flexibility of the word order and scrambling in free constituent order languages. It should also be investigated to see whether there is any language that violates RBP. If such a language does not exist, then this performance constraint can be regarded as a property of the architecture of the human sentence processor.

6.5.2 Control as Resource Copying

The notion of *Control* creates possibilities for export of resources into embedded clauses and brings forward challenging problems.

Earlier we defined soft and hard constraints which restrict the possible domain of local scrambling. The word order constraints imposed soft constraints by reducing the activity level of an alternative, while the resource limitation equations could block the progress of one of the alternative paths and act as hard constraints for local scrambling.

For long distance scrambling we also have a set of soft and hard constraints. In the previous section we saw that the presence of a PP in a clause blocks the scrambling of other PPs from higher clauses into it or into lower clauses dominated by this clause. This acts as a hard constraint on long distance scrambling. Another hard constraint which affects the result of competition between two alternatives is the notion of *control* in the grammar. In our implementation we have considered the effect of PP control on long distance scrambling.

(6.24) ali be mohammad goft [ke seab bexorad].
Ali to Mohammad told-3S that apple eat-3S

[21]Note that still another ambiguity can occur for the attachment of *to hasan* to the middle clause or the lowest clause.

'Ali told Mohammad to eat the apple.'

In this example when the verb *goft* governs the PP *be mohammad*, the PP will control the subject of the embedded clause. The control cases are marked in the lexicon for each subcat resource (on verbs) by an extra feature. The system will test this feature when it generates a dependency link for such resource. The feature that we have considered is *+control* and *-control*. If the feature is *+control*, then the matching resource/chunk is copied into the export/subject of the clause. When the embedded clause is created, the marked-set of the new clause will be initialised by the export/subject element and with *subj* role. The final version of algorithm (6.16) is shown in (6.25) with expanded b part.

(6.25) **Main Body: (final version)**

attach-new-clause(Clause)

a. Derive the dependencies in Clause by Commit(Clause,Export).

b.i. Initialise the Import-set of New-Clause with the Export-set/nonsubj of Clause.

b.ii. Initialise the path-set of New-Clause with the Export-set/subject of Clause.

c. Input a Chunk from the pipe.

d. Do while Chunk not end of sentence.

i. If Chunk is a complementiser: attach-new-clause(New-Clause).

ii. else If Chunk is a phrase: attach(Chunk,New-Clause).

iii. Input a new Chunk from the pipe.

d. Call Commit(Clause,Export)[22].

Description:

This will generate an embedded clause. The Import-Set and Export-Set correspond to the gap-nogap pair in the gap threading technique. But the word order constraints and control constraints can update the value of the Export-set. The value of Export-set will be determined by Commit which determines the final dependencies for the clause by choosing the optimal alternative and combines the resources of the clause

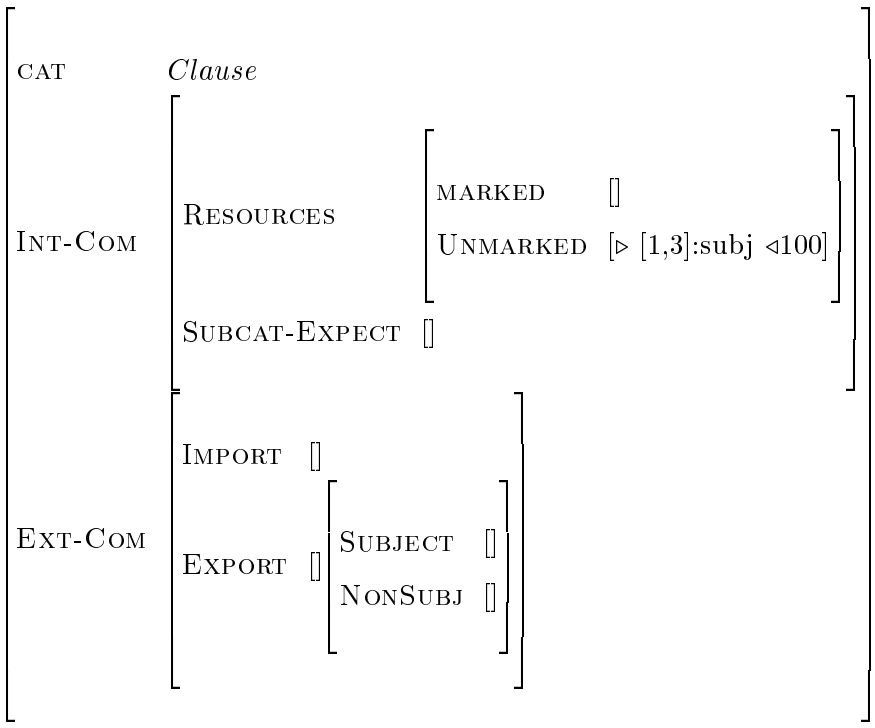
[22]Commit is called to generate dependencies and is described in (6.11). At the end of sentence Export must be empty.

(i.e. subcategorisation frame of the verb) with the possible grammatical functions. After that it can finalize the value for Export-Set. When the parser terminates, the parser makes sure that the Export-Set is empty.

For the previous example (6.24) the structure that the parser creates is illustrated in (6.26).

The details of the dependency generation will be the same as in the previous examples that we studied.

(6.26)



Upon initialisation of the embedded clause this subject resource will be copied into the path set of the embedded clause. Control is achieved by this mechanism. The interaction of control and long distance scrambling creates interesting cases in Persian.

6.5.3 Resource Competition in LDS

Consider the examples in (6.27) and (6.28):

(6.27) ali be mohammad goft [ke seab bexorad].
Ali to Mohammad told-3S that apple eat-3S
'Ali told Mohammad to eat the apple.'

(6.28) ali *be mohammad* goft [ke seab __ bedaham].
Ali *to Mohammad* told-3S [that apple __ give-1S]
'To Mohammad, Ali told (me) to give the apple.'

In (6.27) Mohammad controls the subject of the embedded clause and agrees with it. In contrast in (6.28), Mohammad cannot control the subject of the embedded clause, because it doesn't agree with the embedded verb. Nevertheless the sentence is grammatical and *be Mohammad* is exported by long distance scrambling into the embedded clause. Since the resource is not attached to the main verb, it also does not act as a controller. It is an instance of garden path in Persian where the control and long distance scrambling interact.

Our solution for representing these cases is to consider the PP resources which act as controller as possible long distance scrambling cases[23] and for these cases if the resource cannot act as the subject of the embedded verb, then we allow the scrambling case to occur. The algorithm for generate dependency is extended so that for PP subcat-resources which are +control (1) a dependency is generated (2) a possible LDS case is added to the export and (3) the export/subject is also initialised with an alternative[24]. We also make sure that when the subject control is satisfied in the embedded clause, the LDS one becomes inaccessible[25] The general principle is that one cannot use an entity or parts of it twice in the same clause. Algorithm (6.29) is the extended version of Algorithm (6.11).

(6.29) **Generate Dependencies: (final version)**
Commit(Clause, Export)
a. Get subcat-resources of the clause.
b. Find the Best-path in the Path-set of the Clause.
c. Do while Best-path list not empty
If head of Best-path matches a subcat-resource
i. Generate the dependency
ii. Delete the resource from subcat-resource

[23]Recall that we have limited our study to the implementation of LDS for PPs.
[24]Note that the subject control one has the activation value of 100 (compared to 80) which makes it the first alternative among the two to win (of subject control and no subject control).
[25]This is achieved by considering an additional flag shared between the LDS resource and the competing copied subject. When the copied subject can unify as the subject of the embedded clause then this flag is set.

iii. If the resource is marked as +control then copy it as subject in export.

Else add the head to the export.

Remove head of Best-path.

d. Get mark-set of the clause (containing the PPs in the clause).

e. Do while mark-set list not empty

If head of mark-set matches a subcat-resource

i. Generate the dependency.

ii. Delete the resource from subcat-resource.

iii. If the resource is control then

iii-1. copy it as an alternative path with a copied subject and no copied subject into export/subject.

iii-2. add it also to the export/nonsubj.

iv. Else skip.

Else add the head to the export/nonsubj.

Remove head of mark-set.

f. Do while import-set not empty

If head of import-set matches a subcat-resource

i. generate the dependency.

ii. Delete the resource from subcat-resource.

iii. If the resource is control then copy it as subject in export.

Else add the head to the export/nonsubj.

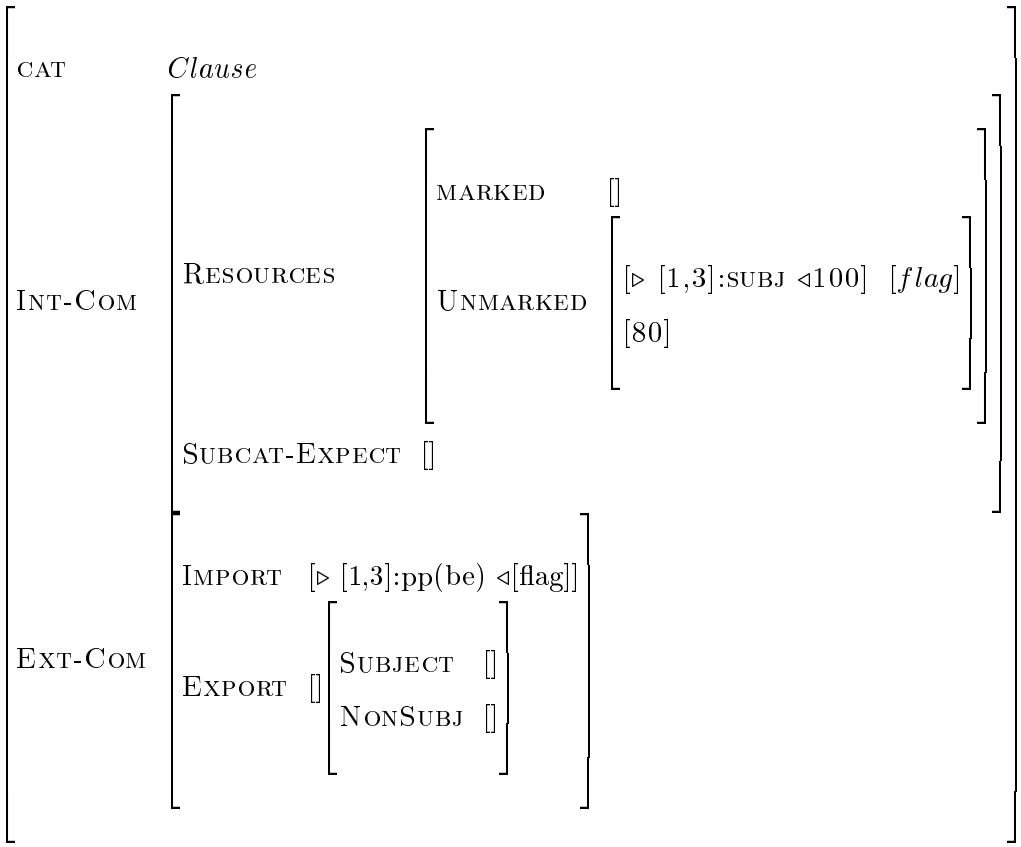
Remove head of import-set.

Description:

The program generates the necessary dependencies when a complementiser or the end of sentence marks the end of a clause. commit(Clause,Export) generates the dependencies and produces the exported constituents to be passed to the next clause (or checked to be empty for the end of sentence). Control is imposed by initialising the embedded clause with an exported subject.

(6.30) illustrates the structure of the embedded clause generated by the parser when it is initialised for (6.27) and (6.28). Only when the verb of the embedded clause is joined, the analysis of the two sentences depart from each other. For the former, the control version is chosen and LDS becomes inaccessible, while for the latter, the control fails and as a result, the LDS version is selected.

