Finite Paradigm Grammar:
A Computational Analysis of Icelandic Inflections

By

Gregory Karl Shenaut
B.S. (Michigan State University) 1970
M.A. (Michigan State University) 1975
C.Phil. (University of California) 1981

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Linguistics

in the

GRADUATE DIVISION

OF THE

UNIVERSITY OF CALIFORNIA, BERKELEY

Approved: 5/19/1982
Chairman Date
.... 5/19/1982
.... 5/20/1982

DOCTORAL DEGREE CONFERRED
JUNE 19, 1982
....

PREFACE

Inflectional paradigms (in languages which have them to a significant degree) are often tantalizingly regular in form. These regularities are often most apparent when the paradigms are written out in some format which draws the eye to regions of similarity, such as the traditional two dimensional declension or paradigm table. The visual method of communicating structural regularity has often been used in modern linguistics, by such writers as Pike, Haugen, Bierwisch, Anttila, Woods, Johnson & Postal, and Lakoff, although not all of these have been concerned with the inflectional paradigm. There can be a certain conceptual elegance or beauty in a inflectional system which is all-too-easily lost when the focus shifts from morphology as a system to morphology which is diffused over some larger system of which it is a part. Most accounts of inflectional morphology attempt to condition inflectional processes on (morpho-)syntactic or semantic grounds, and as a result much of the regularity in the system is obscured, to my way of thinking. I feel that the reason for this is that the traditional concept of paradigm, which is based primarily on formal (surface) relations among the different forms and only secondarily on their semantic/syntactic functions, has not been adequately explored.

One of the nice things about computational or linguistic network models is that they are readily transformed into a visual structure which can communicate to the human viewer those relations which the inventor of the model intends to represent. At the same time, the computer model maker can also implement a machine whose behavior is predicted by the model. The performance of this machine can be used to further communicate the intentions of the model builder, but it can also be used to perform arbitrary, (possibly) useful tasks.

The aim of this dissertation is to describe a computational model for grammatical paradigms, called Finite Paradigm Grammar (FPG). The model is applied to the inflectional system of Modern Icelandic (MI).

Chapter One contains an historical orientation and general introduction to the work. Chapter Two combines a description of FPG syntax and semantics with a discussion of the theoretical motivation (or arbitrariness) of the major design decisions. Chapter Three is an FPG of MI noun and adjective inflections. Chapter Four discusses the issue of psychological reality with respect to FPG.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

This section is an informal introduction to the motivations underlying the development of Finite Paradigm Grammar (FPG). If there is perhaps excessive use of the first person, this is because many of the decisions were personal ones, having to do with aesthetic preference or styles of problem-solving rather than being motivated by work in ′the literature′ or ′the field′. This usage will decrease as the text becomes more substantive.

Throughout this chapter and the next one, Icelandic examples are chosen out of context in order to illustrate certain points. In later sections the same examples will come up again, but as parts of a larger system. This will in some cases result in a reanalysis. Although this is potentially confusing, I felt that the introductory and explanatory sections would be made much more clear if I didn′t try to be exhaustive in the analysis of the preliminary examples.

1.1. Linguistic theories should be more problem-oriented. When I first became a student in linguistics (in 1972), the topic which most attracted my interest was what I saw as the inevitable transition from Aspects-style syntax (e.g. Chomsky 1965, 1971) to generative semantics (e.g. Lakoff 1971, Ross 1970, McCawley 1970, Postal 1969). There was something about the idea of transforming a representation of the logical meaning of a sentence directly into the surface form of the sentence which interested me very much. Although I was fairly convinced of the general correctness of this program, I was also concerned about the issue of verification of linguistic hypotheses (e.g. Labov 1972, Spencer 1973). As the boundaries of what was considered generative semantics widened, with progressively more abstract evidence for particular points, I became more and more concerned, to the point of exploring various possibilities for verification of hypotheses about nondiscrete semantic and pragmatic variables (Shenaut 1975b, 1975a). What I learned from this was primarily that theoretical linguists are not impressed by this kind of evidence.

At about this time computational linguistics was a growing force, perhaps at its strongest during the 1975 summer MSSB workshop at Berkeley. The computational approach seemed to correct some of the problems I felt were inherent in the contemporary theories coming from within linguistics. The reason for this was that the goal of the computational linguists at least ultimately was to actually create a machine which could deal with some particular aspect of natural language in an 'intelligent' way. It seemed to me that this was superior to the goals of linguists at the time (to figure out a universal explanatory theory of language): if such a machine were built, and it did perform in a human-like way, then developing the underlying theory would have been time well spent to the extent that the machine worked. Furthermore, even if the theory turned out to be nonoptimal for any reason, the problems solved by the original program (e.g. data-base access via a questioning-answering system) would still be solved. There were several systems developed which had some success (Petrick 1965, Woods 1973, Winograd 1972, Kay 1973, Kaplan 1973) but again there seemed to be an inherent difficulty. As Rosenberg (1975) points out, the problem this time was in some sense the converse of the problem with generative semantics: while the domain of generative semantics was growing and growing in an attempt to explain more and more aspects of language behavior, computational linguistics was shrinking and shrinking trying to limit the scope of its projects so that programs could be written, 'trivializing the complexities of language understanding'.

The MSSB workshop was a serious effort to get computational linguists and noncomputational linguists to talk to each other, and it seemed at the time that this was succeeding very well. Results include Lakoff (1977) and Bobrow and Winograd (1976). It was during this period when computer specialists and linguists were actively trying to utilize each other's theories that I came to Berkeley. My first research at the university was in this area, in Professor M. O'Malley's lab. Among other things, I became convinced that a real need in linguistics is to develop a computer oriented 'applied linguistics' where linguistic theories would be immediately applicable to problem solving in the field of natural language computation.

A few years later, when linguists' interest in natural language computation, which had been fueled by the then-defunct ARPA speech understanding project, was diminishing, I took part in a field methods course given by Professor C. Fillmore. The language was Finnish. At that time, I had written an ATN interpreter for a class in Artificial

Intelligence. It had been used for various fragments of English syntax and morphology by members of the class. I decided to apply the ATN approach to Finnish inflections. For reasons which I set out in more detail below, the ATN formalism turned out to be not very well suited to this application. I began to consider other ways of solving the problems of inflectional morphology on a computer.

1.2. Using an Icelandic Dictionary. My first experience with Icelandic was in a reading course in Old Norse taught by Professor C. Clover. During this course, I learned that for me at least one of the most difficult matters was looking up words in the Icelandic dictionary. This is true because of the complex morphophonological processes triggered by the inflections. The form of a word in some inflectional category might be drastically changed, in many cases without an overt marker in the surface form to indicate which processes need to be undone to derive the citation form. There are many types of neutralization which take place, obscuring the derivational paths taken by a particular form.

The next experience with Icelandic was in a course in conversational Icelandic taught by Mrs. D. Thordarson, at Stanford. I soon realized that the converse situation was also true: even if you succeed in looking up a word in the dictionary, more information than is generally given is required to derive the form actually needed in conversation or writing. This is because there are many different inflection groups (i.e. declensions and conjugations), and it is difficult in most cases and impossible in many cases to assign a given form (even a citation form) to its group on the basis of surface form alone.

The import of this section is: In any theoretical or practical system which includes an adequate representation of Icelandic inflections more information is needed about inflectional categories in the lexicon than is present in any existing system.

1.3. Lemmatizing Old Norse Sagas. There exists an area of research where such considerations are relevant. In the field of Scandinavian literature, some researchers are involved in making head-word concordances of the ancient sagas and eddic poems. An obvious application of computer technology is to read in some text, strip endings from each word, look up the result in a dictionary, and thereby come up with the head word for the input word. This process is called lemmatization. Gilbert and Hirschmann (1981) report that computer-aided concordances and indexes are often held up in publication because of delays in lemmatization, and comments that unless they are lemmatized they are sometimes less than useful. The lack of a general approach to lemmatization might be responsible for the relative paucity of published lexical aids of this type. There have been several such ending-stripping programs written, for various natural languages (Boot 1980, Hann 1974, Hellberg 1972, Sågvall 1975, Eggers 1981, Dawson 1974).

These ending strippers typically choose not to deal with other morphological processes, such as prefixing or root-vowel changes. For example, Eggers (1981) states that even while it would be possible to generate headwords from stem-forms, as a practical matter it is better to expand the dictionary to include all stem forms. Of course, generally speaking this is correct: if the concern is only to write a program which will recognize the headword given an inflected form in one language, it will probably always be much faster-running if the number of stem-variants to be checked is reduced.

In Winograd's (1972) dissertation, which involved a program that performed simple tasks in an imaginary world given English input, there was a simple ending stripper for English inflections. This program, printed in flowchart form in Winograd (1972), uses primitives such as 'cutoff' and 'add' to remove and add suffixes, and 'try' to look up words in the dictionary. It uses categories such as LIQUID and VOWEL, and can follow arcs conditionally depending on membership in the categories. An attempt is made to generalize similarities in 'morphographemic' structure. All analyses are checked in the dictionary and must be found there; stripped endings are checked only to determine whether they apply to the word class of the root (e.g. -est only can go on an ADJ).

Another approach is in Geens (1979). The main program produced relatively nonabstract analyses of English text. The technique for undoing inflections involved a list of endings arranged in a hierarchy. The actual endings cause marking of position on a stack, and a terminal symbol causes a return to the previous position. Associated with each ending is a code indicating its possible functions. Unlike the Winograd approach, there is no 'try' primitive--no dictionary is required. On the other hand, no attempt is made to group similarities among endings.

There is a project involving automatic lemmatization of Modern Icelandic going on at the University of Iceland (Randa Mulford, pers. communication) however I don't know what approach is being used there.

To summarize, lemmatization programs typically work from a surface form to a head-word, with or without a dictionary. As far as I know, there is no attempt to recognize the inflectional category of the input form, and no need to produce other related forms. For these reasons, it seems to me that systems oriented toward recognizing head-words might provide useful insight about certain aspects of inflectional morphology, but for theoretical purposes or for relatively distant applications, something more general is needed.

The existence of a program to lemmatize Old Norse or modern Icelandic text would certainly be useful for those directly concerned with concordance construction and lexical retrieval. It would also serve, by definition, as a tool for students of Icelandic who needed help in finding the correct headword for their dictionary lookups. In addition, if it were put together with emphasis on generality, then it could also be used to state theoretical hypotheses and test them. Therefore, one of the requirements of this project is to invent a theory of inflections which can be used with little or no modification, to lemmatize text.

1.4. A Brief Overview of ATN-type Grammars. The ATN formalism (Woods 1970) is the best known of a family of computer languages developed to parse sentences in natural languages (e.g. Kay 1973, Kaplan 1973, Johnson 1976, Marcus 1980). For our purposes, the similarities among these languages are more important than their differences, and we will briefly describe the ATN system as an example of them all.

In an ATN grammar, there is a set of nodes connected by arcs in a network. During parsing, there is an input pointer which points to a particular word in the sentence. Each arc has conditions on it which must be met before the arc can be traversed. Often traversal is conditional on the category of the current word pointed to by the input pointer. Each node corresponds to a state in the machine, and the successful traversal of an arc causes the machine to change state. Most changes of state causes the input pointer to advance. The recognition process begins by entering the start state with the input pointer at the first element. The process terminates successfully when the input pointer is on the last element and the machine is at a terminal state. When a state is left, alternate exit paths may exist. If so, then these paths are stored on a stack, because if the machine reaches a state where no transition is possible, it will back up to the most recently stored state and take the next alternate arc.

It is possible for there to be arbitrary actions which take place as a result of traversing an arc. Some of these actions result in modifications to the hierarchical tree structure which is built up as the parse continues. This tree structure usually corresponds to the deep structure tree in transformational grammar, but could be any arbitrary structure, including procedures implementing semantic-like actions (Woods 1968). Some of the actions effect the tree similarly to the action of transformations. Other actions cause data such as pieces of trees to be stored in registers where they can be recalled later and tested in conditions or added to the tree structure.

It is possible for such networks to be recursive. This is implemented in terms of a condition or action on an arc which recursively calls a node that is either in the current network or another one. We will consider a simple example (from Winston 1981). (By the way, this example is what Woods refers to as a recursive transition network (RTN) since it is not augmented). It represents what is referred to as a "simple noun group":

```
                                adjective
                                /------)
                                |      )
------  determiner            ------  /
| S1 |----------------->| S2 |<-/
------                        ------
                               |      noun         ------
                               |----------------->| S3 |
                                                   ------
```

This network will accept sequences like "the dull book" or "a big red ant", but not "dull books" or "an ant". Supposing that this is acceptable, then a sentence recognizer could be set up like this:

```
------      [noun group]       ------     verb     ------
| S1 |--------------------->| S2 |----------->| S3 |
------                         ------              ------
```

This would handle "A big red ant died", but not "I hate dull books".

From the point of view of syntax, one of the primary problems with ATN grammars is that it is difficult to introduce heuristics into the interpreter. This means that the usual path through the network (serious ATN networks are quite complex) is very inefficient. For a trivial example, consider what the simple sentence parser above would do with the sentence "A famous red died". Since "red" is either a noun or an adjective, and since either a noun or an adjective is legal at state S2 in the noun group network, it might happen that the initial attempt will try to make "red" an adjective, fail to find a following noun, and back up. In a real parser, this kind of failure could add hours to the parse-time for a complicated sentence. While there have been attempts to alleviate this situation (Bates 1975), there is no universally accepted solution.

An interesting possibility was suggested by Kay (1975) with regard to an extension of this type of grammar model. Since the network formalism allows free movement around the grammatical space of a language, and since there are no a priori limits on the tests and actions permitted on the arcs, it might be possible to create a network grammar which can be interpreted by both parsers and language production programs. This is a very interesting possibility, but as far as I know it has not been realized for syntax. Lakoff and Thompson 1975 developed a related approach for their "cognitive grammar". In this system, each rule had certain conditions and processes associated with it. Some of these were for input (recognition) and others for output (production). However, I am not aware of any implementation of a cognitive grammar system on a computer.

Another variant of the ATN approach is the General Language Processor (Johnson 1976). In GLP, instead of a constant grammar network being traversed for all sentences, the lexicon contains procedural information for all words in the language. When a sentence is parsed, only the process information in the dictionary for each word in the sentence is executed in parallel on the sentence. Each parallel process radiates outward (left and right) from the word whose lexical entry started it. The processes interact by sending information to one another; no one process ´controls´ any other process directly.

1.5. ATN Grammars and Morphology. Remember that as one traverses an arc in the ATN network the input pointer advances to the next element. This normally works well for languages with strict word order such as English. It is much less useful when the word order is free. In this case, it is necessary to try all of the possible orderings, which is in some cases a significant subset of all permutations of the elements of the sentence. This causes a significant increase both in processing time for particular sentences and also in the grammar of the language.

A similar situation was reported in Bates (1975). Her parser was the syntactic portion of the BBN system, developed under the ARPA speech understanding project. Its input was the output of the phonetic segmentation portion of the system. Because the phonetic subsystem was very ambiguous, there were significant portions of the input string which could essentially be anything. For this reason, she located portions of the input which the phonetics unit had given relatively high levels of confidence, and began her parsing from these "islands of reliability". As a result, the parsing was not strictly left-to-right.

In morphology, a similar situation is the rule. In the first place, an ending stripper will operate from right-to-left. In languages with both prefixes and suffixes, the order of parsing becomes very complex. This is true even more so because it seems to be a generalization about language that the scope of the outer portions of words tends to include all of the relatively inner portions of the word. Thus words like luckiest would tend to be bracketed "(((luck)i)est)". This can be illustrated by the replacement of -est by -er, but no possibility of replacing the y/i adjectivizer. This tendency would best be represented in an ATN system by nesting. There could be an adjective subnetwork, including -ing, -ed, -y, and other forms. This network would be nested both in the adverb network (which could include -(i)ly, etc.) and in itself to derive lucky, luckily, luckier, and luckiest.

This seems quite clear until you get right down to the business of implementing the grammar. Since the parser proceeds from left-to-right (or trivially from right-to-left), it is necessary to divide the word into subelements and parse them. You get "l-u-c-k-i-e-s-t" or "t-s-e-i-k-c-u-l". Suppose you get to a place where an adjective is needed. If you go left-to-right, you have to try for every possible adjective prefix including in-, un-, re- and so forth. If you find one, you have to keep trying for more (for cases like "unredoable"). When you can't find more prefixes, you have to try for roots. This essentially means, in the most direct approach, trying every possible adjective root in the language. Remember, this will involve seeking, for example, adjectivized nouns and verbs. When one is found, then suffixes must be examined. In the right-to-left model, a more efficient parser for English would result, but there would still be problems, and for many other languages, this would be detrimental. Obviously, both for languages with free word order and for morphology, the ATN formalism is far from optimal.

By the way, apparently there exist languages such as Turkish which have an inflectional system that is very syntax-like in that only suffixes are used, the order of suffixes is rigid, and the morphophonology is relatively straightforward. In such languages, the ATN formalism might hold its own. But there are other languages, such as Icelandic and Hebrew, where much of the inflectional system depends on whole-word changes such as vowel mutation or replacement, with concomitant changes in other parts of the word. There seems to be no clearcut way to apply the ATN-like system to such languages. Processing inflections simply cannot be done efficiently by scanning through words from one end to the other.

Johnson's GLP system, discussed in the preceding section, is not suitable for morphology either, because it depends on the individual elements of the input string being segmented and looked up in a dictionary. This is putting the cart before the horse, because in morphology what we are often attempting to do is to recognize the elements. The problem is compounded by the fact the concept of 'element', while shaky even in syntax (e.g. clitics, idioms), is useless in morphology due to various phenomena such as suppletion, fusion, discontinuous elements, and suprasegmentals.

1.6. Paradigms. The inflectional paradigm is certainly one of the most obvious ways to organize the various forms which an inflected word can take. Classical grammars make use of tables listing all of the forms of some exemplary word; the forms of words which patterned in a similar way were to be deduced from the examples. Some classical grammarians went a step further and discussed the sorts of processes involved in deriving each form in a paradigm from the root form (Matthews 1972). It is possible to organize other sets of linguistic data into arrays similar to inflectional paradigms; there seems to be a tendency among linguists to favor this kind of grouping, although usually only during the process of analysis rather than in ´official´ reports (Pike 1963).

Lotz (1978) provides a useful discussion of certain properties of linguistic paradigms. His usage extends beyond inflections: he includes among his examples the stops of English, verb conjugation, interrogative sentences and verbs referring to locomotion. He lists five kinds of abnormalities in paradigms: 1) abundance (e.g. brothers vs. brethren); 2) variance (e.g. going vs. goin´); 3) syncretism (e.g. English -s genitive and plural); 4) defectiveness (e.g. isn´t aren´t weren´t etc., but am not); 5) suppletion (e.g. happy happier happiest but good better best).

Among the reasons given for replacing paradigms with other types of analyses are those of explicitness and simplicity (Matthews 1974). The classical grammar is filled with exemplary paradigms, possibly with some compendium of inflectional processes usually given in the same paradigm form. After the example is given, there is often a list of words which ´go like´ the example, often with slight variations noted for some. For example, Einarsson (1945) gives the following information as part of his description of Icelandic strong masculine noun declension:

Sg.	nom.	hest-ur	horse	hatt-ur	hat
	acc.	hest		hatt	
	dat.	hest-i		hatt-i	
	gen.	hest-s		hatt-s	
Pl.	nom.	hest-ar		hatt-ar	
	acc.	hest-a		hatt-a	
	dat.	hest-um		hött-um	
	gen.	hest-a		hatt-a	

This is followed by: "Hestur is the most frequent of these types. Thus go words in -ingur, -ungur, -dómur, -leikur (dative singular -i missing), and -undur. Hattur (words with a in the root syllable) is a fairly common type...The dative plural of words with a as a root vowel shows u-shift...: hattur : höttum."

In this description an attempt is made to include several kinds of information. First, all of the forms for the examples are given, with hyphens inserted to separate the suffixes. Notice that this is very redundant; at first glance, the two examples seem identical except for dative plural. Next, heuristic information is given which could be used to identify a word as ´going like´ one or the other of the examples, based on a final sequence of segments. This information is normally not given explicitly in a generative approach, although perhaps it could be deduced from an analysis of the interplay of lexical features and rules used in inflectional marking. Finally, some indication is made concerning variations in the paradigms and an explanation is given for the difference in the dative plural between hattur and hestur. It is not clear, however, why the paradigm for hattur is given completely, but no example is provided of a word ending in -leikur with the missing -i in dative singular (e.g. dansleikur dance), although this seems to be as unique a situation as hattur. We will refer to these paradigms in the next sections as well.

1.7. Relations among and within paradigms. One of the objections to the use of paradigms in linguistic theory is due to the fact that there is no accepted way to make generalizations about similarities in the inflectional system.

1.7.1. Syncretism. The most obvious kind of simplification which could be made in inflectional systems, but which often cannot be easily made in a classical paradigm analysis is syncretism, or the falling together of several categories into a single form. In the Icelandic examples given above, notice the formal identity between the forms for accusative and genitive plural. Despite the syncretism in these examples, it is not the case in certain other paradigms in the language (e.g. smiði accusative plural, but smiða genitive plural ′smith′). Therefore, no attempt is normally made to reduce the paradigms. The situation is more extreme in cases where more syncretism is found. For example, in English, we often find a six-slot paradigm for the present tense of verbs, even though no verb in the language needs more than three, and the overwhelming majority need only two. There are several possible reasons why this is true (here I am only speculating): First, five of the six contrasts are found in the pronoun paradigms (I/me, you/you, he/him, we/us, they/them). The perception is that the missing you+PLURAL is an accident. Second, the holes which do exist are located in awkward places: in the middle of the matrix, or in some asymmetric set of locations. It is probably felt to be unaesthetic to contrast third person singular with all other forms. Sometimes we see a four-slot paradigm, with three singular but only one plural: this emphasizes the importance of symmetry and aesthetics in paradigm construction ([am vs. are vs. is] vs. are is favored over am vs. are vs. is). A third reason is that grammarians of English have often been trained in Latin. Finally, semantic considerations, especially if taken in historical context, can motivate the shape chosen to represent paradigms.

Some authors of paradigm grammars will remark that certain categories are not formally distinct, but the paradigms contain repeated instances of the syncretic forms. The ideal model for inflectional morphology should contain some way of explicitly marking cases where several categories fall together, and this marking should result in a simpler analysis.

1.7.2. Proportions. It is possible to find cases where there are similarities in paradigms which can not be accounted for by simply collapsing categories as for syncretism. These are situations where the formal similarity is between different paradigms or where the formal similarity stops short of identity. Notice in the paradigms for hestur and hattur given above that all forms are identical except dative plural. It should be possible to state that "hattur ´goes like´ hestur except for dative plural", and then give the dative plural. The statement would have to be made made in terms of a set of explicit operators on paradigms. It would have to be unambiguous with respect to the exact forms included in the paradigm. If this were possible, the resulting analysis could be simplified considerably.

A similar situation obtains in the comparison of adjectives in Icelandic. The inflections of the superlative and comparative forms are nearly identical to those of the positive. This is a case where the similarity is within a single paradigm.