(6.30)



In our solution we have considered that control has priority over long distance scrambling. But is this the only case of competition? Consider the following examples:

(6.31) ali be mohammad goft [ke seab bedahad].
Ali to Mohammad told-3S that apple give-3S
'Ali told to give apples to Mohammad'
'Ali told Mohammad to give apples (to someone)'

(6.32) ali *be madrese* goft [ke __ beravad].
Ali it to school said-3S [that __ go-3S]

'Ali told (him) to go to school.'

In (6.31) depending on the context of the sentence, two alternatives are possible in which *be Mohammad* either attaches to the main verb or to the embedded verb. In (6.32) the semantic category of the noun rules out the interpretation in which *be madrese* is attached to the main verb. We have experimented with our model and have allowed this type of competition between the two verbs for one resource to occur. Our restricted solution was to allow LDS to occur for all PPs even if they can join to the main verb. In our extended implementation each exported resource has an extra value attached to it which by default is zero. This value shows the previous offer in the higher clause for this clause and each time a higher bid for attachment is made in a lower embedded clause the value will be updated. The RBP limits the possible alternatives.

Our solution is to keep track of the most plausible attachment and when an attachment is possible which is more plausible than the previous ones, then retract the previous attachment and commit to the new one. But resolving the competition conflicts ultimately needs semantic information also.

For considering these examples we need to add an extra semantic concept to the model, in order to allow verb control-type agreement and scrambling to compete with each other. In future work one would need to derive semantic restrictions of a verb and its arguments from a corpus of Persian to let no scrambling cases compete with scrambling cases. Even adding semantic information is not enough for disambiguating some competition cases and further information about world knowledge and context of the sentence is required.

Note that this type of competition also needs an extra level of non-monotonicity added to the system. This is required to keep track of the highest bid for the resource and to make the previous bid invalid. We have not implemented this and it needs further research. The non-monotonic extension can be considered as the evolution of the use of flag in the competition between control and LDS. Such a parser that deals with all instances of competition needs an additional level of representation that assigns names (or numbers) to the potential paths[26].

[26]The paths in conflict should be marked by assigning them to a set and the working memory of the parser must make sure that once a path fails, another potential one will be selected.

6.6 Discussion

6.6.1 Parallel Structures and Competition

In Chapter 4 we discussed a verb-driven ID/LP parser/grammar system for parsing the grammar of Persian [Rezaei, 1993]. The underlying formal automaton of this approach is an instance of a set based embedded automaton. But the implemented parser spends a lot of time in backtracking. Since the potential for backtracking is clearly great for parsing Persian, non-backtracking competitive models are helpful.

Another approach for parsing Persian under a competitive framework is [Rezaei and Crocker, 1995]. The parsing architecture suggests a parallel distributed model of parsing, where all possible interpretations are expanded and run in parallel together.

The major problem with this approach is adding the necessary mechanism for competition among the different interpretations. This requires complex notions of synchronisation such as blackboards in a shared environment or barriers in Message Passing Interface (MPI) [Gropp et al., 1994].

Our solution to this problem for running different interpretations in parallel is to pack all possible alternatives in a static data set. This was inspired by the use of the notions of fuzzy sets. Some of the features that we used in our model could have a range of possible values and a number associated with each value. For example the grammatical functions that we used in some cases could be either object or subject. This is an example of a feature which has a fuzzy set as value. Kim [1994] proposes the use of such fuzzy sets and introduces graded Unification. In our approach we have extended this notion of graded unification to yield a notion of graded grammaticality. We also introduced fuzzy word order rules that restrict the range of possibilities in a path set.

The path set implements a mechanism for modeling competition among a set of paths. Corresponding to each path we had an activation value. By assigning a value to each path, we implemented a numerical notion for competition. Such competition notion is robust enough to capture soft and hard constraints for word order rules.

Each word order rule had numeric value attached to it. This gives the grammar writer the flexibility to model degrees of variation from canonical word order. Again we borrowed terms from fuzzy set literature for modeling *always*, *most of the time* numerically. Another

alternative was to use constraint ranking in optimality theory. But we will argue later that the numeric approach is more powerful than the optimality approach.

In modeling competition numerically, one can easily represent blocking as reduction of the activation value to zero. Another advantage is that such mechanism can be extended to allow robustness and degrees of ungrammaticality.

These were some of the advantages of using features with fuzzy set values in modeling competition. One can use feature sets instead of feature values for all instances of underspecification. Another approach is to use functional uncertainty as in LFG. If one wants to add probabilities to functional uncertainty, then one needs to use a notion as fuzzy set that we have used. This introduces a notion of *functional competition* instead of functional uncertainty.

We have concentrated mainly on underspecification in grammatical functions in a verb final language with flexible word order. In many of the head driven approaches to parsing, the arguments of a verb are not attached until the verb appears in the sentence. These approaches are very useful in languages such as English where most of the arguments of the verb appear after it, but for verb final languages there is a level of competition among the arguments before the verb appears. In our approach we considered a path set for each clause. As the arguments are attached to a clause in a somehow incremental fashion, all possible interpretations of the argument with the degree of their certainty is added to the path frame of the clause. Later when the verb appears, the verb's subcategorisation frame restricts the possible interpretations.

We have specified the constraints that a marked NP puts on another marked NP and have derived some of the necessary constraints that one argument imposes on another. These defuzzification constraints are examples of surface word order constraints and they can include other grammatical constraints such as control.

Unlike previous approaches, that specify constraints on syntactic structures, we specify them on the surface word order itself. In the previous approaches the role of surface word order has been captured by using word order rules [Pereira and Warren, 1983] or principles [Crocker and Lewin, 1992]. In the former the rules generate the possible structures and the latter the principles are imposed on a set of phrase structure rules (e.g. X-Bar) to restrict the possible syntactic structures. For head driven approaches, the constraints are mainly

specified as constraints between heads and their arguments. Again in absence of a completed head, these principles cannot be applied and the process has to be delayed until the head is projected. In contrast to these traditional approaches, in our approach, the arguments contribute to the disambiguation process of grammatical functions as soon as they appear in the surface word order.

Our approach is useful for capturing grammatical function competitions between the arguments of the verb in SOV languages. There are further performance restrictions that are imposed on these path sets. For example no verb can have more than four arguments, or in Persian it is not possible for a clause to dominate two sisters with the same grammatical function.

In our approach, we used two distinct levels of representations: one with fixed word order(constituent level) and another with flexible word order (clause level). Phrase structure rules are used for the representation of fixed word order component and process structures with fuzzy path sets are employed for representing flexible word order at the constituent level.

Finally PPs in Persian (i.e. phrases marked with preposition) can scramble freely and in parsing them we do not add them to the competition set, because they contribute the same to all competing paths and instead of adding them to all paths we factor them out and store them in another structure. For other languages this might not be the case.

6.6.2 Resource Limitations

We introduced a notion of limited resources by the Resource Limitation Principle (RLP) for representing local scrambling and long distance scrambling in Persian. By using blocking word order rules we blocked the progress of paths which contained two instances of a grammatical resource. Not that these constraints are in addition to the structural constraints and barriers that we studied in Chapter 3.

We also discovered another blocking constraint, the Resource Barrier Principle (RBP) for long distance scrambling and export of resources. The notion of limited resources for long distance scrambling is a grammatical constraint that hasn't been investigated for Persian.

The use of import and export of resources easily captures this constraint, while in theories and frameworks such as LFG it is not clear how this constraint can be enforced. This is another notion of barrier which might be true in languages with a limited notion of long

distance scrambling.

Hoffman [1995] gives interesting examples of a similar phenomenon in Turkish. The examples in (6.33)-(6.36) repeated here from (43)-(45) in [Hoffman, 1995] demonstrate this.

(6.33) Fatama [Ali ev-e git-ti] san-di.
Fatama [Ali house-Dat go-Past] think-Past
'Fatama thought Ali went home.'

(6.34) Eve$_i$ Fatama [Ali e$_i$ git-ti] san-di.
House-Dat Fatama [Ali e$_i$ go-Past] think-Past.
'To the house, Fatama thought Ali went there.'

(6.33) shows a typical example of center-embedding in Turkish. In (6.34) a dative marked element *Eve* is scrambled into the main clause and the sentence is grammatical. The scrambling of Ali into the main clause as demonstrated in (6.35) is not possible because in our framework there is already an instance of subject resource (i.e. Fatama) in the main clause and two instances of the same resource cannot be present in the same clause.

(6.35) * Ali$_i$ Fatama [e$_i$ ev-e git-ti] san-di.
* Ali Fatama [– house-Dat go-Past] think-Past.
* 'As for Ali, Fatama thought he went home.'

Recall that in Chapter 5 we criticised the application of clause union in CCG for capturing long distance scrambling for Turkish in [Hoffman, 1995]. The CCG framework for Turkish over-generates for the above examples. Hoffman argues that there are exceptions to this phenomenon (i.e. our resource limitation principle(RLP)) in Turkish and gives examples such as (6.36) as exception where the two dative marked elements are far enough apart. According to her such exceptions support the hypothesis that the restriction on unique case in long distance scrambling is a processing limitation rather than a syntactic one. She argues that the intuition is that we have difficulty processing these sentences with two NPs with the

same grammatical function because we cannot easily disambiguate the predicate-argument structures of each clause and figure out which NP belongs to which verb[27].

(6.36) Esraya$_i$ Ahmet [ben-im e$_i$ yardim et-tig-im-i], Fatmaya soyle-di.
Esra-Dat$_i$ Ahmet [I-Gen e$_i$ help do-Ger-1S-ACC] Fatma-Dat say-Past.
'As for Esra, Ahmet told Fatama that I helped her.'

If we want to apply our incremental argument attachment strategy for Turkish, then the above example causes problem. But one can argue that in an incremental way the first NP is attached after the introduction of its corresponding verb, and when the parser reaches the second Dative NP, the first NP is already attached to the first verb and the parser sees no unattached Dative NP which violates the Resource Limitation principle. But it might be the case that RLP in Turkish is only valid for a group of grammatical functions and not for all.

Regardless of whether our claim for applicability of RLP to Turkish is justified fully or partially, the existence of RBP for long distance scrambling further categorises free word order languages into two groups: those such as Persian which obey RLP and RBP and those as German which doesn't obey RLP. In conclusion, we agree with Hoffman that constraints as RLP should be considered as performance constraints on scrambling. The cooperation of RLP and an incremental approach rules out examples which might look like an exception to this principle under a non-incremental approach to attachment of the arguments.

6.6.3 Comparison With Classical Word Order Rules

ID/LP [Gazdar et al., 1985] can be considered as the classical approach for representing flexible word order. ID/LP uses a set of immediate dominance (ID) rules and a distinct component for linear precedence (LP) which specifies the precedence relations between the right hand side sisters in ID rules.

Unlike a phrase structure (PS) rule which specifies two distinct relations of ID and LP at the same time, the order of the constituents in an ID rule is specified separately by LP component and in this way ID/LP format captures word order generalisations. The advantages

[27] We don't discuss the other exceptions which involve subject raising into object position. Those examples may suggest a similar discourse function for accusative marker *-i* in Turkish similar to *ra* accusative marker in Persian which hasn't been investigated for Turkish.

arising from factoring out of the ordering component from constituency rules are particularly evident in the case of languages with a flexible word order.

The linear precedence relations in LP component are binary relations and they can only be specified for two sister categories in the right hand side of an ID rule. As a result, no precedence relation can be specified for two categories which do not occur as sisters of a single ID rule.

Another restriction of classical ID/LP rule is the prohibition against referring to the categories inside the internal structure of phrases, in the LP relations. In other words, the LP relations can only specify relations between two sisters in an ID rule and not relations between one sister and another category dominated by the other sister.

In our approach we haven't employed ID rules and instead have used regular rules which allow different word orders. The possible word orders are restricted by a separate notion of word order binary constraints which restricts the possible order of grammatical relations in the paths.

A general criticism to binary relations (which also applies to our method) is that the relative precedence relation of any two categories in such a relation must be interpreted as being independent of the presence or location of a third category, i.e. ternary relations cannot be specified.