One way of expressing these relations is in terms of proportions. For example, the nominative singular of hestur is to the genitive plural of hestur as the nominative singular of hattur is to the genitive plural of hattur. This is a comparison involving two different paradigms. More abstract proportions can be stated for adjectives; for example, the strong masculine accusative singular positive (e.g. háum from hár "high") is to the strong masculine dative singular positive (háum) as the strong masculine accusative singular superlative (hæstan) is to the strong masculine dative singular superlative (hæstan). This is a comparison within a single paradigm. These proportions have been discussed in the literature, usually with respect to the issue of the status of analogy in grammars and language change (see Anttila 1977 for a discussion) and language acquisition (e.g. MacWhinney 1978). We will return to them at several points below.

It should also be possible to generalize about partial similarities in categories. For example, notice that for hestur the nominative plural form hestar could be seen as being formed by adding -r to the accusative-genitive plural form hesta. This type of similarity is captured much more naturally by a generative approach where inflectional processes act similarly to phonological rules (cf. Anderson 1977). Thus if two rules existed in the grammar, one to suffix -a and the other to suffix -r, then the two forms could be derived by ordering the two rules in the order given and enabling application of -r suffixation only for nominative plural. A straight paradigm representation would be unable to make this kind of generalization, as far as I can see. It seems to me that ability to capture this kind of generalization is another prerequisite for a reasonable approach to inflectional morphology.

There are certain situations in inflectional paradigms where quasi-recursive processes occur. In these situations, an inflectional suffix is added which can then take other inflectional endings, including in some cases the suffix itself. Examples are Icelandic adjective comparison, where normal case-gender-number inflections are added to the superlative suffix, and causatives in Turkish or Hungarian, where double or even triple sequences of the causative suffix can be appended before other inflectional suffixes. In all such cases which I know of (but I don't know of very many) a closer look makes the recursive analysis unattractive.

First, the maximum depth of recursion is usually two or three, which means that not very much simplicity is given up by a nonrecursive analysis. Second, there is usually some phonological variation in the suffixes which marks a suffix as 'first instance' or 'second instance'. That is, the situation doesn't look like true, unbounded recursion is an accurate representation of the facts. Instead, some sort of bounded or limited recursiveness is needed.

In Cuban (and possibly other dialects of) Spanish, it is possible to add the superlative marker quasi-recursively to adjectives, up to around three times:

rico "delicious"
riquísimo "most delicious"
riquitísimo "most very delicious"
riquisitísimo "very most very delicious"
?*riquitisitísimo

These forms are generally used to express intensification rather than logical superlative degree. They all receive the normal adjectival gender-number suffixes. Notice how the form of the suffix varies as it is added more than once: -ísim- to -it- to -is-. Clearly there is a great similarity here between the endings, but it is not at all clear that the complications resulting from allowing true recursive application of the suffix are worth the advantage which would be gained, especially since each application has a unique result (there is no phonological process in Cuban Spanish which could change *riquísimísimísimo into riquisitisímo) and the depth of recursion would have to be limited to three. Instead, some means of representing the inflectional similarities among the various degrees of ′superlative′ should be incorporated into the paradigm, and each form should be present as a unique category. This will sometimes be referred to below as bounded or limited recursion.

1.8. Operations on Paradigms. The preceding observations about similarities in paradigms lead to the related question of manipulating paradigms as data objects in the grammar. A familiar way to formally represent similarities or other relations is in the form of a set of transformations which change one data object into another. The question might be asked, "what operations are needed to formally relate related paradigms"?

1.8.1. Subparadigms. The most primitive facility which seems to be needed is that of partitioning a larger paradigm into smaller pieces. There needs to be some way of specifying what portions of the paradigm are to be retained, and the representation of the paradigm must be flexible enough so that arbitrary pieces can be chopped out without altering any of the relations in them. There are several ways of specifying subparadigms which come to mind.

First, remembering back to the paradigms for hestur and hattur, we suggested above that a relevant generalization might be that "hattur ´goes like´ hestur except in dative plural". To capture this generalization, there must be a facility to eliminate certain categories from a paradigm while retaining all the rest. It is also easy to think of cases where it is more natural to specify the categories which are to be retained, deleting all the others.

Second, to implement the bounded recursion we described for adjective comparison, it is necessary to copy a piece of a paradigm such that all members whose categories include some particular set of values (e.g. all ´positive´ categories) are included, and all others excluded.

Furthermore, it is also necessary in this case to map certain category values to other ones in the copy (for example, 'positive' goes to 'superlative').

A third factor derives from the fact that some paradigms have several possible forms for the same category. For example, in English the noun brother has two plurals: brothers and brethren. Many other cases of this kind of ambiguity can be easily found. Therefore it is not sufficient to specify subparadigms solely in terms of categories. There must be some more abstract way to refer to parts of a paradigm.

1.8.2. Combining Subparadigms. Once the facility for extracting pieces of paradigms has been developed, then it is possible to construct new paradigms out of pieces of existing ones. When paradigm segments are combined, the problem is to specify the way in which the pieces are to be joined. In the classical matrix representation for paradigms, this problem does not exist, because the placement of subparadigms is determined completely by the categories found in it. But as mentioned above, the matrix representation has the disadvantage of often being grossly redundant. We concluded that it would be advantageous to structure the information in the paradigm in some way which would emphasize the formal similarities among members. A reasonable approach to this might be to derive members using inflectional rules (cf. Matthews 1972, Anderson 1981).

If this is done, the problem of how to join together subparadigms becomes more complex. This is because it is necessary to specify in some way similarities in derivation for the members of the to-be-joined subparadigms. For example, in the Icelandic adjective example alluded to above, the marker of the superlative degree is a suffixed -ast. The endings used in the positive degree are then suffixed to this. It is necessary that there be some way of indicating the ordering relation in the derivation; otherwise the superlative marker might end up being added to some positive ending.

To sum up, several kinds of reference to portions of paradigms are needed in order to capture generalizations. First, it must be possible to refer to paradigm categories. This is needed in order to specify subparadigms. It is also necessary to be able to refer to elements of category labels. There must be some way to refer to locations in paradigms independently, without referring to categories. Finally, there must be some way to refer to stages in the derivation of paradigm members.

1.9. Preview of Coming Attractions. The following chapters describe a formal approach to the description of inflectional paradigms. The approach emphasizes surface similarities among the members of the paradigm, and downplays considerations of how the paradigm and its members relate to other domains of grammar. The working hypothesis underlying this is that the paradigm itself is a linguistically significant entity.

The paradigm is seen as an abstract network of derivations, rather than a mere list of forms. The networks are arranged in such a way that all members of a paradigm are at the same level, in that there is no single underlying or abstract form. Instead, the dictionary entry for each group of forms related by the paradigm refers to the paradigm itself. There may be one or more surface forms in the dictionary as well, used as points of reference into the paradigm network.

A linear representation for the nonlinear structure implied by this approach is developed. This notational scheme is set up in such a way that similarities among paradigms are easily represented. The notation for writing derivational rules is such that all rules may apply in either direction. This naturally implies that there may be ambiguous derivations.

The approach is applied in a synchronic description of Modern Icelandic nominal and adjectival inflections. This reasonably thorough analysis provides a useful testing ground for the theory, the linear notation, and the rule language. Many issues relevant to traditional phonological analysis are raised during the analysis.

In the final chapter, the question of psychological reality is considered. Evidence from two sources is used to support the notion that inflectional paradigms of the type described here are psychologically real. The first source is an experimental study carried out based on reaction time in a lexical decision task using English strong verb inflections. The second source is a consideration of the role of the paradigm in language change. In particular, the concept of analogy as it has been applied to changes in inflectional systems is examined.

CHAPTER 2

FINITE PARADIGM GRAMMAR

2.1. Introductory remarks. Finite paradigm grammar (FPG) is a computational model for representing and manipulating linguistic paradigms. In this dissertation, it is applied exclusively to problems in inflectional morphology. Therefore, for reasons, of economy, ′paradigm′ will mean, unless otherwise noted, ′inflectional paradigm′.

The model is not closely related to work in the areas of morphology and phonology in the generative framework (Bierwisch 1967, Chomsky and Halle 1968, Kiefer 1973, Aronoff 1976, Griggs and Rulon 1974, Anderson 1981, Siegel 1979, Allen 1978). It bears somewhat more resemblance to some of the work of Matthews (1972, 1974) but there are certain fundamental differences there, as well. There are superficial similarities to computational network grammars (Woods 1970, Winograd 1972, Bates 1975), but in fact these amount to stylistic preferences. Certain phonologists have, in exploring the terrain of morphophonology, arrived at many conclusions which are quite close to FPG in some respects. The similarities have to do with such factors as separating morphology from phonology and the rest of the grammar (Wilbur 1974); insisting on the priority of the surface form (Hooper 1974, Vennemann 1974); and consideration of the issue of applying rules backwards (Leben and Robinson 1977, Vennemann 1972).

Some of the elements missing from FPG with respect to the more usual treatments are phonological features, underlying forms, unidirectional derivations, and recursion. (None of these are explicitly excluded, just absent in the current formulation of the model.)

In brief, an FPG is a set of named rules, a set of named paradigms, and a lexicon. The lexicon contains (possibly among other things) citation forms and the names of paradigms to which the entry pertains. Linguistic forms are represented here as linear strings. (There is no inherent restriction to linear strings, however). Rules specify processes which apply to linguistic forms, resulting in new linguistic forms. The rule syntax is such that all rules can be undone, recovering the initial input (possibly ambiguously). Paradigms consist of a set of nodes connected by arcs in a network. Each arc refers to a rule; traversing the arc implies applying the rule to the current form, deriving new forms. Since rules are bidirectional, arcs may be traversed in either direction. Nodes may have names and/or categories. The categories consist of sets of contrast-grades, such as (nominative singular), and the categories may be used to refer to particular nodes or groups of nodes. Names are used to refer to nodes without using their categories. Nodes which are without categories are non-terminal; they do not correspond to a form of the paradigm. It is possible for there to be nodes with neither names nor categories. Finally, there is a mechanism for stating generalizations about similarities between portions of paradigms, such as "the second declension is like the first in the plural" and "nominative plural is formed by adding -r to the genitive plural".

Using this model, it is possible to derive any member of a paradigm from any other member. Therefore, the notion ´underlying form´ (or ´leading form´ or ´root´) is not necessary. As a result, it is possible to view FPG as either a recognition model or a production model (this possibility is discussed in Kay 1975). Any of the paradigm members can be used as the citation form: this has a mnemonic function as well as a theoretical one. That is, there is a dictionary associated with the grammar in which each entry contains a reference to the paradigm(s) to which it may belong, together with one or more surface forms. These surface forms can be used as known starting points to generate other forms, and can be used as a check as to whether the recognition process correctly recognized a form associated with the entry.

When the paradigm is known, a processor can use an FPG to generate all of the forms of a dictionary entry. It would do this by simply following all of the arcs in the paradigm, performing the operations indicated along the arcs, and whenever a terminal node is reached, the current form and the category of the node would be stored or output. When this process terminates, all of the members of the paradigm will have been generated. If only a form but neither its category or paradigm are known, the combined network representing all of the paradigms in the FPG must be used to determine them. If the rule on an arc fails to apply, then the arc cannot be traversed. Again, the form, category, and this time the paradigm associated with any terminal node reached is stored, and at the end, the citation forms associated with any successful paradigms (all members were generated) can be checked in the lexicon. Citation forms found in the lexicon would then be checked to see if they belong to the paradigm used to find them, and if they match, then the system will have successfully recognized the input form.