There are different approaches for extending the classical ID/LP notation, and the above characteristics can be relaxed or extended. For example the precedence relations can be restricted to be immediate precedence relations and not restricted to two sisters in the right hand side of an ID rule. In our approach, the precedence relations can be precedence or immediate precedence and they are not restricted to the right hand side categories of ID rules.

Reape introduces word order domains to deal with word order in Germanic. In his approach, the word order domains of the constituents that join with each other are merged. Unlike the word order domain in Reape's notation, in our approach we only allow one instance of a grammatical resource to be present in a word order path in each domain and each word order domain can consist of a set of parallel competing word order paths. Corresponding to each possible word order path in the word order domain, we have an activation measure.

The activation measure for a specific word order path is reduced if a word order constraint

is violated. The reduction corresponds to the strength of that constraint and the stronger the constraint, the bigger is the reduction. In the case of hard constraints, violation of a constraint makes the word order illegal and blocks that word order path.

In this way relaxation of word order constraints can be achieved. A general restriction of classical LP constraints is that they cannot be relaxed and they must always be satisfied. Uszkoreit [1985] proposes to extend these by use of complex LP constraints. In complex LP constraints, as long as at least one of the LP rules is satisfied the other LP rules can be violated. Our approach is a numerical extension to LP rules which allows the possibility of relaxing the LP rules. In fact our approach is more flexible than complex LP constraints and the degree of violation can be measured and a certain level of acceptability be introduced.

Similar to fuzzy logic sets, one can consider linguistic measures for referring to different relaxation possibilities for each word order rule and a linguist can use these for encoding the strength of the word order rules. Ideally, the relaxation of word order relations and their strengths should be derived from a corpus of texts, so that the most dominant word order gets the highest activation. In other words, these word order rules are statistically prevalent and are designed in such a way that the less plausible word orders get penalized and their activation gets reduced.

In our framework we have used the unmarked order as the most optimal path and deviations from this unmarked order are penalized. This approach can be considered as an extension to a notion of optimal parsing introduced in [Hawkins, 1990] based on typological research.

6.6.4 Comparison With OT based Competitive Models

Smolensky and Stevenson [1997] have recently proposed an extension to Optimality theory [Prince and Smolensky, 1993] for comprehension/parsing based on the GB approach. They consider constraints on the language processor that correspond to some of the grammatical constraints in theories such as GB and they specify a ranking of these constraints. A major shortcoming of their approach for processing verb final languages is that they consider the theta-criterion as a constraint with almost the highest ranking. For a language such as English this way of ranking the constraints and pruning the search space might be appropriate but in a verb final language such as Persian their solution faces problems.

In verb final languages subcategorisation information won't be available until the end of the sentence. Before processing the verb of the sentence other low ranking constraints of the grammar might rule out the correct parse and applying the theta criterion in an incremental fashion at the end of the sentence doesn't help with alternatives that have been already ruled out.

For parsing free word order languages , the cumulative sum of constraints from Syntax, Semantics, Discourse and world knowledge determines the grammaticality of an utterance and the preference for one alternative over another. It is not clear whether one can come up with a constraint ranking of these separate modules.

A general assumption in using a ranking of constraints in OT is not that the ordering of constraints are necessarily the same for all languages, they can be different Tesar and Smolensky [1999]. But if one language uses two constraint rankings as grammatical at the same time, then how OT can represent this multiple ranking? Does such a language exist?

The assumption of ranking faces problems in learning OT hierarchy of constraints. OT community has worked on different learning algorithms for deriving the hierarchy of the constraints Tesar and Smolensky [1999] that requires the learning machinery to reorder the hierarchy of constraints until it reaches the one that matches a specific language. This is not compatible with the way that a human acquires the constraints. If OT adopts a notion of acceptability measure for representing the hierarchy of constraints (as we describe in our work) the learning problem and the abrupt shift from one hierarchy to another will be removed and as we will describe the 'ganging-up' of constraints can also be captured in such model.

A recent example of a competitive approach based on OT is [Choi, 1996] which tries to extend LFG with OT. This extension to LFG is in conflict with some of the basic principles in LFG for capturing long distance scrambling.

Karttunen [1998] has shown that for implementation of OT in a finite state framework, one should restrict the OT model and a subset of it be considered. Another alternative is to use a cumulative and weighted approach to ranking the constraints[28].

A general criticism to Optimality Theory (OT) is that the ranking of constraints does not allow any cumulative effect in which a number of lower ranked constraints can compete against a higher ranking constraint. This so called 'ganging up' effect can be represented by

[28]See [Gibson and Brienhier, 1998] for another discussion on optimality ranking vs weighted constraints.

using a numerical representation. In order to overcome this problem one can implement OT as an exponential function of C^r [Rezaei, 1998]. Here **r** stands for rank and corresponds to the rank of the constraint. In this model **C** is a constant value for the model.

By assigning different values to **C**, one can obtain different OT implementations. If **C** has a small value, then a number of lower ranked constraints can counteract the effect of a higher rank constraint. If **C** is chosen to have a big value then the implemented model will correspond to the OT model where no ganging up is possible. This *algebraic* implementation of OT is more flexible but it has its own limitation and in its present form only allows a limited and uniform cumulative effect for all constraints.

With a change of combination function in our framework, this algebraic notion of optimality can easily be incorporated into our framework. For example in our model, we can consider an instance of C^r for the numeric values of the constraints, where **r** specifies the ranking of the constraint.

A further criticism to OT parsing model has been raised in the literature Hale and Reiss [1997]. In OT model of parsing, only the most harmonic alternative will be selected and the algorithm does not allow for a number of alternatives with a lower degree of harmony to compete in parallel with the most harmonic alternative, so that if the most harmonic alternative fails, one of the alternatives with lower degree of harmony be selected and the parse continues. In our work, we have adopted a parallel competitive model that allows a number of alternatives to be run in parallel, this also allows a degree of robustness to be incorporated into the parser in the future.

6.6.5 Psycholinguistic Aspects

In this section, we will highlight some of the aspects of the parser which is relevant to psycholinguistic research. Recall that in (6.28) repeated as (6.37), we gave an example of a garden path sentence in Persian.

(6.37) ali be mohammad goft [ke seab bedaham].
Ali to Mohammad told-3S that apple give-1S
'To Mohammad, Ali told (me) to give the apple.'

In (6.28), Mohammad cannot control the subject of the embedded clause, because it doesn't agree with the embedded verb and when the parser tries to attach the embedded verb

it discards its previous assumption and commits itself to a new analysis of the sentence in which *be Mohammad* is considered as long distance scrambling into the embedded verb. This is an example of *reanalysis*, which has been the focus of much study in psycholinguistics.

[Gibson and Brienhier, 1998] is one of the works in psycholinguistics which uses a restricted notion of parallelism in parsing. In our model we keep the two competing alternatives progressing in parallel and when one is blocked we switch to the other. The majority of other approaches in psycholinguistics use a serial model of computation, and when the analysis breaks down they either backtrack to another representation, or they *revise* the present representation e.g. by lowering a phrase in the tree structure corresponding to the parse. Most of the research in psycholinguistics provides evidence for the serial model.

In our model, we haven't used tree structure for representation in the second stage of parsing, instead the relations between phrases (i.e. linguistic processes) are modeled by dependency links between them. The parser generates new relations between chunks which are produced in the first stage as the parse progresses in the second stage. Using this distributed and flexible notion of representation in the second stage is reminiscent of the notion of assertion sets proposed in [Barton and Berwick, 1985]. Nevertheless we have used the tree structures for representation in the first stage and our approach therefore combines the two modes of representation.

Another aspect of our restricted parallelism for both local scrambling and long distance scrambling is that the parser cannot commit itself to an alternative before the head of the clause (i.e. verb) is attached. In this respect our model is closer to models in psycholinguistics such as [Pritchett and Reitano, 1990] which focus on information about θ-marking rather than the position of a phrase in a tree structure as advocated by works such as [Gorrell, 1993].

Nevertheless, our approach does not exactly follow any extant processing model in psycholinguistics. As we have outlined in this chapter, the implemented pipeline model uses a restricted notion of parallelism which is further constrained by word order constraints, subcategorisation information of the verb and the Resource Limitation Principle which we discovered for Persian.

6.7 Evaluation

In this section we will look at the evaluation of the parsing system and contrast it with the previous systems for parsing Persian. A wide variety of parser evaluation methods have been used and justified in the literature. Carroll et al. [1998] summarises some of these approaches. In general, these methods are divided into non-corpus and corpus-based methods. Since no corpus of texts or speech with scrambling are available for Persian, we will use a non-corpus evaluation approach.

One should also note that the parser is intended to reflect graded grammaticality judgments of Persian speakers. Hence it cannot be fully evaluated in the absence of appropriate psycholinguistic data on grammatical judgments by Persian speakers.

We will adopt a traditional approach to parser evaluation, by enumerating the construction types which are or are not covered by our parser. To improve this we will also discuss the interaction of some of these aspects in the parser.

From a computational perspective, the parser is a continuation of previous parsers built for parsing Persian. The scope of the coverage of parsers, their method of structural representation can be compared.

	PERSIS85	R. Ghasem91	Rezaei92	Rezaei93	Riazati97	SHIRAZ	Our system
Approach	Production Rule	Conceptual Dependency	ATN	ID/LP	KIMMO PATR	feature str/type	PATR path/LP
Parser	Bottom-Up (BUP)	Procedural	BUP Top Down	BUP	BUP	BUP	BUP
Tokenisation	NO	NO	NO	NO	NO	YES	NO
Morphology	NO	NO	NO	NO	YES	YES	NO
Explicit Ezafe	YES	YES	YES	YES	YES	NO	YES
Coordination	YES	NO	YES	NO	NO	YES	NO
Local Scram.	V-final	unrestricted	YES	YES	Limited	V-final	YES
Complement Cl.	YES	NO	NO	YES	NO	NO?	YES
Relative Cl.	YES	NO	NO	YES	NO	YES	NO
Long Dis. Scram.	NO	NO	NO	Fronting	NO	NO	YES
Control	NO	NO	NO	NO	NO	NO	YES
Multiple Parses	NO	NO	NO	YES	YES	YES	NO

Table 6.4: Comparison and Evaluation

In Table 6.4 we have contrasted the implemented system with the previous systems for parsing Persian. The system was developed with the goal of complementing the capabilities

of previous systems and it has its limitations.

The above lists the main constructions that the parser can analyse. We discussed the main features of earlier parsers in Chapter[29] 4. The present parser like most of those parsers does not handle either tokenisation or morphology levels of representation. Its strength, as is shown, is in capturing local scrambling and long distance scrambling. In capturing local scrambling, it has the advantage of taking into account the graded grammaticality which has been neglected in Rezaei [1992] and Rezaei [1993] the two parsers that considered the constraints on scrambling. The parser has also taken into account *control* in Persian and its interaction with long distance scrambling.

Note that in most examples that we considered, we justified the operation of the parser by appealing to the resource limitation principles and *hard* constraints like control. These hard constraints have priority over the constraints for graded grammaticality or the *soft* grammatical constraints. This suggests that the graded grammaticality idea has limited usefulness for a language like Persian where resource based performance constraints restrict its word order. Nevertheless the approach can be employed for parsing free constituent order languages where the word order is not restricted by these hard performance constraints. Even for Persian, this approach can be adopted in applications like Computer Assisted Language Learning (CALL) to help the language learner with problems of specificity, object marking and word order.

We can also briefly consider the interaction of some of these construction types in order to convey further information about the implemented system, its capabilities and weaknesses.

Local and Long Distance Scrambling

The parser can analyse a range of examples involving local and long distance scrambling (LDS). The implemented system is restricted to allowing LDS only for PPs. This limits the interaction of LDS and local scrambling to PPs. For these PPs, the competition between the verbs of a sentence (main and embedded clauses) is also considered in the parser, but further semantic information is needed.

The parser implements for the first time the two performance constraints on scrambling in Persian, namely Resource Limitation Principle (RLP) and Resource Barrier Principle (RBP).

[29] See Page 90.

These two constraints reduce the range of possibilities and over-generation drastically. This is an improvement to capturing long distance scrambling (LDS) in the only parser that has tried to implement it, i.e. Rezaei [1993].

Scrambling and Control

The parser can handle examples in which local scrambling interacts with the control phenomenon. This causes no problem for the system, but the different possibilities of the interaction of LDS and control cause more problems.

In our study we have only concentrated on LDS for PPs, and the parser can handle examples in which LDS interacts with the finite control phenomenon in Persian. But as we argued a layer of non-monotonicity should be added on top of the system to allow the system to revise its earlier decision when finite control fails inside the embedded clause as a result of disagreement with the verb of the embedded clause. The effect of finite control on this type of competition is more straightforward. The evidence showed that syntactic constraints have the final say in such a competition and the parser takes these issues into account.

6.8 Summary

An important aspect of the parser is the notion of graded grammaticality in the parser. One parsing solution might be less grammatical than another solution. In some cases the only parsing solution might be below the accepted level of grammaticality. In this way robust parsing can be achieved.

This framework provides a method for adding the notion of graded grammaticality to the principles of the grammar. In traditional approaches to principle based grammars a principle can be satisfied or violated. In our view some of the principles of the grammar can be violated, but the overall relaxation of the principles (when added together) should not reduce the acceptability of the solution below a certain level.

The acceptability of a particular solution is reduced by a factor whenever a principle of the grammar is violated. This factor depends on the contribution and importance of that specific principle.

At present we have chosen arbitrary numbers to model the relative grammaticality of

different word orders and scrambling in Persian. The exact value of these numbers and the relative importance of different principles of grammar in Persian is a topic of research which needs to be complemented by psycholinguistic studies and grammaticality judgment of different Persian speakers.

Some of the constraints can be derived based on corpus analysis of large texts. For free word languages, where the order of arguments is flexible, statistics about the order of arguments can be used as a measure to weight the possible alternatives in the path sets, or to specify the most plausible parse so far.

By adding features to the word order rules, we can introduce more complex word order rules to take into account features such as animacy.

We also studied the interaction of control and long distance scrambling in Persian and we introduced two performance constraints in scrambling: the Resource Limitation Principle (RLP) on local scrambling and the Resource Barrier Principle (RBP) in long distance scrambling. These two performance constraints restrict the possible instances of long distance scrambling in some free word order languages.

Chapter 7

Conclusion and Further Work

7.1 Summary

Free word order languages create numerous problems for designing parsing and natural language processing systems. In Chapter 4 we showed previous examples of systems for processing Persian. These systems either do not deal with examples of local and long distance scrambling, or if they have dealt with such phenomenon, they suffer from a great amount of backtracking. A possible solution to this problem is to use a mixture of deterministic approaches and parallel processing methods to avoid the problems of backtracking. So parallelism could be a solution for this problem.

From another perspective, the examples with scrambling are interlinked with the level of discourse representation and a notion of graded grammaticality. Not all the permutations of a sentence have a similar intonation and not all the permutations are as acceptable as the others. In this thesis our focus has been on graded grammaticality and acceptability for constraints on word order rules, in order to find the major parameters and a way to implement them in a restricted parallel and competitive architecture. We have also studied some of the discourse marking of *rā* in Persian which interacts with its syntactic marking, but our focus have been on syntactic representation and syntax-based processing.

In order to analyse scrambling in Persian we have looked at a different range of phenomena which impose constraints and restrictions on scrambling. These range from blocking constraints to fuzzy notions of specificity in Persian, in addition to the ambiguities in subject/object marking in Persian. To develop a competitive framework that can adequately

capture and represent scrambling in Persian, the notions of parallelism, graded grammaticality, structural and numeric constraints have been studied in this thesis. In parallel the possibility of developing an underlying formal algebra for such competitive approach has also been considered to provide a more solid foundation for research in this area. The contributions of the thesis are in three areas of:

- Linguistic analysis of word order and formal analysis of scrambling and movement constraints in Persian.
- Computational study of using fuzzy word order rules Rezaei [1999]and sets in a parallel/competitive language processing framework.

Linguistically, we have discussed in detail the constraints on local scrambling and long distance scrambling in Persian word order. The interaction between specificity in Persian and *ră* object marking and the syntactic/semantic process of disambiguation of subjects and objects add a further level of complexity for analysing local scrambling which needed to be investigated and formalised. Furthermore, other functions of *ră* such as a syntactic/discourse marking were analysed and contrasted with analogous markings in Arabic and Turkish to help us analyse the different functions of *ră*.

In order to analyse constraints on long distance scrambling, the controversial canonical position of complement clauses was investigated and we argued for the extraposition of embedded clauses from a pre-verbal position. Based on this analysis we captured the two different cases of long distance movement i.e. long distance scrambling and fronting. We proposed a structure for representing embedded clauses and their extraposition. This structure takes into account the existence of barriers in front of long distance movement, e.g. in relative clauses. In addition, this structure can represent examples of case attraction in Persian embedded clauses.

In Chapter 3 We have looked at the examples of control in Persian in order to analyse the more complex examples in which control interacts with long distance scrambling. This part of the analysis provides an insight to representing the constraints on complex examples of garden path in Persian in Chapter 6. It is the first time that the interaction of control and long distance scrambling is considered for such analysis.

Computationally, we introduced numeric word order relations and Resource Limitation

Principle (RLP) in order to represent the constraints on scrambling and word order in a competitive framework. We introduced a stochastic word order precedence relation and a set structure for capturing case competition and the complex disambiguation process between *rā* marked constituents and non-marked constituents. The approach was contrasted with other competition based approaches, and we claimed that the fuzzy word order rules can be extended as an algebraic and numeric version of Optimality Theory (OT) based model for representing word order constraints.

The study argued for a resource-based model of scrambling and we introduced two principles based on such model for implementing constraints on scrambling in spoken Persian. The Resource Limitation Principle (RLP) is an example of such a restriction on long distance scrambling in Persian that blocks some instances of this phenomenon. We claimed that this constraint could be used to distinguish between different types of free word order languages. The Resource Limitation Principle (RLP) has been a performance constraint which has been modeled by the use of stochastic word order rules. This principle prevents the existence of two constituents with the same case marking in a clause. We have introduced another performance constraint, the Resource Barrier Principle (RBP) which restricts the possibilities of long distance scrambling.

In order to use parallelism in the parsing architecture, we have implemented the parsing system as a two stage pipeline of chunking and argument attachment which work concurrently. Another advantage of using a pipeline model has been the advantage of using two different types of grammar rules and constraints in each stage. The first stage (chunking) uses Context Free Grammar (CFG) rules, while the second stage for argument attachment uses finite state rules and stochastic word order precedence relations. The implemented parser avoids the inefficiency of previous approaches for parsing Persian and employs fuzzy sets for resolving conflicts and competition among possible alternatives.

7.2 Further Issues

Although our computational model has been useful in identifying basic mechanisms for representing the constraints on scrambling in Persian, our work in this area has been limited by our use of a small-scale grammar of Persian. The work can be extended in different ways in linguistic, theoretical and computational aspects.

Head Order/Language	Turkish	Persian	Arabic
Adjective Noun	Adj-N	N-Adj	N-Adj
Genitive Noun	Gen-N	N-Gen	N-Gen
Relative Clause	Rel-N	N-Rel	N-Rel
Complementiser Clause	Comp-Cl	Comp-Cl	Comp-Cl
Pre/postposition	Po	Pr	Pr
Verb Position	SOV	SOV	VO
Auxiliary verb	V-Aux	V-Aux	Aux-V

Table 7.1: Phrases in Persian, Turkish and Arabic

marker/Language	Turkish	Persian	Arabic
marker	(-i,#)	(*rǎ*,#)	-o/-on
Syntactic marker	Object	Object	Subject
Semantic marker	Specific/Nonspec.	Specific/Nonspec.	Definite/Indef.
Discourse marker	(–,/Focus)	(Topic2,–)	Topic

Table 7.2: Discourse and Syntactic markers

7.2.1 Syntax and Pragmatics

In our study we explained some of the similarities between the grammar of Persian and the grammar of Turkish and Arabic. This is summarised in 7.1.

It will be useful to study the presence of the Resource Limitation Principle for Turkish and Arabic. A general examination of the role of discourse markers in these languages would help to build a better understanding of the way that they compare with grammatical markers. This would be especially beneficial for analysing the behavior of *rǎ* in Persian which has both a discourse and a syntactic function. Table 7.2 summarises preliminary findings. In this table, a (*rǎ*,#)[1] pair acts as a specific object marker in Persian, but only the former (i.e. *rǎ*) acts as a secondary topicalisation discourse marker. In Turkish, a (-i,#)[2] pair acts as specific object marker, *-i* but unlike its counterpart *rǎ* does not play a Topicalisation role. But the non-marked pre-verbal position for objects in Turkish exhibits a focus marking which has not been investigated in Persian. In the Arabic case the same morpheme is used to mark subjects and topics.

These constraints and their role in local and long distance scrambling is independent of the implementation that we choose for parsing or analysing these languages.

[1] # is used to show the cases when *rǎ* or the other alternative is missing, i.e. empty marker.
[2] # is used to show the cases when -i or the other alternative is missing, i.e. empty marker.

7.2.2 Parsing

A general extension to the model is to use chart parsing techniques, to deal with the non-determinism in constituent boundaries, especially if *ezafe* is not explicit in the text. In our study we assumed that *ezafe* is explicit in the text. This is true of spoken language, but in Persian texts it is not represented explicitly. Among the parsing systems that we studied for Persian, only SHIRAZ MT system assumes no explicit *ezafe*. The Resource Limitation Principle can also restrict the number of competing alternatives for such texts and the same competition mechanism by use of fuzzy sets can be used.

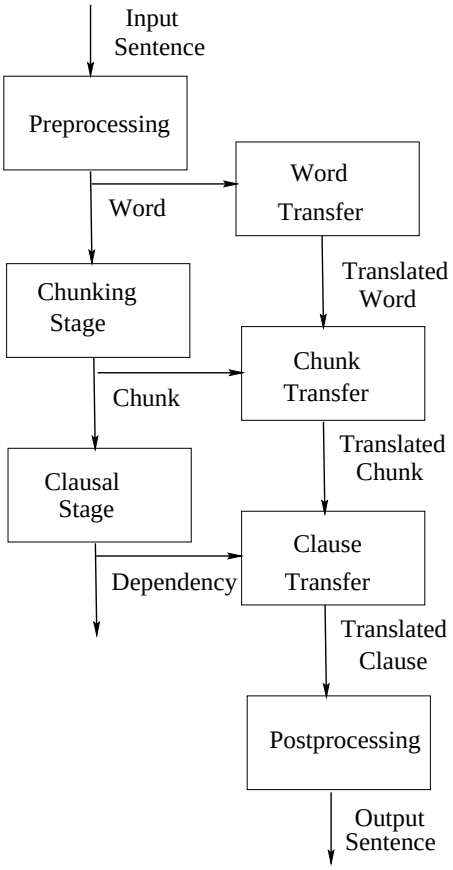


Figure 7.1: Pipeline Transfer MT

Other future possibilities are to use the system as the parsing stage in a pipeline machine translation (MT) system. The output from the two stages in the pipeline can be fed into the input of a 3 stages transfer pipeline to construct a translated sentence corresponding to the

input sentence. Figure 7.1 illustrates the transfer modules for word, chunk and clause and their connections with the present parser in such a model. Note that each of these units in the pipeline can have one (or more) look-ahead chunk. So this does not imply that words are transferred without considering their context. One of the preliminary long term goals of this research was development of a parser to be used in an MT transfer system for Persian.

Another area to look at is the psycholinguistic studies of Persian language and development of a psycholinguistic based parser for parsing Persian. Some of the results of this study can also be tested and used as a starting point for such a research.