In addition to being a formal model, FPG is also a computer programming language, embedded in the Artificial Intelligence (AI) language LISP (McCarthy, Abrahams, Edwards, Hart, and Levin 1965, Weissman 1967, Allen 1978, Winston and Horn 1981). Using the primitive functions of the FPG implementation, linguistic paradigms can be manipulated to perform arbitrary functions such as contrasting paradigms, combining paradigms, listing all forms in a paradigm for a particular word, recognizing the inflectional category of a particular form, and performing lemmatization (classifying input forms by head-word, as for concordance generation). In addition, implementing the model on computer allows the various analyses to be more easily tested for consistency and correctness (cf. Friedman 1971).

The rest of this chapter combines a description of FPG syntax and semantics with a discussion of the theoretical motivation (or arbitrariness) of the major design decisions.

2.2. FPG Networks. The most direct way to explicate the basic concepts of FPG is by way of example. Table 2.2.1 sets out the paradigm for the Icelandic nouns hestur ′horse′, hlutur ′thing′, and smiður ′smith′:

(2.2.1)	Noun class:		1	2	3
	Sg.	Nom.	hestur	smiður	hlutur
		Acc.	hest	smið	hlut
		Dat.	hesti	smið	hlut
		Gen.	hests	smiðs	hlutar
	Pl.	Nom.	hestar	smiðir	hlutir
		Acc.	hesta	smiði	hluti
		Dat.	hestum	smiðum	hlutum
		Gen.	hesta	smiða	hluta

The classification of the nouns is that of Einarsson (1945); these nouns are fairly typical members of their classes, except they have been selected because they are almost free from complex sound changes.

Generally speaking, the forms in Table 2.2.1 show 6 processes of change, all suffixations. The suffixed material for the 6 processes are *ur* *i* *a* *um* *s* *r*. Since suffixation is such a common process for inflectional systems, we will adopt the convention that a suffixation rule is named by some approximation to the suffixed material. Furthermore, we will adopt the convention that the name of a rule must begin with either a ´+´ or a ´-´. Rules in ´-´ undo rules in ´+´ and vice-versa. Thus there are 12 rule names required to specify the relations in Table 2.2.1, which we will call +ur/-ur, +i/-i, +a/-a, +um/-um, +s/-s, and +r/-r. (Note: +ur and +um are not analyzed into +u +r +m for reasons which will become clear in Chapter Three). These names have only mnemonic value: the actual processes associated with the rules must be specified explicitly.

Figure 2.2.2 shows FPG networks for the words in Table 2.2.1:

(2.2.2)

```
                      +um
Class 1.             -----> DP
                    /
                   /   +s
                  |     ----> GS
                  |   /
           +ur    |  / +a           +r
   NS <---- AS ----> AP,GP ----> NP
                   (
                    (  +i
                     ----> DS

                      +um
Class 2.             -----> DP
                    /
                   /   +s
                  |     ----> GS
                  |   /
           +ur    |  /     +a
   NS <---- AS,DS ----> GP
                   (
                    (  +i         +r
                     ----> AP ----> NP

                      +um
Class 3.             -----> DP
                    /
           +ur     /   +a         +r
   NS <---- AS,DS ----> GP ----> GS
                   (
                    (  +i         +r
                     ----> AP ----> NP
```

For convenience, these networks are set up using only the positive ('+') rules. The arcs point in the direction of rule application; therefore any arc may be reversed to point in the opposite direction if the valence of the rule is changed. That is, the subnetworks

```
     +rule1                          -rule1
  XX --------> YY     and      XX <-------- YY
```

are equivalent in this notation.

Notice that certain relations are constant in all three classes. These are NS:AS (-ur), AS:DP (+um), AS:GP (+a), AP:NP (+r). Furthermore, AS:DS (identity) and AS:AP (+i) are the same in classes 2 and 3, while AS:GS (+s) is the same in classes 1 and 2. These similarities can be extended trivially: for example, NS:NP (-ur +i +r) is the same in classes 2 and 3. All such relations are obviously reciprocal (if xx:yy :: XX:YY, then yy:xx :: YY:XX).

FPG is capable of specifying such similarities in order to reduce the complexity of an analysis. For example, if a fully expanded paradigm for class 1 nouns has been given, the description of class 2 would explicitly give only those processes unique to it, with a statement equivalent to "from AS, NS and DP are as for class 1, and from AP, NP is as for class1".

The graphical notation used in Figure 2.2.2 is unwieldy, especially in combination with statements about inter- and intra-paradigm similarities. To get around this, a linear notation has been developed. This notation, which will be used throughout the remainder of the text, is presented in the next section.

2.3. Linear FPG notation. There are many different ways to put relational information into the computer. These vary from painstakingly figuring out the machine representation of the data and entering it into the machine in some low-level language, to an interactive system wherein the machine asks for only the information needed and uses canned procedures for entering the material into the database. Somewhere between these two positions lie most of the extant systems. Typically, some language with a specially constrained syntax is used to enter the data into the system, and a program is used to interpret this intermediate representation and actually construct the database from it. For a discussion of a general approach to the problem, see Allen (1978). For the purposes of this project, an entry system is needed which can be written down and printed in a dissertation. Such a system should be easily understandable by nonprogrammers who are familiar with the kinds of structures used in FPG. The particular system used here is an experimental one, developed mostly for use in the dissertation. Eventually, the kind of system which will be most useful will probably involve a computer with access to the database and a high-resolution CRT-based graphics system which is capable of displaying selected portions of the network as a relational graph. Using this equipment, the linguist will enter the various paradigm relations into the graph, and when an entry is complete, the graph will be rewritten in order to accommodate the new entry. Unfortunately this kind of approach is far beyond the scope of this dissertation project. It is interesting to speculate on the relation, if any, between the problem for the linguist of mapping nonlinear models into a linear representation and the problem for the speaker of mapping nonlinear thought into the linear structure imposed by speech.

Marcus (1980) uses a notation based on the language PIDGIN to input a restricted network grammar of English. PIDGIN is a narrative language which looks similar to English, but which has a very restricted syntax and specialized semantics. It is internally translated into LISP. (For a useful introduction to LISP, see Winston and Horn 1981.) The main advantage of this approach is that the grammar then reads much like a description in English of the network; in other words, the system is self-documenting. It also enforces constraints upon the possible operations used in the grammar. In order to increase the power of the grammar, it would be necessary to change its metasyntax. For these reasons, the linearized notation for FPG is a rigidly specified English-like language which describes the operations used in creating the paradigm network(s). This language is embedded in LISP, and is interpreted by it.

The principal data objects in LISP are the atom, the list, the number, and the string. Atoms are more or less like words. Examples are the, +ur, and supercalifragilisticexpialidocious. Lists begin with the left parenthesis ′(′, end with the right parenthesis ′)′, and may contain any number of any kind of lisp object. Of course, the parentheses must balance out. An example is (this list contains (a sub-list)). The empty list () can also be referred to by the atom ′nil′. Numbers are more or less what one would expect: 3, -24, 72.1 are examples. Strings are similar to atoms in their syntax, but begin with the dollarsign character ′$′: $the.

The following is a description of the basic syntax for linearized FPG networks:

```
(2.3.1)  grammar      ::=  <paradigm>* "END"
         paradigm     ::=  "BEGIN" <p_label>* <operation>*

         p_label      ::=  ATOM
         operation    ::=  [<index>] <borrowing>


         index        ::=  FOR <p_label>*
         borrowing    ::=  "BORROW ALL FROM" <p_label> "EXCEPT"
                           <n_label> [<x>*]
                           | "BORROW" <p_label>* "FROM" <n_label> [<x>*]
         x            ::=  "AND" <action>
         action       ::=  <identity>
                           | <derivation>
                           | <property>
                           | <mapping>
                           | <movement>
                           | <deletion>
                           | <split>
         identity     ::=  "EQUATE" <n_label>*
         derivation   ::=  "DERIVE" <n_label>* "FROM" <n_label>*
                           "VIA" <rule>* ["WITH PROBABILITY" NUMBER]
         c_label      ::=  ATOM
         n_label      ::=  ATOM | NUMBER | "(" <c_label>* ")"
         rule         ::=  +ATOM | -ATOM | ?ATOM | ~ATOM
         property     ::=  "PROPERTY" ATOM [ATOM | LIST]
         mapping      ::=  "MAP" <c_label>* "TO" <c_label>*
         movement     ::=  "MOVE" <n_label>* to <n_label>*
         deletion     ::=  "DELETE" <n_label>*
         split        ::=  "SPLIT BETWEEN" <n_label> <n_label>
```

The sequence ´::=´ means ´is rewritten as´. The asterisk ´*´ indicates indefinite repetition of the preceding element (i.e. one or more times). The ´|´ symbol indicates disjunction. ATOM and NUMBER refer to the corresponding LISP data types. Material inside <angle brackets> is to be rewritten via the corresponding rule. Material inside "double-quotes" is literal material which must be present. Material inside [square brackets] is optional. Figure 2.3.2 gives a linearized version of the class 1 network of the previous section:

(2.3.2)

```
BEGIN class1
        DERIVE (nom sg) FROM (acc sg) VIA +ur
        DERIVE (dat pl) FROM (acc sg) VIA +um
        DERIVE (gen sg) FROM (acc sg) VIA +s
        DERIVE (gen pl) FROM (acc sg) VIA +a
        EQUATE (gen plu) (acc pl)
        DERIVE (dat sg) FROM (acc sg) VIA +i
        DERIVE (nom pl) FROM (acc pl) via +r
```

The interpreter uses this specification to construct a network corresponding to the paradigm. It does this by keeping a list of all nodes introduced up to the current point, and attempting to use already existing nodes to perform the specified operations. However, if no node matching an n_label is in existence, a new node is created. This implies that the first derivation operation in the class1 description would cause two new nodes to be created (NS and AS), connected via an arc carrying the rule +ur. The second derivation in class1, however, only creates one new node (DP). This node is connected to the previously created AS node via +um. The identity simply connects a node for AP to the already existing GP node via a null identity arc.

For derivations, if there is more than one rule involved, each stage corresponds to a node. However, an attempt is made to follow existing arcs as long as possible. This overlaying of arcs can be done in either direction.

For identity, the first node is considered the ′base′ and all of the other nodes in the list are connected to it via null identity arcs. Notice that if there is already a path between two of the nodes in the list the paradigm becomes ambiguous. Usually, this would simply be an error, because normally paradigms are not allowed to be ambiguous.

Now we are ready to explicate the facility of representing similarities among networks. FPG does this by borrowing a modified copy of an already existing paradigm and attaching this copy to the new paradigm. The specification of the borrowing operation must contain two kinds of information. First, it is almost always the case that only certain of the nodes in the target are relevant to the similarity. Therefore, nodes which are not relevant should not be copied. In addition, it is sometimes the case that dimensions (c_labels) in the category labels of the target must be modified. For example, if there is a similarity between a noun paradigm and an adjective paradigm, the gender and degree grades of the adjective paradigm must be deleted in the corresponding nodes of the noun paradigm.

Here are paradigms for class 2 and class 3 strong nouns, using borrowing to show the similarities:

(2.3.3)

```
BEGIN class2
        BORROW ALL FROM class1 EXCEPT (acc pl) (nom pl)
        BORROW (acc pl) (nom pl) FROM class1
        EQUATE (dat sg) (acc pl)
        MOVE (dat sg) TO (acc sg)
BEGIN class3
        BORROW ALL FROM class2 EXCEPT (gen sg)
        DERIVE (gen sg) FROM (gen pl) VIA +r
```

The specification for class2 first splits the class1 network into two sections, namely AP and NP versus all others. This is accomplished by independently borrowing the two sections. This cannot be done by the split action, because that is only appropriate for adjacent nodes (nodes connected by a single arc). At this point there is no path between any pair of nodes which are in different sections. Next, the two sections are joined together by combining the DS and AP nodes. This causes one node to be lost, and now all nodes are connected. Finally, DS is moved up to join AS. Of course, the +r arc from GP which used to lead over to NP was pruned.