7.2.3 Towards a Channel Algebra

In our study we represented grammaticality as the result of competition between communicating processes. These processes compete for the limited grammatical resources. The nature of competition and the role of semantics is another area which need to be investigated. The performance/competence distinction and models that incorporate these two in an appropriate way needs to be investigated. And we should move towards developing a formal foundation for this.

In the rest of this section we will have a closer look at a possible direction to derive a formal algebra for such a model. In this proposal (for future research) we show that linguistic processes can be defined and these processes can communicate with other processes via grammatical channels Rezaei [1997].

Introduction

The dominant approach in Computational Linguistics divides the problem of language processing into developing a grammar for the language in terms of a set of rules (or constraints) and developing a processing algorithm (or parser) in which rules are selected and applied in bottom-up, top-down or a combination of the two strategies [Joshi, 1987]. The rules and structures are "statically" used by the parser.

Extensive research has been done on the formal specifications of these grammars and properties of different parsers. Recent research has been attempted to extend these formalisms with probabilities extracted from a corpus and to develop a stochastic model of language [Brew, 1995]. In other words, such approaches have the goal of adding "performance" mea-

sures into the "competence" grammar.

Many of the attempts in this area have tried to add a layer of probabilities on top of the existing rule based formalisms and these have raised new interesting problems. In some cases, some of the fundamental mechanisms in current theories cannot be simply extended by the addition of probabilities. For example in LFG, the * under-specification causes problems in developing a stochastic model [Kaplan, 1996], [Rezaei and Crocker, 1997].

A radical departure from these approaches are "dynamic models" for language processing [Milward, 1994]. In these approaches the interpretations are built incrementally from left-to-right. A word introduces a change or transition from one state to another. The probability of transition to another state is dependent only on the current state and current word and the grammar is in the form of a Markov model [Tugwell, 1995]. These probabilities are calculated based on a corpus of a language and different factors such as lexical frequency, co-occurrence probabilities.

A dynamic model specifies the possible states and state transitions. In some of these dynamic approaches, e.g. [Philips, 1996], the relation between parser and grammar in the model is not very transparent and "the parser is the grammar". These finite state models for language modeling have been extensively criticised for their inability to describe structures which involve an indefinite amount of nesting. These models also do not take into account the possibility of scrambling and free word order, and issues such as syntactic control in the corpus.

An intermediate approach is to use two levels of modeling and use the previous stochastic and dynamic approaches for the constituent level, and at the clause level introduce and use a richer notion of dynamism which takes into account scrambling and syntactic constraints such as control. This enriched dynamic model can be an extension to the present formalisms such as LFG.

In the Computer Science field, a series of developments in dynamic modeling and process models have also been investigated. In the rest of this thesis we will look at one possible attempt for using such dynamic models and notions in Computational Linguistics and will investigate its application for processing syntax. This chapter is an attempt to introduce such dynamic framework for the language processing system that we described in the previous chapter.

In our work, instead of "state", we consider and use a richer notion of dynamism called "process" and we specify the internal structure of these processes. For specifying an algebra and language for these dynamic processes, we will turn to research in "Process Algebras" [Abramsky, 1996], where dynamic systems are modeled as communicating processes. The main obstacle in this regard is the development of the notion of communication among linguistic objects.

[Fujinami, 1996] is another approach which looks at computational linguistics from a process algebra view point. Fujinami is mainly concerned with utterances and gives a process algebraic account of discourse and dynamic semantics. He uses an extension to π-calculus [Milner, 1993] for this purpose.

Our work is a syntactic complement to his model and it is concerned with looking at competence theory and dynamic syntax from a process algebra perspective. We introduce the essential principles for a formal process algebra for such purpose and will contrast ours with the process algebra proposed by Fujinami. We look at the problems of extending such framework with probabilistic operators and notions. We have developed a framework that deals with different notions such as control and long distance scrambling in a competitive, *communication based* approach. The dynamic notion of *communication* in communicative based models, such as process algebras, can be contrasted with the notion of *unification* in constraint based approaches. Unification and feature passing in constraint based approaches have also a dynamic aspect. But communication and unification differ in respect to "resource sensitivity". In Section 6.1 we introduced the notion and in the previous chapter introduced two resource based performance constraints.

The phenomena such as argument attachment in natural languages are inherently resource based and most linguistic theories use some mechanism of resource sensitivity for argument attachment. One important aspect of the notion of communication is that it is "resource based". When a process communicates a value, it will be consumed by another process. It has been shown that the process algebras which are built on top of this notion of communication are mathematically compatible and consistent with Linear Logic (see [Miller, 1992],[Fujinami, 1996]). In our study we use the notion of communication and apply it as a mechanism for argument attachment.

One of the aspects of using a process algebra for linguistic modeling is that it will provide

a solid framework built on top of linear Algebra. Another advantage of such a model is that there is already a large amount of research on the formal and mathematical modeling of dynamic systems; language will be considered as another system for analysis.

Work such as [Johnson, 1997a] try to introduce a complete resource based notion in order to replace unification, and re-express all feature well-formedness constraints in terms of such feature resource dependencies. In our work, we restrict the domain of our study to using resource based notion of "communication" and use this notion only as an alternative perspective to feature interaction for argument attachment, local scrambling and long distance scrambling. It is to be demonstrated in future research whether communication can be used as an alternative for capturing all properties of f-structures and the mechanism of unification, but we have not studied this in our work.

Specifying scrambling in terms of communication provides a basis for development of a formal foundation for long distance scrambling. In constraint based approaches such as LFG, long distance scrambling is added on top of the theory, without much consideration for specifying a formal basis for it or for the theory and notions such as barrier over the movement of constituents are not fully considered. Developing a formal foundation for long distance scrambling and local scrambling, based on communication-based models helps to bridge this gap.

Another motivation for our work is that the application of process algebras for linguistic modeling will also contribute to the development of new models which are tailored for linguistic analysis. The different constraints in language modeling provide a limited domain and a new direction for development of stochastic process algebras and this will open a new direction for the theoretical research in foundations of computer science and process algebras.

Finally, our incremental, resource based approach, shares in spirit some similarities with theories such as LFG (Lexical Functional Grammar), but our main focus is on defining the notion of "grammatical channels" for communication between processes. We have investigated the possibility of representing grammatical relations in terms of resources and communication of resources.

In the rest of this section we deal with the conceptual and theoretical aspect of the work. We attempt to provide a part of the necessary conceptual framework by attacking the question: what sort of linguistic object is a process? what should be the domain for interaction of these

processes and what is the medium for their communication? Is there any notion in linguistic theory that can be used as communication channel?

Process Structures and Grammatical Channels

The major building block in our model is process structure. We assume that structures or constituents like NP or PP exist in languages. One simple approach for representing this in terms of process algebra is to use the recursive definitions to specify grammars.

(7.1) $\text{NP} \stackrel{def}{=} \text{N}$
$\text{NP} \stackrel{def}{=} \text{ADJ.NP}$

Or alternatively with choice operator:

(7.2) $\text{NP} \stackrel{def}{=} \text{N} + \text{ADJ.NP}$

The first problem that arises is when one wants to add features to these processes. One possible solution for this is to represent context free grammars by communicating processes. The previous example can be recaptured by (7.3).

(7.3) $\text{n(x)}.\overline{np}(\text{y}) \mid \text{adj(z)}.\text{np(w)}.\overline{np}(\text{w})$

Now for each phrase we have considered a channel and the feature values and annotations can be passed as communicated values. In our notation the channel for communication might be a telephone line or open air. One process emits a value or message over a channel and the other process having access to the same channel will receive the message[3]. When the sender sends the message **m** over channel **c** by $\overline{c}(m)$, the receiver will receive it by executing the process c(x) which will receive the message and upon receiving the message, will bind its free variable **x** to it, that is $\{m/x\}$. The sender channel is marked with a line over it marking it as negative polarity, that communicates with a positive polarity channel with the same name,

[3]The reader is assumed to be familiar with π-calculus and basic parallelism notions in concurrency. For a general introduction see Milner [1993] or Fujinami [1996].

but no extra marking. But we need to consider additional machinery and general joining processes that accept two feature structures and give back the result of joining/comparing them together.

Note that this channel based notation is more powerful than the previous recursive notion. In this notation we can also express type 1 and even type 0 grammars. We only need to consider a minus (-) polarity communication for each terminal or non-terminal in the left hand side of a rule, and a plus (+) polarity communication for each terminal or non-terminal in the right hand side of the rule. This new channel based perspective corresponds to a bottom up realisation, while the former recursive one corresponds to the top down realisation of a grammar.

In this way we can represent constituents or structures as process structures. And we can view them dynamically and associate time-period, locality, activation and other measures with them. The main issue is that these processes should be able to communicate with each other and interact, and hence we can have communicating and interacting process structures [Rezaei, 1997].

Another problem that arises is when one wants to capture scrambling and free word order. We can simply consider ID rules by using the parallel operator " |". For example another way to represent example (7.4) is illustrated in (7.5).

(7.4) $\quad \mathrm{np(x).v(p).}\overline{vp}\mathrm{(y)} \mid \mathrm{v(z).np(w).}\overline{vp}\mathrm{(w)}$

(7.5) $\quad \mathrm{(np(x)|v(p)).}\overline{vp}\mathrm{(y)}$

The problem that will arise is when we want to introduce the linear precedence rules to restrict some of the possible word orders. One solution is using guards. Fujinami [1996] considers an extension of π-calculus with guards: a guard operator "$\lfloor$" can be considered as a generalisation of the prefix operator.

(7.6) $\quad \overline{c}(m).\mathrm{p(y)} \mid \mathrm{c(x).}\overline{p}(n)$

(7.6) can be decomposed into a set of primitive processes and the constraints on them. The two precedence constraints $\overline{c}(m) \prec \mathrm{p(y)}$ and $\mathrm{c(x)} \prec \overline{p}(n)$ can be separated by a guard operator

and be written as (7.7) as pointed out in [Fujinami, 1996].

(7.7) $[\overline{c}(m),\ \mathrm{p(y)},\ \mathrm{c(x)},\ \overline{p}(n)\ \mid\ \overline{c}(m) \prec \mathrm{p(y)},\ \mathrm{c(x)} \prec \overline{p}(n)]$

In other words guards provide the necessary notion for abstracting away the precedence constraints on the processes. This is analogous to the notion of separate Linear Precedence in ID/LP notation in computational linguistics, where one abstracts away the notion of precedence and separate it from the dominance in CFG rules. Two parallel processes can happen in any order, the same way that two phrase on the right hand side of an Immediate Dominance rule can appear in any permutation. The LP constraints put constraints on the possible alternatives and for process models the guards can be used to restrict the possible order of execution of processes, but implementing guards might face some theoretical problems.

In addition to using ID notation, one can also use another conceptualisation for expressing local scrambling. Turning to finite state models we can represent the previous example by using a flat structure such as (7.8) and then specify the linear precedence relations for each clause and its constituents. This is in contrast to ID/LP notation in which the LP constraints are specified for the right hand side constituents of an ID rule.

(7.8) $\mathrm{vp(x).v(p).}\overline{vp}\mathrm{(w)}\ \mid\ \mathrm{vp(x).np(p).}\overline{vp}\ \mathrm{(y)}$

If we want to conceptualise LP constraints according to this new idea for representing local scrambling, then we need to introduce a notion of *locality* for imposing the linear precedence constraints.

Grammatical Channels and Mobility

We also need to elaborate on the grammatical relations and the way we conceptualise them. In the previous example we can change our focus and identify each NP by the grammatical relation that it can play. Then (7.8) can alternatively be represented as in (7.9).