Class3 is even easier to specify, since it is identical to class2 except for dative singular (in this example, anyway). The specification simply borrows everything from class2 except DS, then adds the derivation for DS.

Putting indexes on actions can be used to describe closely related paradigms such as the above. What this implies in effect is that non-indexed operations are shared by all of the paradigms in the p-label list in the beginning of the definition, but indexed operations apply only to the ones mentioned. Using this approach, here is a combined specification for all three paradigms:

```
BEGIN class1 class2 class3
        DERIVE (dat pl) FROM (acc sg) VIA +um
        DERIVE (nom sg) FROM (acc sg) VIA +ur
        DERIVE (gen pl) FROM (acc sg) VIA +a
        DERIVE (nom pl) FROM (acc pl) VIA +r
        FOR class1 EQUATE (acc pl) (gen pl)
        FOR class2 class3 DERIVE (acc pl) FROM (acc sg) VIA +i
        FOR class1 DERIVE (dat sg) FROM (acc sg) VIA +i
        FOR class2 class3 EQUATE (dat sg) (acc sg)
        FOR class1 class2 DERIVE (gen sg) FROM (acc sg) VIA +s
        FOR class3 DERIVE (gen sg) FROM (gen pl) VIA +r
```

In general, this scheme is used in those cases where there is a group of paradigms of the same general type which are very similar. In this case, the first four derivations are common among the three classes. Notice that they comprise two networks, as NP and AP have no linkage to the others. The remaining operations are all indexed. The first pair of operations connect the two common subnetworks. For class1 this is an identity, but for class2 and class3 a derivation is added. Finally, the remaining categories are added at the appropriate places in the network.

Another way to take advantage of similarities in subnetworks is by use of the DEFINE ... INCLUDE macro substitution capability. The way this works is that at some point in the grammar to be input, the shared arcs are defined by a DEFINE statement. Then at later points in the grammar, the defined material can be inserted as a body into the current paradigm. For example, if this approach were to be used for the three classes of nouns given above, the following grammar would result:

```
DEFINE sharedstuff
        DERIVE (dat pl) FROM (acc sg) VIA +um
        DERIVE (nom sg) FROM (acc sg) VIA +ur
        DERIVE (gen pl) FROM (acc sg) VIA +a
        DERIVE (nom pl) FROM (acc pl) VIA +r
BEGIN class1
        EQUATE (acc pl) (gen pl)
        DERIVE (dat sg) FROM (acc sg) VIA +i
        DERIVE (gen sg) FROM (acc sg) VIA +s
        INCLUDE sharedstuff
BEGIN class2
        DERIVE (acc pl) FROM (acc sg) VIA +i
        EQUATE (dat sg) (acc sg)
        DERIVE (gen sg) FROM (acc sg) VIA +s
        INCLUDE sharedstuff
BEGIN class3
        DERIVE (acc pl) FROM (acc sg) VIA +i
        EQUATE (dat sg) (acc sg)
        DERIVE (gen sg) FROM (gen pl) VIA +r
        INCLUDE sharedstuff
```

The main difference in these two approaches (combined paradigms versus macro substitution) is that the macro substitution is done on the input text, and involves just a blind copying operation, while the combined paradigm is interpreted while the networks are being created, and takes advantages of structural relations in the networks. In other words, macro capability is just a bell (or whistle) added to the input language, which after all is probably only a temporary one, while constructing indexed combined networks is a basic procedure of FPG.

2.4. The String Manipulation Language. From the point of view of paradigm manipulation, the details of the rule language are of secondary importance. The only requirements are that the rules be reversible and compatible with the form used for representation of linguistic items. The actual content of the rules is not of theoretical importance here. Instead, each linguistic form is viewed holistically from the perspective of the paradigm. It is up to the rule to look into this whole in order to test it or to come up with a related whole corresponding to a different node in the network. That is, there is a different grammatical point of view for the paradigm and for the rule. See Lakoff (1977) for a modern discussion of this point; e.g. Köhler (1947) for a review of the traditional notions underlying it. For present purposes, since linguistic items are represented as strings, rules will normally be stated in String Manipulation Language (SML).

Table 2.4.1 lists the special characters used in SML (X,Y,Z stand for arbitrary sequences):

```
(2.4.1)   Character(s)   Significance
          #              word boundary mark
          _#             end of word
          #_             beginning of word
          0              null
          {X,Y, ... Z}   alternatives
          (X)            X is optional
          [X=Y]          replace X with Y
          ´`"^~          diacritics
          <X>            call on macro X
```

These constructions can be freely intermingled and nested, with a few provisos. First, the bracketing characters (){}[]<> must be balanced overall, and within the following environments:

```
(2.4.2) ( ... )
        { ... ,
        , ... ,
        , ... }
        [ ... =
        = ... ]
```

The comma ´,´ and equal-sign ´=´ are limited to the environments shown in 2.4.1. The diacritics must precede either a diacritic or an alphabetic character (e.g. ca~non, m^eme, f´`´oob^a~r). The material inside angle-brackets ´< ... >´ must be the name of a macro corresponding to a subrule (e.g. {i,a,u}<NASAL> where NASAL corresponds to {m,n,~n,^n} would yield {i,a,u}{m,n,~n,^n}).

It is possible to reverse automatically any SML rule by exchanging the material from each ´[´ to the next ´=´ with the material from the ´=´ to the next ´]´. This implies that ´[=]´ constructions may not be nested. For example, simple suffixation of a constant X can be spelled _#[0=X]. This rule is reversed by swapping two fields: _#[X=0]. Notice that when suffixation is reversed, the operation is actually deletion.

Some examples of suffixation rules used in the noun paradigms are

```
(2.4.3)   process     positive form   negative form
          suffix a    _#[0=a]         _#[a=0]
          suffix um   _#[0=um]        _#[um=0]
          suffix r    _#[0=r]         _#[r=0]
```

The category {i,a,u} is deleted in the Icelandic syncope rule, which can be spelled

```
(2.4.4) +syncope <C>[{i,a,u}=0]{l,r,n}#
        -syncope <C>[0={i,a,u}]{l,r,n}#
```

The macro <C> represents the set of Icelandic consonants. Notice that -syncope is inherently ambiguous, since it is impossible to figure out which vowel must be inserted. FPG handles this by dealing only with lists of linguistic forms. For example, +syncope applied to (himin) produces (himn) and -syncope applied to (himn) produces (himin himan himun). Normally such ambiguities are not present in lexical paradigms.

It is possible to have more than one area of change. A good example of such a rule is the Hebrew "Passive Verb Rule" given in Horvath (1981). This version of the rule is translated into SML from her Figure 11:

```
(2.4.5) +pvr    (<C><V>)<C>[{i,a}=u]<C>(<C>)[{i,e}=a]<C>#
        -pvr    (<C><V>)<C>[u={i,a}]<C>(<C>)[a={i,e}]<C>#
```

(I make no claims about the aptness of her formulation or my translation of the rule, and I have omitted the information about inflectional categories given in the original). The macros <C> and <V> are assumed to contain the Hebrew consonants and vowels, respectively. The rule relates forms such as the following (adapted from her Figure 12):

(2.4.6)	'verb'		'be verb-ed'	
Past	šiber	--->	šubar	'smash'
	hixtiv	--->	huxtav	'dictate'
Present	méšaber	--->	méšubar	
	maxtiv	--->	muxtav	
Future	yéšaber	--->	yéšubar	
	yaxtiv	--->	yuxtav	

The character _é_ is a schwa.

SML is capable of performing very powerful changes in strings, but there are a few limitations in its current implementation. Some of these have to do with the lack of memory in the pattern language and the related lack of a facility to refer to portions of the input string. For example, the rules may not directly move material, since this involves copying material from one location in the string to another. Similarly, straightforward reduplication and doubling are not possible. Also, it is not possible to match repeated subexpressions: the sequence {a,b}{a,b} matches not only aa and bb, but also ab and ba. SML will be extended when it becomes necessary to include these or other manipulations in an FPG.

Another aspect of SML is the lack of phonological features. It is possible to use {alternative classes} and <macros> to achieve some of the aspects of features. We have just seen some examples of these in preceding paragraphs. The problem comes in feature-changing rules. For example, a rule devoicing final stops could be written

(2.4.7) [+stop] ---> [-voice]   / ____ #

if the features 'stop' and 'voice' were available in the system. Obviously, an attempt to translate this into anything like

(2.4.8) +devoice* [<STOP>=<UNVOICED>]_#

would fail grotesquely, since the set of unvoiced segments could include many non-stops. The correct SML formulation would have to be

(2.4.9) +devoice {[{d,t}=t],[{b,p}=p],[{g,k}=k]}_#

in order to get the alignment and reversal correct. 2.4.7 seems obviously more compact and more general than 2.4.9. However, 2.4.7 runs into problems when it is reversed,

(2.4.7′) [-voice] ---> [+stop] / ____ #

although it is easy to imagine a feature-system formulation which would be more reversible:

(2.4.7′′) [?voice] ---> [-voice] / [+stop] #

The problem of reversibility of rules for feature systems is much more serious in processes involving deletion. For example, suppose word-final stops are to be deleted rather than devoiced. The two formulations might be

(2.4.10) SML +delete [<STOP>=0]_#
Feature +delete [+stop] ---> 0 / #

The SML rule is reversed in the obvious way, but what about the Feature rule? In order to restrict the set of insertions to the set of stops actually in the language, there must be a description somewhere of the set of [+stop] segments occurring in the language. Whatever form this description takes, it is probably not going to be much more general or compact than the SML definition as a list of alternatives. Therefore, the claim being made by FPG is that for systems which deal with both encoding (production) and decoding (recognition) of morphology, such as human natural language processing, features turns out to be no more useful than lists of alternatives.

There have been several approaches to the problem of reverse application of rules. Leben and Robinson (1977) suggest that all phonology should be done in the reverse direction, that is, starting at the surface. The function of phonology, then, is to indicate lexical relations among forms (this is related to what Vennemann 1972 calls 'via rules': rules which relate but do not generate members of paradigms). The rules undo morphophonology just until a match between the surface form and the lexical form is made. In their discussion of Old Icelandic syncope, they skip over the fact that the rule is multiply ambiguous in the reverse direction. Vennemann also discusses the effect of undoing rules on language change. He discusses several cases where an historical deletion rule becomes a synchronic insertion rule (e.g. the English a(n) indefinite article from cliticized one).

There is another relevant consideration. Since the actual segments to be deleted must be specified in SML, rules can be made more efficient by limiting this set as much as possible. The syncope rule given in 2.4.4 is a good example. Since the application of the rule is lexically marked (there is no way to specify its application on solely phonological grounds), it could have been formulated as

```
(2.4.11)    +syncope    <C>[<V>=0]<C>#
                        V ---> 0 / C ___ C # .
```

As long as only forward application is considered, this is obviously to be preferred as much more compact. But consider the situation where a syncopated form is being recognized. The listener (or recognition device) must apply syncope to each form meeting its description. This includes many more for 2.4.11 than for 2.4.4. In addition, where for 2.4.4 the result is ambiguous in three ways, for 2.4.11, there is a result for every vowel in the language.

Of course, information limiting the application of syncope as much as possible on phonological grounds could be encoded into a feature analysis, for example as

(2.4.12) C [+voc -cons -tense] --> 0 / [+cons -voc +son] # .

The point is that this formulation is more complex, and the formulation in 2.4.11 would be preferred, in spite of the efficiency argument.

2.5. Rule application. The order of application of rules is strictly determined by the definitions of the paradigms. It is possible for a rule to apply more than once in a ´derivation´ and rules may apply either in a positive or negative direction, possibly both in a single derivation. It is common to refer to the ´flow of control´ of a processor ´through´ a program. The implied metaphor of water trickling through a system of rivers and streams is also useful here. The flow of control passes along one arc at a time, applying rules as needed, and changing state as a result.