(7.9) $\mathrm{vp(x).v(p).}\overline{vp}\mathrm{(w)}\ \mid\ \mathrm{vp(x).subj(p).}\overline{vp}\mathrm{(y)}\ \mid\ \mathrm{vp(x).obj(p).}\overline{vp}\mathrm{(y)}\ \mid$

One can go one step further and look at each grammatical relation (such as subj, and obj) as a linguistic resource. Under this perspective each nounphrase introduces a positive resource (e.g. subj()) which will be consumed or cancelled by the corresponding negative resource of the verb (e.g. $\overline{subj}()$). This shows that the grammatical relations can be represented as communication between verbs and nounphrases in a sentence. Neither of the examples in (7.8) and (7.9) demonstrates this aspect. In (7.8) we have hidden the notion of grammatical relations as a feature in the feature structures of the verb and np. Unlike grammatical relations other features doesn't have this complementary (positive and negative) characteristics which is suitable for a resource sensitive conceptualisation. In (7.9) the grammatical relation corresponding to the nounphrase is used as a channel name, but the corresponding grammatical relations or resources of the verb are represented as features. What we need is a way to introduce a notion such as clause in which these communications can happen locally. If we adopt this framework, then we can also express linear precedence relations over these relations and express for example that a subj resource should precede an object resource. The introduction of locality for imposing linear precedence relations will also provides us with a notion of local domain for local communications over channels such as subj and obj.

Another advantage of introducing a notion for locality is that we can impose barriers in front of constituents which want to enter or exit the locality. Then one can express long distance scrambling in terms of movement of constituents by using mobile channels (in *pi*-calculus) or mobile processes.

One problem with π calculus is that it does not allow such a notion of local communication to be represented directly. We need a notion which allows us to introduce a new abstraction, a notion which specifies a boundary enclosing a group of local communications. Recent developments in Process algebras have extended π-calculus to accommodate such property. One such abstraction is *ambient* [Gordon and Cardelli, 1998].

The notion of grammatical channels for modeling grammatical relations as primitive elements in the theory can be used and further constraints on channels can be introduced. In the last chapter we introduced some fuzzy binary word order constraints that can also be incorporated into a channel algebra.

To deal with Long Distance Scrambling (LDS)[4] one can also use a mechanism which can be considered as a communicative replacement for functional uncertainty in theories like LFG. Under certain conditions (e.g. barriers theory of GB) some channels can be passed or exported from one clause to an embedded one.

A hypothetical example of this is shown in Figure 7.2 and Figure 7.3.

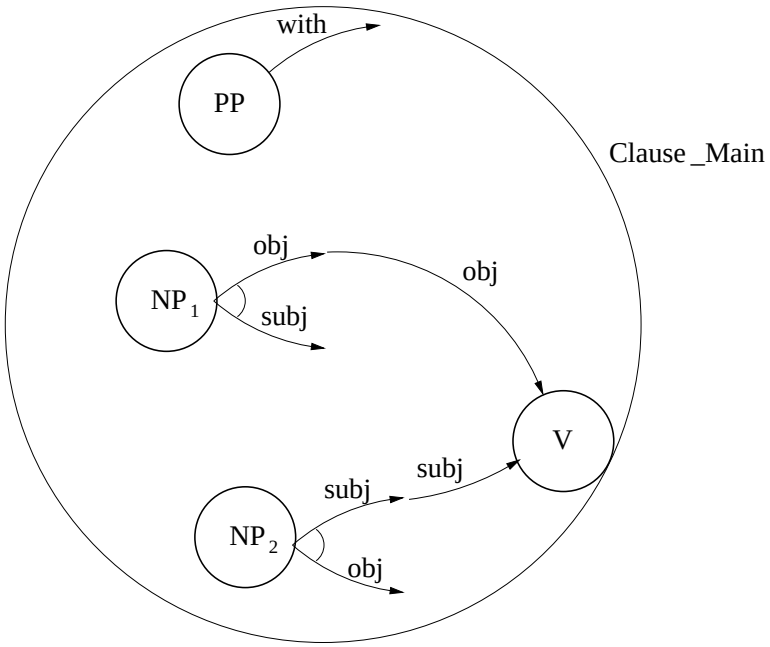


Figure 7.2: Before exporting, and after local communications

What one needs to introduce is the notion of locality so that the local word order constraints that we explained in last chapter can be enforced and the long distance scrambling be modeled as communication between localities. Ambient provides us with such a formal notion of locality.

In such a model one can distinguish between three kinds of process structures. First there are the clause processes in which communication can occur. Inside a clause there are process structures like unmarked NPs and marked NPs (with preposition or postposition) which compete with each other for the grammatical resources of the clause, such as subject and object. The resources are offered by another type of process structure such as verb. In general, a process structure may receive and/or offer a number of resources at the same time.

[4]In LDS a constituent will be moved across the boundary of two clause boundaries.

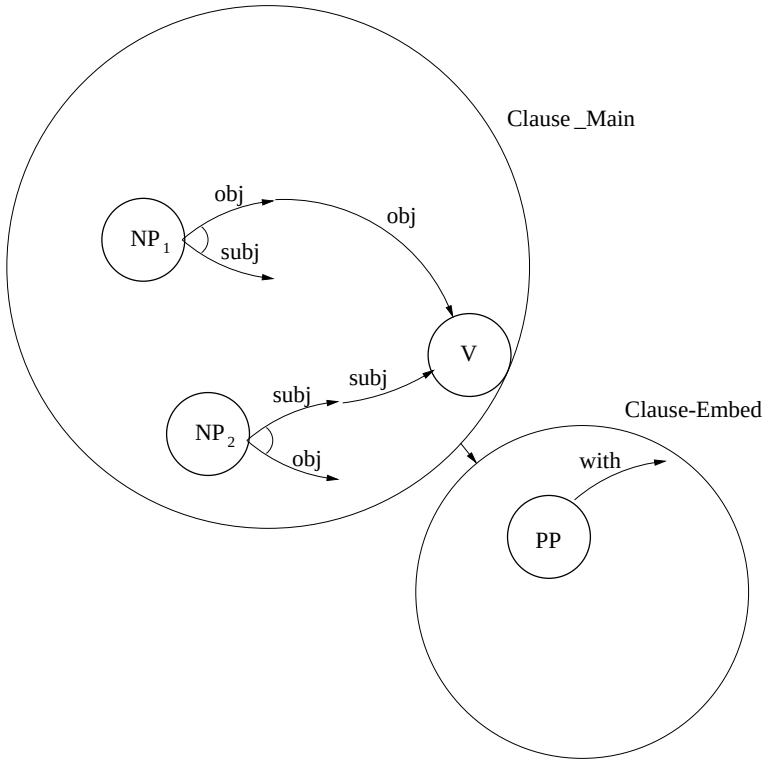


Figure 7.3: After exporting and local communication

The basic abstraction that we use for representing clauses is *ambient*. According to Cardelli and Gordon [1997] an ambient is written as n[P], where n is the *name* of the ambient, and P is the process running inside the ambient. n[P] is understood as an ambient or locality, in which P is actively running, and P can be the parallel composition of several processes.

An ambient provides us an abstraction for capturing locality inside constituents such as clause. Another interesting property of an ambient is that it can also include a set of ambients. This notion allows one to model the embedding of clauses inside other clauses. The other useful abstraction relevant to our research is that in *ambient calculus* there are notions called *capability* for allowing entrance into or exit from an ambient. We can use these notions to model barriers in long distance scrambling. Finally an ambient can move as a whole.

The notion of channel makes the communication medium explicit and gives a name to it, while the notion of ambient provides a local boundary for such communication to take place and gives a name to that locality. The word order constraints on local and long distance scrambling can be represented for these ambients.

Note that for representing the internal structure of phrases one can use the sequence operator and there is no need to consider ambients at lower levels of linguistic representation. This is analogous to the two level modeling of phrases with fixed word order and phrases with flexible word order in the previous chapter.

In the last chapter, we illustrated some of the word order constraints that can be defined for Persian word order. The possible channel combinations are restricted by channel order constraints which are imposed on channel pairs. They are of the form:

(7.10) chnl1 $\prec^{no}$ chnl2

The **no** contributes to lowering the activity of a path. Another constraint on channels are that they can only be allocated once. This is the Θ-criterion. We represent this in our framework as a hard precedence constraint. e.g. subj $\not\prec$ subj. Violating a hard constraint makes the candidate inactive and hence closed. This contributes to closing down of some paths and reducing the number of alternatives.

The long distance scrambling constraints can be applied as barriers in front of communication of mobile channels.

The channels in our model are binary communication links and hence are more restricted than π-calculus channels. In addition an uncertainty number is associated with each of them that shows the level of activity of the channel (path). We have used sequence, parallel and choice operators for constructing process structures. The sequence operator is needed for constructing a sequence of processes and the parallel operator is needed for capturing parallelism among processes. We need to have a choice operator to represent the choice between two competing alternatives. In addition, we have used a time precedence binary operator to represent the channel precedence constraints.

In this section we were mainly concerned with the structural and communicative aspect of a grammatical process model or algebra. Another important aspect that we didn't discuss here is the resource competition and commitment strategies that interact with the activation

measures. Some of these were highlighted in the implementation that we discussed in the previous chapter.

As we explained, a channel resource cannot be allocated twice in a channel sequence. We elaborated on a structure called path set (or channel set) which provides an efficient mechanism for a process to compete for one or more channels at the same time. This structure allows a set of paths or threads to progress in parallel. The competition strategy that we have adopted is a partial commitment strategy. We are following neither committed choice nor incremental commitment strategies. In committed choice strategy (e.g. in Parlog), a process must commit itself to one of the successful choices and discard (de-activate) the others, while in incremental commitment strategies (in NLP) a single choice is committed to and in case of deadlock or failure, by backtracking or reanalysis another choice can be adopted. As we explained in the previous chapter, we have used a partial commitment strategy, and all active channel paths are partially active at the same time, but the most active path will win at the end.

The choices for possible channel paths are restricted by channel order constraints and resource limitation constrains. Hence these context dependent constraints reduce the range of possibilities and make the strategy decidable. Put another way, we have introduced a notion of partial and soft commitment, which is fitted into the general model.

We will let all competing paths be active in parallel and will commit to one path as late as possible[5]. The path with highest activity will be the winning path, if its activity level doesn't go down.

It is worthwhile to investigate the possibility of developing a process algebra for this framework in the future. This will be an instance of discrete time probabilistic process algebras.

[5]This is the clause boundary position, where a choice is committed to.

Bibliography

Steven Abney. Partial Parsing via Finite-State Cascades. In *Proceedings of the Workshop on Robust Parsing at Eighth Summer School in Logic, Language and Information*, pages 8–15, August 1996.

Samson Abramsky. Retracting Some Paths in Process Algebra. In *Proceedings of CONCUR 96, number 1119 in Lecture Notes in Computer Science*, pages 1–17, Springer Verlag, 1996.

Tor A. Afarli. A Promotion Analysis of Restrictive Relative Clauses. In *The Linguistic Review, 11*, pages 81–100, 1994.

Gul A. Agha and Carl Hewitt. Concurrent Programming Using Actors. In *A. Yonezawa; M. Tokoro (eds) Object Oriented Concurrent Programming*, Cambridge, Massachusetts, 1987. MIT Press.

Ali A. Aghbar. *Case Grammar and Persian Verbs.* PhD thesis, Georgetown University, 1981.

Kazimierz Ajdukiewicz. Die Syntaktische Konnexiat, 1:1-27. English Translation in Storrs Mccall (ed), Polish Logic 1920-1939, Oxford University Press, pp. 207-231. In *Studia Philosophica*, 1935.

Mahdi Meshkat al Dini. *dastur-e zabăn-e fărsi bar păye nazarye gashtări (An Introduction to Persian Transformational Grammar).* Ferdowsi Univ., Mashhad, 1987.

Markus Bader and Ingeborg Lasser. German Verb-Final Clauses and Sentence Processing: Evidence for Immediate Attachment. In *Perspectives On Sentence Processing, edited by Charels Clifton, Lyn Frazier and Keith Rayner*, pages 225–242, Lawrence Erlbaum Associates, New Jersey, 1994.

Yehoshua Bar-Hillel. A Quasi-Arithmetical Notation for Syntactic Description. In *Language, 29*, 1953.

G. Edward Barton and Robert C. Berwick. Parsing with Assertion Sets and Information Monotonicity. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence (IJCAI 85)*, Los Angeles, California, 1985.

Mohammad Reza Bateni. *tosif-e săxteman-e dasturi-ye zabăn-e fărsi (Discription of Persian Syntactic Structure).* Amir kabir Publications, Tehran, 1970.

T. Becker, A. K. Joshi, and O. Rambow. Formal Aspects of Long Distance Scrambling. In *Proceedings of 5th Conference of theEuropean Chapter of the Association for Computational Linguistics (EACL'91)*, pages 21–26, 1991.

R. Bisiani and A. Forin. Parallelization of Blackboard Architectures and the Agora System. In *V. Jagannathan, R. Dodhiawala, and L.S. Baum (eds), Blackboard Architectures and Their Applications*, pages 137–152, London, 1989. Academic Press.

Joan Bresnan. Optimal Syntax: Notes on Projection, Heads, and Optimality. MS. Stanford University, 1996.

Joan Bresnan and Ronald M. Kaplan. Lexical-Functional Grammar: A Formal System for Grammatical Representation. In *Joan Bresnan,(editor) The Mental Representation of Grammatical Relations*. MIT Press, 1982.

Chris Brew. Stochastic HPSG. In *Proceedings of the 7th EACL*, pages 83–89, 1995.

Norbert Broker, Micheal Strube, Susanne Schacht, and Udo Hahn. Coarse-Grained Parallelism in Natural Language Understanding: Parsing as Message Passing. In *Proceedings of the International Conference on New Methods in Language Processing (NeMLaP)*, pages 182–189, Manchester, UK, Sept. 1994.

Wales Browne. More on Definiteness Markers: Interrogatives in Persian. In *Linguistic Inquiry, 1.3: 59-63*, 1970.

Miriam Butt. *The Structure of Complex Predicates in Urdu*. CSLI, Stanford, 1995.

L. Cardelli and A. D. Gordon. Mobile Ambients. In *Proceedings of the workshop on Higher Order Operational Techniques in Semantics*, Stanford University, December 1997.

Bob Carpenter. *Lectures on Type-Logical Semantics*. MIT Press, Cambridge, Massachusetts, 1996.

Nicholas Carriero and David Gelernter. Linda in Context. In *Communication of the ACM, 32(4)*, pages 444–458, April 1989.

J. Carroll, E. Briscoe, and A. Sanfilippo. Parser evaluation: a survey and a new proposal. In *Proceedings of the 1st International Conference on Language Resources and Evaluation*, pages 447–454, Granada, Spain, 1998.

Hye-Won Choi. *Optimizing Structure in Context: Scrambling and Information Structure.* PhD thesis, Stanford University, August 1996.

Noam Chomsky. Degrees of Grammaticalness. In *Jerry A. Fodor and Jerrold J. Karz (eds), The Structure of Language: Readings in the Philosophy of Language*, pages 384–389. Printice Hall, 1964.

Noam Chomsky. *Barriers.* MIT Press, Cambridge, Massachusetts, 1986.

Noam Chomsky. *The Minimalist Program.* MIT Press, Cambridge, Massachusetts, 1995.

Thomas Christaller and Dieter Metzing. Parsing Interaction and a Multi-Level Parser Formalism Based on Cascaded ATNs. In *Automatic Natural Language Parsing, K. Sparck and Y. Wilks (eds)*, pages 46–60, England, UK, 1983. Ellis Horwood.

Bernard Comrie. *Language Universals and Linguistic Typology.* Basil Blackwell Publisher, Oxford, England, 1981.

Jim Cowie, Sergei Nirenburg, Siamak Rezaei, and Remi Zajac. Proposal for Persian-English Machine Translation Project. Computing Research Lab(CRL), New Mexico State University, July 1997.

Matthew W. Crocker. *Computational Psycholinguistics: An Interdisciplinary Approach to theStudy of Language.* Kluwer academic Publishers, Dordrecht, 1995.

Matthew W. Crocker and Ian Lewin. Parsing as Deduction: Rules versus Principles. In *Proceedings of the Tenth European Conference on Artificial Intelligence*, Vienna, 1992.

Matthew Walter Crocker. *A Logical Model of Competence and Performance in the Human Sentence Processor.* PhD thesis, University of Edinburgh, 1992.

Mohhammad Dabir-Mogaddam. *Syntax and Semantics of Causative Constructions in Persian.* PhD thesis, Univ of Illinois at Urbana, 1982.

Najaf Daryabandari. Zabăn-e goftări va neveştări (Spoken Language and Written Language). In *Motarjem, Vol 3, No 9*, 1993.

L.D. Erman, V. R. Lesser F. Hayes-Roth, and D. R. Reddy. The Hearsay-II Speech-Understanding System: Integrated Knowledge to Resolve Uncertainty. In *ACM Comput. Surv. 12*, pages 213–253, 1980.

Mehrdad Fahimi and Mehrnush Shamsfard. Moarefi denă: yek system dark-e matn-e fărsi (DENA: a Persian Text Understanding System). In *Proceedings of ICSCS*, Sharif Univ. of Technology, Tehran, December 1995.

Sandiway Fong. The Computation of Movement. In *ACL International Workshop on Parsing Technology (IWPT1997)*, Boston, Sep 1997.

Tsutomu Fujinami. *A Process Algebraic Approach to Computational Linguistics.* PhD thesis, Centre for Cognitive Science, University of Edinburgh, 1996.

Gerald Gazdar, Ewan Klein, Geoff Pullum, and Ivan Sag. *Generalized Phrase Structure Grammar.* Harvard University Press, Cambridge,Massachusetts, 1985.

Gerald Gazdar and Chris Mellish. *Natural Languge Processing in Prolog.* Addison Wesely, Netherlands, 1989.

A. V. Gershman. A Framework for Conceptual Analyzers. In *Strategies for Natual Language Processing, Lhnert, W. G. and Ringle, M. H. (eds.)*, 1982.

E. Gibson and Brienhier. Optimality Theory and Human Sentence Processing. MIT Working Papers in Linguistics (In press), 1998.

A. D. Gordon and L. Cardelli. Mobile ambients. In *Foundations of System Specification and Comutation Structures, Lecture Notes in Computer Science*, Springer Verlag, 1998.

Paul Gorrell. Contrasting Structural and Licensing Approaches to Parsing: the Case of Minimal Attachment, 1993.

William Gropp, Ewing Lusk, and Anthony Skjellum. *Using MPI: Portable Parallel Programming with the Message Passing Interface.* MIT Press, Cambridge,Massachusetts, 1994.

Zelal Gungordu. *Incremental Constraint-based Parsing: An Efficient Approach for Head-final Languages.* PhD thesis, Centre for Cognitive Science, University of Edinburgh, 1997.

T. Gunji. *Japanese Phrase Structure Grammar.* Dordrecht, Reidel, 1987.

L. Haegeman. *Introduction to Government and -Binding Theory.* Blackwell, Oxford, 1994.

Mark Hale and Charles Reiss. What an OT parser tells us about the initial state of the grammar. In *Proceedings of the GALA'97 conference on Language Acquisition*, pages 352–357, Edinburgh, UK, April 1997.

Margaret Marie Hashemipour. *Pronominalization and Control in Modern Persian.* PhD thesis, Univ. of California, San Diego, 1989.

John A. Hawkins. A Parsing Theory of Word Order Universals. *Linguistic Inquiry*, 21(2): 223–264, 1990.

John A. Hawkins. *Word Order and Performance.* Cambridge Univ. Press, Cambridge, 1994.

C. A. Hoare. Monitors : an Operating System Structuring Concept. Technical Report STAN-CS 73-401, Department of Computer Science, Stanford University, 1973.

Beryl Hoffman. A CCG Approach to Free Word Order Langauges. In *Proceedings of 30th Conference of the ACL, Student Session*, pages 60–66, 1992.

Beryl Hoffman. *The Computational Analysis of the Syntax and Interpretation of Free Word Order in Turkish.* PhD thesis, Dept. of Computer and Information Science, University of Pennsylvania, 1995.

X-M Huang and L. Guthrie. Parsing in Parallel. Technical Report MCCS-85-40, New Mexico State University, Las Cruces, NM, 1985.

Ajay N. Jain and Alex H. Waibel. Parsing with Connectionist Networks. In *Current Issues in Parsing Technology, M. Tomita (ed)*, pages 243–260. Kluwer, 1991.

Mark Johnson. *Attribute-Value Logic and The Theory of Grammar.* CSLI, Stanford, 1988.

Mark Johnson. Resource Sensitivity in Grammar and Processing. In *The Ninth Annual CUNY Conference on Human Sentence Processing*, New York, 1996.

Mark Johnson. Features as Resources. In *Proceedings of the LFG Conference*, University of California, San Diego, June 1997a.

Mark Johnson. A resource-Sensitive Interpretation of Lexical Functional Grammar. In *WWW (Available online)*, Brown University, June 1997b.

A.K. Joshi, K. Vijay-Shanker, and D. Weir. The Convergence of Mildly Context-Sensitive Grammatical Formalisms. In *Peter Sells, Stuart M. Shieber and Thomas Wasow (editors),*

Foundational issues in natural language processing, pages 31–81, Cambridge, Mass, 1991. MIT Press.

Aravind K. Joshi. Phrase Structure Grammar. In *Encyclopedia of Artificial Intelligence (Ed. Stuart C. Shapiro)*, pages 344–41, New York, 1987. Wiley.

Aravind K. Joshi, L.S. Levy, and M. Takahashi. Tree Adjunct Grammars. In *J. Comput. Syst. Sci., 10(1)*, pages 136–163, 1975.

T. Vahidian Kamyar. Momentary, Durative, and Momentary-Durative Verbs in Persian. In *Iranian Journal of Linguistics, 9(2)*, 1992.

Ronald M. Kaplan. A Probabilistic Approach to LFG. In *LFG Colloquium and Workshops*, Grenoble, August 1996.

Ronald M. Kaplan and John T. Maxwell. An Algorithm for Functional Uncertainty. In *Proceedings of the 12th International Conference on Computational Linguistics (COLING '88)*, 1988.

Ronald M. Kaplan and Annie Zaenen. Long-Distance Dependencies, Constituent Structure, and Functional Uncertainty. In *Mark R. Baltin; Anthony S. Kroch (eds) Alternative Conceptions of Phrase Structure*, Chicago and London, 1989. The University of Chicago Press.

Simin Karimi. *Aspects of Persian Syntax, Specificity, and the Theory of Grammar.* PhD thesis, University of Washington, 1989.

Simin Karimi. Obliqueness, Specificity, and Discourse Functions: ră in Persian. In *Linguistic Analysis, vol 20, Number 3-4*, pages 139–191, 1990.

Simin Karimi and M. Brame. A Generalization Concerning the EZAFE Constructions in Persian˙ Unpublished manuscript presented at the annual Conference of the Western Conference of Linguistics, Canada, 1986.

L. Karttunen and M. Kay. Parsing in a Free Word Order Language. In *D. Dowty, L. Karttunen and A. Zwicky (eds.), Natural Language Parsing, Psychological, Computational, and Theoretical Perspective*, pages 279–306, Cambridge University Press, Cambridge, 1985.

Lauri Karttunen. Radical Lexicalism. In *Mark Baltin and Anthony Kroch (eds.) Alternative Conceptions of Phrase Structure*, The University of Chicago Press, 1989.

Lauri Karttunen. The Proper Treatment of Optimality in Computational Phonology. In *Proceedings of International Workshop on Finite-state Methods in Natural Language Processing (FSMNLP'98)*, Bilkent University, 1998.

Michael B. Kashket. Parsing a Free-Word Order Language: Warlpiri. In *24th proceedings of the ACL*, pages 60–66, 1986.

Martin Kay. Parsing in Functional Unification Grammar. In *D. Dowty, L. Karttunen and A. Zwicky (eds.), Natural Language Parsing, Psychological, Computational, and Theoretical Perspective*, pages 251–278, Cambridge University Press, Cambridge, 1985.