2.5.1. Normal application. The normal mode of operation for a rule on an arc is that if it fails, then the nodes on the other side of the arc cannot be reached (of course there may be some other path to them, usually resulting in different forms). If the rule is ambiguous, then more than one result will be produced, and the process forks at that point. This means that unless all but one of the alternate forms are removed before reaching a labeled node, then the paradigm slot corresponding to the label has an abundance of forms for the category.

2.5.2. Abundance. As in syntax, there are certain inflectional processes which appear to be optional. An example of this is the fact that in certain Icelandic noun paradigms (Einarsson 1945 gives kviður "belly" and liður "joint" as examples) the -i ending of the dative singular may be missing. One way way that this may be represented in FPG is by using two nodes in the paradigm for the category dative singular. Other optional derivations can involve completely independent paths. For example, some nouns, such as leikur "play" have either -s or -jar as their genitive singular. In general, paradigms which have more than one node for a category are interpreted as having an abundance of forms for the category. Both forms are part of the paradigm, as for fish (plu.) ~ fishes, brothers ~ brethren.

2.5.3. Vacuous application. Another mode of application for rules is primarily useful in cases where it is desired to apply some general phonological process and it is not known whether the environment will be met. A normal rule will abort the arc if the environment is not met, but this is not what is desired. Optional rules as they are usually implemented would be far too inefficient, since they are always ambiguous. Vacuous rules in FPG are implemented in such a way that they attempt to apply, and if they succeed, the output becomes the new form. If they fail, however, the arc is still traversed, but the input form of the rule is used in the next node as well. To take a concrete example, there is a process in Icelandic known as u-shift (or u-umlaut). This process, in its simplest form, changes a to ö. It can be phonologically or paradigmatically triggered. It is very often the case that words vulnerable to u-shift pattern exactly like other words which are not. In order to capture this generalization, it is sometimes desirable to use a single paradigm for both categories of words. Therefore, it is possible to mark rules as optional by enclosing them in parentheses. This mode of application is most similar to the ordinary phonological rule, which is allowed to apply vacuously if its input pattern is not matched.

2.5.4. Identity. If the name of the rule on an arc is the atom identity, then the two nodes are identical in form. However, since the arc is present, the network can be split apart between the two nodes, which would not be possible if the two nodes were collapsed into a single node with extra labels. These identity arcs are produced either explicitly by the "derive x from y via identity" action, or implicitly using the equate action. Conceptually, the rule can be thought of as returning its input unchanged. In addition, identity arcs are sometimes useful when it is desired to assign a probability level to alternate realizations. They correspond more or less to ATN jump arcs.

2.5.5. Conditional arcs. Some rules are used not to modify the linguistic form of an item, but to guide the flow of control through the network. Such rules return either the input form or nothing. Unlike structure-changing rules, these rules can be positive or negative. We will informally adopt the convention of prefixing '?' to the names of positive conditions, and '~?' to the names of negative conditions. Positive conditions return their input forms if and only if their condition is met; negative conditions return their input forms if and only if their condition fails.

2.6. Self-sufficiency. A constraint on FPG paradigms is that all paradigms must be self-sufficient. A paradigm is self-sufficient just if it is possible to derive all of the members of the paradigm from any single member, without using external information. That is, knowledge of the paradigm to which a certain form belongs and the category of the form must be sufficient to resolve all uncertainty about all of the other members of the paradigm. This restriction implies that certain common types of process, such as deletion of a member of a category (such as 'vowel' or 'narrow vowel') or mutation (such as u-shift or i-shift) cannot normally be used. The reason for this becomes apparent after a moment's consideration. All processes must be reversible, which means that it must be possible to uniquely recover the input material from the output. An example was given above in the discussion of syncope and a similar argument holds for mutation rules. The situation with respect to syncope is discussed for Swedish in Eliasson (1972). A main point of his discussion is that an epenthesis rule which inserted a vowel is not adequate in comparison to a syncope rule, since it is not possible to predict the vowel which is to be inserted. However, since the reverse of a generalized syncope rule is in effect an ambiguous epenthesis rule, syncope as usually formulated cannot be used in FPG paradigms at all (unless the vowel in question can be predicted from context).

This situation seems somehow repugnant. What good is a theory if it forces us to toss out some of the hoariest generalizations about the language? Besides, there seems to be no question that the native speaker ´knows´ in some sense that the individual instances of syncope, u-shift, and i-shift are functionally similar. Furthermore, the native speaker can make use of information present in certain forms in a paradigm to predict other forms. That is, if any nonsyncopated form is known, it is possible to use the fact that a certain vowel is present to make a guess as to the other nonsyncopated forms. This kind of knowledge, while not represented in individual paradigms due to the self-sufficiency restriction, can be represented in FPG by use of ´bulges´, arc ´indexing´, and arc ´conditions´.

A ´bulge´ is a subnetwork which contains two terminal nodes with more than one path between them. Graphically, this looks a little like a swollen point in the network, hence its name. Bulges result from combining paradigms on the basis of partial similarity, and correspond to areas of possible ambiguity in the combined paradigm. There are bulges which are non-ambiguous: for example, there is a rule which can be stated

(2.6.1) <V>{l[0=l],n[0=n]}#

which is used in various paradigms. Since this rule contains alternatives, it is in essence a bulge. It could be restated as two rules, each on its own arc. The rule is nonambiguous because the distinctive properties of the forms are present in both the forward and reverse directions (forward: add l for stems ending in l, and add n for stems ending in n; reverse: delete final l before l, delete final n before n).

Most bulges, however, are ambiguous. For example, there is a constraint on the formation of masculine/neuter genitive singular. The usual rule is to add s to the stem. There is a single exception to this: if the stem ends in consonant + s, then the unmodified stem is used (fús ~ fúss, gulur ~ guls; but hvass ~ hvass, frjáls ~ frjáls). In the forward direction there is no ambiguity, since the constraint as stated would produce the correct forms in all cases. But in the reverse direction, it is impossible to distinguish the two situations. The factors here are: (1) knowing the stem, it is always possible to predict uniquely the correct genitive; (2) for many genitives (i.e., those not ending in consonant + s), it is always possible to predict the correct stem; (3) making a guess in one of the ambiguous cases constitutes a hypothesis about which paradigm(s) the word belongs to.

It is possible to encode these things into a combined network by putting indexes on all arcs and labels which essentially express the notion that "traversing this arc (or using this label) is consistent only with the following paradigm(s): [list of names of paradigms]". This is to be interpreted as doing two things. First, if the paradigms in the index list do not overlap with the set of hypotheses currently under consideration, then the arc or label is blocked. Secondly, the set of current hypotheses is reduced to the intersection of the indexes on the arc or label and the set of current hypotheses. If the set of current hypotheses is empty, then the derivation is blocked, and the next backed-up alternative is attempted. Thus the convention of indexing arcs and labels in combined networks guarantees at least that derivations resulting in more than one form for a label will correctly classify each form as to which paradigm(s) it can be a member.

Applying this to the genitive problem above, if the genitive input were hests, the AS outputs would be hests, indexed for only consonant + s paradigms, and hest, indexed only for the complementary paradigms. Furthermore, indexing guarantees that the combined paradigm is self-sufficient for each of the individual paradigms in it, because if the paradigm name is known, only the arcs and labels in the combined paradigm can be taken.

Explicit indexing of paradigms is accomplished in the entry language through the optional "FOR <index>*" preliminary sequence on actions. In the example of the use of these in the previous section, there was a line

FOR class1 class2 DERIVE (gen sg) FROM (acc sg) VIA +s

Ignoring other differences, the GSs for these nouns with ASs ending in Cs (let's call them class4) can be incorporated into this system by changing this line to

```
FOR class1 class2 DERIVE (gen sg) FROM (acc sg) VIA ~Cs +s
FOR class4 DERIVE (gen sg) FROM (acc sg) VIA ?Cs
```

where ?Cs and ~Cs are conditions rather than rules. The negative condition ~Cs accepts all forms which do not end in consonant + s; the positive condition ?Cs accepts only those forms which do. In the forward direction AS-->GS, this bulge is nonambiguous even in the absence of lexical information, because ~Cs and ?Cs are mutually incompatible. In the reverse direction, it is obviously ambiguous, but the information about paradigm membership which is implied by the alternatives is available for eventual disambiguation. The portion of the combined paradigm under consideration could be graphically represented as follows:

```
                  ~Cs           +s
------> AS -----------> ----------> GS
        |  [class1,2]   [class1,2]   ^
        |                            |
        |           ?Cs              |
        |----------------------------|
                  [class4]
```

Of course, there are bulges where all paths change form, although in this example, one path doesn't change form, in contrast to the other.

CHAPTER 3

SOME ICELANDIC PARADIGMS

This chapter covers two of the primary Icelandic inflectional paradigms: adjectives and nouns. I make no pretense of introducing new data. The analyses here are based on data in Kress (1963) and Einarsson (1945), for the most part, with some reference to Cleasby, Vigfusson, and Craigie (1957). The overall purpose of the chapter is to study the mechanisms of FPG as outlined above in more detail by applying them to specific examples. In addition, the process of performing the analysis helped in the development of the linear entry language, and helped clarify certain points about how an FPG must perform.

3.1. Icelandic Orthography. This is a brief guide to the system which will be used here to write Icelandic forms. It is based upon the standard orthography used in modern Icelandic publications. First, this table contains the most common values for the vowels as they are pronounced in careful speech.

Orthography	IPA	Comments
a	[a]	ang,ank = [aung,aunk]; au = [öy]
á	[au]	
e	[e]	
é	[je]	
i	[I]	ing,ink = [ing,ink]
í	[i]	
o	[o]	open as in ´law´; ong = [oung]
ó	[ou]	
u	[y]	ung,unk = [ung,unk]
ú	[u]	
y,ý	[I,i]	(same as i,í above)
æ	[ai]	(usually an ae digraph)
ö	[ö]	

There are a few diphthongs which should be listed here as well:

Orthography	IPA
au	[öy]
ei	[ei]
ey	[ei]

The consonants are more or less as in modern German, with a few exceptions:

Orthography	IPA	Comments
þ	[θ]	(unvoiced interdental)
ð	[ð]	(voiced interdental)
s	[s]	never voiced
r	[r]	apical trill
rl,ll	[dl]	often not voiced
rn,nn	[dn]	often not voiced

The Icelandic vowels and diphthongs have been divided into two classes, here called ´strong´ (breiðir) and ´weak´ (grannir). The ´strong vowels´ are á, é, í, ó, ú, ý, æ, au, ei, and ey. The ´weak vowels´ are i, y, e, a, u, ö, o.

Kress uses a system of classification for stems based on presence or absence of final consonants and the type of the root vowel. Light (leicht) stems have either a weak (leicht) vowel followed by one consonant or a strong (schwer) vowel with no following consonant. Heavy (schwer) stems have either a weak vowel followed by two or more consonants or a strong vowel with one or more following consonants. In addition, all polysyllabic stems are heavy. Obviously, the translations of grannir, breiðir, schwer, and leicht are compromises at best.

Use of this orthography sets this analysis apart from most morphological analyses. Usually the sound system is reduced to some set of abstract symbols which is felt to be useful in terms of simplification of other parts of the grammar, either in the phonetics, the phonology, or the morphology (Haugen 1972a, 1972b, Anderson 1972, 1969, Garnes 1971). There are two reasons why this is not done here. The first reason is practical--if the system is to be used to deal with Icelandic texts, then the project would become much more complex if a more abstract and controversial representation were used. The second reason is that in FPG, the representation of linguistic forms is only of secondary importance. In the more usual process models, the form often contains abstract segments which guide the flow of the process through the set of rules. This kind of thing is not nearly as desirable in FPG, because in the first place, the linguistic forms at all labeled nodes in the system must be terminal (surface) forms, and in the second place, much of the formal economy which seems to result from the very abstract kind of phonological treatment is not even possible in a system where all sequences of rules must apply in either direction, without introducing ambiguity. Because of this convention (using standard orthography), this analysis will probably be more useful, but less general. Since the standard orthography is reasonably consistent with respect to pronunciation, many of the generalizations made elsewhere based on phonetics or phonology will still be incorporated here when desirable.