Frank Keller. How do humans deal with ungrammatical input? experimental evidence and computational modelling. In *Dafydd Gibbon (ed.) Natural Language Processing and Speech Technology*, pages 27–34. Mouton de Gruyter, 1996.

Frank Keller and Theodora Alexopoulou. Phonology competes with syntax: Experimental evidence for the interaction of word order and accent placement in the realization of information structure. 1999. submitted.

P. Khanlari. *Junior Highschool Textbook (in Persin).* Iran Ministry of Education, Tehran, 1965.

Albert Kim. Graded Unification: A Framework For Interactive Processing. In *32nd Annual Conference of ACL*, 1994.

Teracy King. *Configuring Focus and Topic in Russian.* CSLI, Stanford, 1993.

Simon Kirby. *Function, Selection and Innateness: the Emergence of Language Universals.* Oxford University Press, April 1999.

George J. Klir and Tina A. Folger. *Fuzzy Sets, Incertainty, and Information.* Printice Hall, London, 1988.

Robert Kluender. Deriving Island Constraints from Principles of Predication. In *Helen Goodluck and Michael Rochemont (eds.), Island Constraints: Theory, Acquisition and processing*, pages 223–258, Kluwer, Dordrecht, 1992.

Lars Konieczny and Barbara Hemforth. Incremental Parsing with Lexicalised Grammars. In *First Analysis, Reanalysis, and Repair*, 1994.

Robert J. Kuhn. A Parlog implemetation of government-binding theory. In *Proceedings of COLING'90, Vol 3*, pages 394–396, Helsinki, Finland, 1990.

David Kuznick. A Parameterized Machine Translation System. Master's thesis, Brandeis University, 1988.

A. K. S. Lambton. *Persian Grammar.* Cambridge University Press, 1953.

Dale Miller. The π-calculus as a Theory in Linear Logic: Priliminary Results. In *Proceedings of*

Workshop on Extensions to Logic Programming, number 660 in Lecture Notes in Computer Scienc, pages 245–265, Springer Verlag, October 1992.

Robin Milner. The Polyadic π-Calculus: a Tutorial. In *F. L. Bauer, W. Brauer, and H. Schwichtenberg, editors, Logic and Algebra of Specification*, pages 203–246, Springer Verlag, 1993.

David Milward. Dynamic Dependency Grammar. In *Linguistics and Philosophy 17*, pages 561–605, 1994.

Mohammad Dabir Mogaddam. On the (in)dependence of syntax and pragmatics: Evidence from the postposition -ra in Persian. In *Cooperating with Written Texts, The Pragmatics and Comprehension of Written Texts, (editor) Dieter Stein*, Mouton de Gruyter, 1992a.

Mohammad Dabir Mogaddam. Some comments on *ra* in Persian. In *Iranian Journal of Linguistics, Vol*, 1992b.

Jan Mohammad and Simin Karimi. Light verbs are taking over: Complex verbs in Persian, 1993.

Glyn V. Morill. *Type-logical Grammar: Categorial Logics of Signs.* Kluwer academic Publishers, Dordrecht, 1994.

J. A. Moyne. The So-Called Passive in Persian. *Foundations in Language*, 12, 1974.

J. A. Moyne and G. Carden. Subject Reduplication in Persian. In *Linguistic Inquiry, Vol.5*, 1974.

Mehrangiz Nu-bahar. mofradăt yă anasor-e shenăvar dar zabăn-e fărsi (Singulars and Floating

Elements in Persian Language). In *2nd Conference of Theoretical and applied linguistics*, 1992.

Anastasios Patrikakos. Modern Greek Control Structures in HPSG. Master's thesis, University of Edinburgh, 1995.

F. Pereira and D. Warren. Parsing as Deduction. In *Proceedings of 21st Conference of the ACL*, Cambridge,Massachusetts, 1983.

D. Peterson. *Noun Phrase Specificity.* PhD thesis, The university of Michigan, 1974.

Colin Philips. *Order and Structure.* PhD thesis, MIT, 1996.

Alan Prince and Paul Smolensky. Optimality Theory: Constraint Interaction in Generative Grammar. Technical Report RuCCS Technical Report #2, Center for Cognitve Science, Rutgers University, October 1993.

Bradley L. Pritchett and John W. Reitano. Parsing with on-line Principles: a Psychologically Plausible, Object-Oriented Approach. In *Proceedings of COLING'90, Vol 3*, pages 437–439, Helsinki, Finland, 1990.

Khosrow Qolamalizade. *farăyandha-ye harekati dar zabăn fărsi (Movement Processes in Persian).* PhD thesis, Dept. of Linguistics, Tehran Univ., Tehran, 1993.

Mohsen Rais-Ghasem. pardăzesh-e zabăn-e tabiei va părdazesh-e zabăn-e fărsi (Natural Language Processing and Processing of Persian Language). Master's thesis, Sharif University of Technology, Tehran, 1991.

Owen Rambow and Aravind K. Joshi. A Processing Model for Free Word-Order Languages.

In *Perspectives On Sentence Processing, edited by Charels Clifton, Lyn Frazier and Keith Rayner*, pages 267–301, Lawrence Erlbaum Associates, New Jersey, 1994.

Mike Reape. Getting Things in Order. In *Discontinuous Constituency, Harry Bunt and Arthur van Horck (eds.)*, pages 209–253, Mouton de Gruyter, 1996.

Siamak Rezaei. tarrăhi-e system-e tajziye va tolid-e jomalăt-e zabăn- e fărsi (A Parser and Generation System for Simple Sentences of persian). Master's thesis, Islamic Azad University (South Section),Tehran, 1992.

Siamak Rezaei. Constraint-Based Parsing of a Free Word Order Language: Persian. Master's thesis, Dept. of Artificial Intelligence, University of Edinburgh, 1993.

Siamak Rezaei. Linguistic Communicating Process Structures. In *Australasian Natural Language Processing workshop (ANLPW)*, Sydney, 1997.

Siamak Rezaei. Algebraic Optimality. In *CCS annaul conference, Centre for Cognitive science, university of Edinburgh*, 1998.

Siamak Rezaei. Fuzzy Word Order Constraints. Constraints vs. Preferences Workshop, Poznan, Poland, May 1999.

Siamak Rezaei. Parsing Scrambling with Path Set: A Graded Grammaticality Approach. In *ACL International Workshop on Parsing Technology (IWPT2000)*, Trento, Itlay, Feb 2000.

Siamak Rezaei and Matthew Crocker. A Distributed Architecture for Parsing Persian. In *Proceedings of ICSCS*, Sharif Univ. of Technology, Tehran, December 1995.

Siamak Rezaei and Matthew W. Crocker. A Dynamic Representation of Grammatical Re-

lations. In *Proceedings of the LFG Conference*, University of California, San Diego, June 1997.

Dariush Riazati. Constraint-Based Parsing of a Free Word Order Language: Persian. Master's thesis, Dept. of Computer Science, RMIT, Australia, 1997.

J. R. Ross. *Constraints on Variables in Syntax.* PhD thesis, MIT, 1967.

Yadollah Samareh. *Persian Language Teaching (AZFA), Intermediate Course.* International Relations Dept, Ministry of Islamic Culture and Guidance, Tehran, 1989.

Vida Samiian. *Structure of Phrasal Categories in Persian: An X-Bar analysis.* PhD thesis, UCLA, 1983.

Mohammad A. Sanamrad and Hauya Matsumoto. PERSIS: A Natural-Language Analyzer for Persian . In *Journal of Information Processing, vol 8, No. 4*, pages 271–279, 1985.

R. C. Schank. *Conceptual Information Processing.* North Holland, Netherlands, 1975.

Vida Shaghaghi. *Barresi-ye Văje-bast dar Fărsi (An Overview of Persian Clitics).* PhD thesis, Dept. of Linguistics, Tehran Univ., Tehran, 1993.

Mohammad Javad Shariat. *dastur-e Zabăn-e fărsi (Persian Language Grammar).* Mashal press, Tehran, 1971.

N. E. Sharkey and Ronan G. Reily. *Connectionist Approaches to Natural Language Processing.* Erlbaum, 1992.

Paul Smolensky and Suzanne Stevenson. Extending optimality theory to comprehension: Competence and performance. In *Architectures and Mechanisms for Language Processing (AMLAP) Conference*, September 1997.

A. Soheili-Isfahani. *Noun Phrase Complementation in Persian.* PhD thesis, University of Illinois at Urbana, 1976.

A. Soheili-Isfahani. naqd-o moarrefi-ye ketăb-e mabăni-ye elmi-ye dastur-e zabăn-e fărsi (A Critisism of the Book: Scintific Foundations of Persian Language Grammar). In *Iranian Journal of Linguistics, 6(2)*, 1989.

Mark Steedman. Dependencies and Coordination in the Grammar of Dutch and English. In *Language, 61:523:568*, pages 60–66, 1985.

Mark Steedman. Combinatory Grammars and Parasitic Gaps. In *Natural Language and Linguistic Theory, 5*, pages 403–439, 1987.

S. Stevenson. Competition and Recency in a Hybrid Network Model of Syntactic Disambiguation. *Journal of Psycholinguistic Research*, 23(4):295–321, 1994.

Bruce B. Tesar and Paul Smolensky. Learning Optimality-Theoretic Grammars. In *A. Sorace, C. Heycock and R. Shillcock (eds.), Language Acquisition: Knowledge Representation and Processing*, North-Holland, 1999.

Henry S. Thompson. Chart Parsing for Loosely Coupled Parallel Systems. In *Current Issues in Parsing Technology, M. Tomita (ed)*, pages 231–241. Kluwer, 1991.

R. Trehan and P. F. Wilk. A parallel chart parser for the commited choice non-deterministic(CCND) logic language. Technical Report AIAI-TR-36, Artificial Intelligence Applications Institute, January 1988.

David Tugwell. A State-Transition Syntax for Data-Oriented Parsing. In *Proceedings of the 7th EACL*, pages 272–277, 1995.

Hans Uszkoreit. Constraints on Order. Technical Report 364, SRI, October 1985.

Hans Uszkoreit. Categorial Unification Grammars. In *11th International Conference on Computational Linguistics*, pages 187–194, Bonn, 1986.

Hans Uszkoreit. *Word Order and Constituent Structure in German.* CSLI, Stanford, 1987.

Hans Uszkoreit. Strategies for Adding Control Information to Declarative Grammars. Technical Report RR-91-29, DFKI, August 1991.

D. J. Weir. *Characterizing Mildly Context-Sensitive Grammar Formalisms.* PhD thesis, Deptartment of Computer and Information Science, University of Pennsylvania, 1988.

Robert Weissberg and Suzanne Buker. *Writing Up Research, Experimental Research Report Writing for Students of English.* Prentice Hall, Englewood Cliffs, New Jersey, 1991.

Gernot L. Windfuhr. *Persian Grammar, History and State of Its study.* Mouton Publishers, The Hague New York, 1979.

R. Winograd. *Understanding Natural Language.* Academic Press, New York, 1972.

W. A. Woods. Cascaded ATN Grammars. In *American Journal of Computatioanl Linguistics, 6*, pages 1–12, 1980.

R.R. Yager. On a General Class of Fuzzy Connectives. In *Fuzzy Sets and Systems,4*, 1980.

A. Yonezawa and I. Ohasawa. Object-Oriented Parallel Parsing for Context-Free Grammars. In *Proceedings of the 12th International Conference on Computational Linguistics (COLING'88)*, pages 773–778, Budapest, 1988.

James H. Yoon. A-Chain Locality: Some Cross-Linguistic Variations. In *Proceedings of the 3rd Annual Meeting, Formal Linguistics Society, Mid-America*, 1992.

Arnold M. Zwicky. Clitics and Particles. In *Language 61:283-305*, 1983.

Arnold M. Zwicky. Concatenation and Liberation. In *Papers from the 22nd Regional Meeting of the Chicago Linguistic Society*, pages 65–74, 1986.

Arnold M. Zwicky and Geofferey Pullum. Cliticization vs. Inflection: English. In *Language 59:502-513*, 1983.