3.2. Icelandic Adjectives. The adjectives were chosen as a starting point because they display most of the important phonological alternations in the language, and because their complexity provides a better testing ground for the problem of cross-paradigm similarity. The principal paradigms are given fully so that they may be referred to as needed. In subsequent sections, this will not be the case.

The plan of attack for adjectives (and for the other forms as well) will be to find subnetworks which are in common among many paradigms, with respect to both endings and stem form variations. The various stem forms will also be isolated, and rules will be used to change one stem form into another. Then a combined, indexed paradigm will be used to illustrate the similarities among the forms. New rules will be discussed when they are first used, but the discussion may refer to later sections.

3.2.1. Some Exemplary Paradigms for Adjectives. Icelandic adjectives are inflected for case, number, gender, and degree of comparison. In addition, there is a strong/weak distinction in positive and superlative with a distribution similar to that in German. Because of considerable syncretism, there are in fact only 31 distinct forms for the whole declension; in some paradigms further reduction is found. The following tables contain some redundancy to make them easier to use. They are adapted from Einarsson (1945).

3.2.1.1. Adjectives with little stem change: gulur. Many adjectives show essentially no alternation in their stem forms. We start with one of these to illustrate the suffixal processes.

gulur "yellow" (adjective)

				Masc.	Fem.	Neut.
Po.	St.	Nom.	Sg.	gulur	gul	gult
			Pl.	gulir	gular	gul
		Acc.	Sg.	gulan	gula	gult
			Pl.	gula	gular	gul
		Dat.	Sg.	gulum	gulri	gulu
			Pl.	gulum	gulum	gulum
		Gen.	Sg.	guls	gulrar	guls
			Pl.	gulra	gulra	gulra
	Wk.	Nom.	Sg.	guli	gula	gula
			Pl.	gulu	gulu	gulu
		Obl.	Sg.	gula	gulu	gula
			Pl.	gulu	gulu	gulu
Comp.			Sg.	gulari	gulari	gulara
			Pl.	gulari	gulari	gulari
Su.	St.	Nom.	Sg.	gulastur	gulust	gulast
			Pl.	gulastir	gulastar	gulust
		Acc.	Sg.	gulastan	gulasta	gulast
			Pl.	gulasta	gulastar	gulust
		Dat.	Sg.	gulustum	gulastri	gulustu
			Pl.	gulustum	gulustum	gulustum
		Gen.	Sg.	gulasts	gulastrar	gulasts
			Pl.	gulastra	gulastra	gulastra
	Wk.	Nom.	Sg.	gulasti	gulasta	gulasta
			Pl.	gulustu	gulustu	gulustu
		Obl.	Sg.	gulasta	gulustu	gulasta
			Pl.	gulustu	gulustu	gulustu

This is the most common type of adjective declension. Notice that most of the categories here can be formed by suffixation alone; there are essentially no other morphophonemic processes. There is one principal stem form used here, gul-, plus a comparative stem gular- and two superlative stems, gulast- and gulust-. The presence of two distinct superlative stems here (which is not always the case) is due to the rule of u-shift, which is discussed more fully in the next section.

3.2.1.2. U-shift: svalur. Next comes a paradigm which is essentially similar to the first one, but which shows a very basic and simple stem alternation:

svalur "cool" (adjective)

			Masc.	Fem.	Neut.
Po. St.	Nom.	Sg.	svalur	svöl	svalt
		Pl.	svalir	svalar	svöl
	Acc.	Sg.	svalan	svala	svalt
		Pl.	svala	svalar	svöl
	Dat.	Sg.	svölum	svalri	svölu
		Pl.	svölum	svölum	svölum
	Gen.	Sg.	svals	svalrar	svals
		Pl.	svalra	svalra	svalra
Wk.	Nom.	Sg.	svali	svala	svala
		Pl.	svölu	svölu	svölu
	Obl.	Sg.	svala	svölu	svala
		Pl.	svölu	svölu	svölu
Comp.		Sg.	svalari	svalari	svalara
		Pl.	svalari	svalari	svalari
Su. St.	Nom.	Sg.	svalastur	svölust	svalast
		Pl.	svalastir	svalastar	svölust
	Acc.	Sg.	svalastan	svalasta	svalast
		Pl.	svalasta	svalastar	svölust
	Dat.	Sg.	svölustum	svalastri	svölustu
		Pl.	svölustum	svölustum	svölustum
	Gen.	Sg.	svalasts	svalastrar	svalasts
		Pl.	svalastra	svalastra	svalastra
Wk.	Nom.	Sg.	svalasti	svalasta	svalasta
		Pl.	svölustu	svölustu	svölustu
	Obl.	Sg.	svalasta	svölustu	svalasta
		Pl.	svölustu	svölustu	svölustu

This paradigm is identical to that for gulur except for the alternations between a and ö or u. Since this alternation was conditioned historically by the presence of a u (or other labial) to the right of the alternating segment, it is known as u-umlaut or u-shift. However, in modern times, the presence of an explicit conditioning segment is not required (e.g. svöl, svölust). In addition, note that presence of a following u does not necessarily trigger u-shift (e.g. svalur). Using surface phonological information only, it is not possible to predict whether u-shift will result in ö or u. This is primarily due to the lack of obvious word or morpheme boundary marks in surface forms, due to prefixing as in ó+gammall "not" + "old" = ógammal "not old" or compounding as in panna+kaka "pan" + "cake" = pönnukaka "pancake" (note the oblique case form in the first element).

Keeping this limitation in mind, here is a summary of the general facts. When the alternating segment is in the second syllable of a two-syllable word, it varies between a and u; if it is in the first syllable, it varies between a and ö. An intervening vowel other than a/u, a word boundary (as in compounds) or a prefix boundary (af-langur ~ af-löng "oblong") block application of u-shift. There are certain morphological situations where the rule is blocked (as already seen above), and in addition there are apparently certain loan words which do not u-shift their root vowels even when this would be normally expected (John Lindow, personal communication). Furthermore, there are second-syllable a's which u-shift to ö heilagur ~ heilög "holy"; apaldur ~ apöldrum "apple-tree" (note especially that the first syllable is not u-shifted here). Einarsson uses the construct 'suffix' to account for these variations. In his system, a goes to u only in suffixes, otherwise to ö. The problem with this system is that the suffixation often has only historical relevance--today the suffix may be unproductive, and the historical root may no longer exist as a free form. For our purposes, suffice it to say that u-shift is conditioned by phonological context, the lexicon, and by the morphological situation.

Note that the addition of the u-shift alternation results in an additional stem form in the positive. The u-shifted positive and superlative stems coöccur, and so do the non-u-shifted stems (**svölast*, **svalust*). In these and many other polysyllabic words which contain the pattern *aCCa* in last and next-to-last syllables, u-shift effects both of the *a*´s. The most usual result is shift to *öCCu* as in *svalast* ~ *svölust*, but another possibility is *öCCö* as in the noun *kafald* ~ *köföld* "thick fall of snow" (but I believe this to be rare). In order to keep these alternatives straight, the following naming convention will be used for these and other vowel-shift rules (capital V indicates the input vowel; lowercase v indicates the output vowel): If a single vowel or diphthong is affected, the form of the rule´s name is

{+,-}<V>:<v> .

If two vowels or diphthongs are effected, the rule´s name is of the form

{+,-}<V1>-<V2>:<v1>-<v2> .

This format can easily be extended to more vowels if needed, and if the rule refers to a class of vowels being mapped into another class of vowels, then the rule name can use the class-name in place of V and v. Using this convention, the following rules will be collectively referred to as u-shift:

+a:ö
+a:u
+a-a:ö-u
+a-a:ö-ö

The exact formulation of the rules in SML will not be given, but it should be clear how to formulate them, once the distribution of the surrounding consonants is known. The surrounding environment should be specified as precisely as possible to minimize wrong paths during recognition. For example, a correct formulation for +a-a:ö-ö is

{k,h}[a=ö]f[a=ö]ld#

because only kafald and hafald "heddle, heald (weaving term)" show it, as far as I know. There are probably a few other words which use the rule; the point is that the number is small. Furthermore, since +a:u and +a-a:ö-u apparently are present only in words with suffixes (inflectional or other), and the segmental material in suffixes is reduced in inventory compared to roots, the surrounding environments for these rules would also be less complex than for +a:ö. For example, study of Jóhannesson (1927)'s fairly thorough listing of old suffixes yields only the following as possible candidates:

að	nað	ak	al
ald (Note)	an	and	ang
ar	arð	arn	ast
at			

Note: ald may use +a:ö or +a-a:ö-ö.

This particular list could be summarized in SML as

{nað,a{ð,k,l(d),n({d,g}),r({ð,n}),(s)t}}

and incorporated into +a:u and +a-a:ö-u. As was stated in the previous chapter, the grammar has both a set of paradigm definitions and a dictionary keyed to the definitions. These (and other) rules could be updated as items which use them are entered in the dictionary.

3.2.1.3. Young-u, i-shift: fagur. It is common for adjectives with essentially the same phonological type as the preceding one to show more complex alternations in the comparative and superlative:

fagur "beautiful, fair" (adjective)

				Masc.	Fem.	Neut.
Po.	St.	Nom.	Sg.	fagur	fögur	fagurt
			Pl.	fagrir	fagrar	fögur
		Acc.	Sg.	fagran	fagra	fagurt
			Pl.	fagra	fagrar	fögur
		Dat.	Sg.	fögrum	fagurri	fögru
			Pl.	fögrum	fögrum	fögrum
		Gen.	Sg.	fagurs	fagurrar	fagurs
			Pl.	fagurra	fagurra	fagurra
	Wk.	Nom.	Sg.	fagri	fagra	fagra
			Pl.	fögru	fögru	fögru
		Obl.	Sg.	fagra	fögru	fagra
			Pl.	fögru	fögru	fögru
Comp.			Sg.	1 fegri	1 fegri	1 fegra
			Pl.	1 fegri	1 fegri	1 fegri
Su.	St.	Nom.	Sg.	2 fegurstur	3 fegurst	2 fegurst
			Pl.	2 fegurstir	2 fegurstar	3 fegurst
		Acc.	Sg.	2 fegurstan	2 fegursta	2 fegurst
			Pl.	2 fegursta	2 fegurstar	3 fegurst
		Dat.	Sg.	3 fegurstum	2 fegurstri	3 fegurstu
			Pl.	3 fegurstum	3 fegurstum	3 fegurstum
		Gen.	Sg.	2 fegursts	2 fegurstrar	2 fegursts
			Pl.	2 fegurstra	2 fegurstra	2 fegurstra
	Wk.	Nom.	Sg.	2 fegursti	2 fegursta	2 fegursta
			Pl.	3 fegurstu	3 fegurstu	3 fegurstu
		Obl.	Sg.	2 fegursta	3 fegurstu	2 fegursta
			Pl.	3 fegurstu	3 fegurstu	3 fegurstu

Alternate stems: (1) fegurr- fagrar- (2) fagrast (3) fögrust

Ignoring the comparative and superlative for the moment, this adjective, in addition to exhibiting u-shift, alternates between the stem forms fagur- and fagr-. The variation is conditioned by the presence of an immediately following consonant. The case where the stem alternates between -<u>ur</u>- and -<u>r</u>- is called young-u, expanded-r, or u-epenthesis; other similar cases involving slightly different contexts are referred to as syncopation (see below). Historically, the young-u forms did not alternate; they always appeared with <u>r</u> alone. Later, when the <u>u</u> appeared, it did not trigger u-shift. In what follows, this rule will be referred to as +r:ur.

Another alternation in <u>fagur</u> is between <u>fag</u>- and <u>feg</u>- in the comparative and superlative forms. This alternation was triggered historically by the presence of <u>i</u> (or <u>j</u>) in the following syllable, and is referred to as i-shift, i-umlaut, or i-mutation. I-shift as a general process consists of the following individual vowel-shift rules: +a:e, +ö:e, +á:æ, +e:i, +o:e, +o:y, +ó:æ, +u:y, +ú:ý, +ju:y, +jú:ý, +jó:ý, and +au:ey. Notice that except for +o:e and +o:y, i-shift can be predicted solely on phonological grounds if the input vowel is known, although knowing the output vowel is much less useful.

The other variation in stem forms for <u>fagur</u> is between <u>f{a,e}gur</u>- and <u>f{a,e}gr</u>- in comparative and superlative. Notice the interaction between young-u and syncopation in the stem <u>f{a,e}g(u)r</u>- and in the comparative -(<u>a</u>)<u>r</u>- and the superlative -({<u>a</u>,<u>u</u>})<u>st</u>-. This is probably due to alternative comparative and superlative suffixes, certain of which begin with vowels (and trigger syncope) and others of which do not.

In what follows, we will have occasion to refer to stem forms which precede suffixes beginning with a vowel as vstems (astem, ustem, istem), and to the stem forms which precede suffixes beginning with a consonant as the cstems (castem custem cistem).

3.2.1.4. Stems in {á,ó,ú}: hár. The preceding paradigms all involved weak root vowels, and had stem forms ending in a consonant. The next group has a stem ending in a strong root vowel:

hár "high, tall, loud" (adjective)

				Masc.	Fem.	Neut.
Po.	St.	Nom.	Sg.	hár	há	hátt
			Pl.	háir	háar	há
		Acc.	Sg.	háan	háa	hátt
			Pl.	háa	háar	há
		Dat.	Sg.	háum	hárri	háu
			Pl.	háum	háum	háum
		Gen.	Sg.	hás	hárrar	hás
			Pl.	hárra	hárra	hárra
	Wk.	Nom.	Sg.	hái	háa	háa
			Pl.	háu	háu	háu
		Obl.	Sg.	háa	háu	háa
			Pl.	háu	háu	háu
Comp.			Sg.	hærri	hærri	hærra
			Pl.	hærri	hærri	hærri
Su.	St.	Nom.	Sg.	hæstur	hæst	hæst
			Pl.	hæstir	hæstar	hæst
		Acc.	Sg.	hæstan	hæsta	hæst
			Pl.	hæsta	hæstar	hæst
		Dat.	Sg.	hæstum	hæstri	hæstu
			Pl.	hæstum	hæstum	hæstum
		Gen.	Sg.	hæsts	hæstrar	hæsts
			Pl.	hæstra	hæstra	hæstra
	Wk.	Nom.	Sg.	hæsti	hæsta	hæsta
			Pl.	hæstu	hæstu	hæstu
		Obl.	Sg.	hæsta	hæstu	hæsta
			Pl.	hæstu	hæstu	hæstu

This paradigm is similar to the preceding ones, with three principal exceptions. First, it shows i-shift in comparative and superlative without variation, and the initial a in these endings is always missing. Second, it suffixes -tt instead of -t to form neuter singular nominative and accusative. This is a property of those paradigms where the cstems end in a vowel. Third, it uses a stem in -rr- in those cases where the preceding paradigms use -r- (the doubled -r- in fagur is not the same thing because one of them is there in all the other stems as well). Again, this doubling is a characteristic of paradigms whose cstems end in a vowel. This compound stem which is basically castem+r or castem+rr will be referred to as the rstem, even though there are cases (see below) where no r is actually present.

It should be noted that only certain vowels occur as the final segment of adjective stems: those with á-, ó-, and ú- pattern like hár, and those with ý- and æ- go like the next example. There are no other final stem vowels.

3.2.1.5. Stems in {ý,æ}: nýr. The other type of adjective in the class with stems ending in strong root vowels differs significantly from the preceding type:

ný́r "new" (adjective)

				Masc.	Fem.	Neut.
Po.	St.	Nom.	Sg.	nýr	ný	nýtt
			Pl.	nýjir	nýjar	ný
		Acc.	Sg.	nýjan	nýja	nýtt
			Pl.	nýja	nýjar	ný
		Dat.	Sg.	nýjum	nýrri	nýju
			Pl.	nýjum	nýjum	nýjum
		Gen.	Sg.	nýs	nýrrar	nýs
			Pl.	nýrra	nýrra	nýrra
	Wk.	Nom.	Sg.	nýji	nýja	nýja
			Pl.	nýju	nýju	nýju
		Obl.	Sg.	nýja	nýju	nýja
			Pl.	nýju	nýju	nýju
Comp.			Sg.	nýrri	nýrri	nýrra
			Pl.	nýrri	nýrri	nýrri
Su.	St.	Nom.	Sg.	nýjastur	nýjust	nýjast
			Pl.	nýjastir	nýjastar	nýjust
		Acc.	Sg.	nýjastan	nýjasta	nýjast
			Pl.	nýjasta	nýjastar	nýjust
		Dat.	Sg.	nýjustum	nýjastri	nýjustu
			Pl.	nýjustum	nýjustum	nýjustum
		Gen.	Sg.	nýjasts	nýjastrar	nýjasts
			Pl.	nýjastra	nýjastra	nýjastra
	Wk.	Nom.	Sg.	nýjasti	nýjasta	nýjasta
			Pl.	nýjustu	nýjustu	nýjustu
		Obl.	Sg.	nýjasta	nýjustu	nýjasta
			Pl.	nýjustu	nýjustu	nýjustu

These words, with cstems in <u>ý</u> or <u>æ</u> (and one other word: <u>miður</u> "in the middle"), add a -<u>j</u>- to derive the vstems from the corresponding cstems. Notice here that the comparative stem (compstem) and rstem happen to be the same. That this is true here but not in <u>hár</u> is probably due to the fact that <u>ý</u> is not i-shiftable. This suggests that the process responsible for the variation in rstems (<u>r</u> versus <u>rr</u> so far) is more general (probably phonologically conditioned), and can perhaps apply at other points in the paradigm.

3.2.1.6. Stems in V+r: dýr. While superficially similar to the preceding class (stems ending in strong root vowels), this class has a strong root vowel, but a stem endinng in r:

dýr "dear, expensive" (adjective)

				Masc.	Fem.	Neut.
Po. St.		Nom.	Sg.	dýr	dýr	dýrt
			Pl.	dýrir	dýrar	dýr
		Acc.	Sg.	dýran	dýra	dýrt
			Pl.	dýra	dýrar	dýr
		Dat.	Sg.	dýrum	dýrri	dýru
			Pl.	dýrum	dýrum	dýrum
		Gen.	Sg.	dýrs	dýrrar	dýrs
			Pl.	dýrra	dýrra	dýrra
	Wk.	Nom.	Sg.	dýri	dýra	dýra
			Pl.	dýru	dýru	dýru
		Obl.	Sg.	dýra	dýru	dýra
			Pl.	dýru	dýru	dýru
Comp.			Sg.	1 dýrari	1 dýrari	1 dýrara
			Pl.	1 dýrari	1 dýrari	1 dýrari
Su. St.		Nom.	Sg.	2 dýrastur	3 dýrust	2 dýrast
			Pl.	2 dýrastir	2 dýrastar	3 dýrust
		Acc.	Sg.	2 dýrastan	2 dýrasta	2 dýrast
			Pl.	2 dýrasta	2 dýrastar	3 dýrust
		Dat.	Sg.	3 dýrustum	2 dýrastri	3 dýrustu
			Pl.	3 dýrustum	3 dýrustum	3 dýrustum
		Gen.	Sg.	2 dýrasts	2 dýrastrar	2 dýrasts
			Pl.	2 dýrastra	2 dýrastra	2 dýrastra
	Wk.	Nom.	Sg.	2 dýrasti	2 dýrasta	2 dýrasta
			Pl.	3 dýrustu	3 dýrustu	3 dýrustu
		Obl.	Sg.	2 dýrasta	3 dýrustu	2 dýrasta
			Pl.	3 dýrustu	3 dýrustu	3 dýrustu

Alternate stems: (1) dýrr- (2, 3) dýrst-

This paradigm is actually nearly identical to that of fagur, although of course no u-shift, i-shift, or syncope is possible due to the form of the root (except u-shift in the superlative). Notice, however, that the masculine nominative singular and the rstem of this form correspond to those for hær and nýr in the absence of other information. That is, based only on any of these strings, it is impossible to determine whether the castem ends in r or ý. It would be possible to derive, say, positive genitive plural from positive masculine singular masculine, even in the absence of lexical information, although some of these derivations would be characterized by false analysis of the internal structure of the strings. (But FPG analysis does not presuppose any consistent analysis of the internal structure of linguistic strings.)

3.2.1.7. Heavy stems in -n: seinn. Now we have a strong root vowel, similar to the preceding class, but the stem ends in n:

seinn "slow" (adjective)

				Masc.	Fem.	Neut.
Po.	St.	Nom.	Sg.	seinn	sein	seint
			Pl.	seinir	seinar	sein
		Acc.	Sg.	seinan	seina	seint
			Pl.	seina	seinar	sein
		Dat.	Sg.	seinum	seinni	seinu
			Pl.	seinum	seinum	seinum
		Gen.	Sg.	seins	seinnar	seins
			Pl.	seinna	seinna	seinna
	Wk.	Nom.	Sg.	seini	seina	seina
			Pl.	seinu	seinu	seinu
		Obl.	Sg.	seina	seinu	seina
			Pl.	seinu	seinu	seinu
Comp.			Sg.	seinni	seinni	seinna
			Pl.	seinni	seinni	seinni
Su.	St.	Nom.	Sg.	seinastur	seinust	seinast
			Pl.	seinastir	seinastar	seinust
		Acc.	Sg.	seinastan	seinasta	seinast
			Pl.	seinasta	seinastar	seinust
		Dat.	Sg.	seinustum	seinastri	seinustu
			Pl.	seinustum	seinustum	seinustum
		Gen.	Sg.	seinasts	seinastrar	seinasts
			Pl.	seinastra	seinastra	seinastra
	Wk.	Nom.	Sg.	seinasti	seinasta	seinasta
			Pl.	seinustu	seinustu	seinustu
		Obl.	Sg.	seinasta	seinustu	seinasta
			Pl.	seinustu	seinustu	seinustu

Notice the identity of the comparative stem seinn- to the rstem and that the derivation of the rstem involves doubling the final n- rather than adding -r-. The same process is also used in forming the masculine nominative singular. In general, this process, called here +r:n, is related to +r:ur and applies in more or less the same environment, except the preceding consonant must be n. However, the environment is actually more complicated than that, since +r:ur can apply to stems ending in -n: munur "thing, difference" (mun > +r +r:ur > munur). The general rule is to use +r:ur if the stem has no suffix and the vowel is weak, or if a consonant precedes the -n; use +r:n otherwise. The environment for +r:ur after n corresponds to Kress's light stem; the environment for +r:n corresponds to his heavy stem.

3.2.1.8. Disyllabic stems in -l, syncope: gamall. The adjectives in the next groups are historically derived via an old (no longer productive) set of suffixes, resulting in disyllabic adjectives which have certain unique characteristics:

gamall "old" (adjective)

				Masc.	Fem.	Neut.
Po. St.	Nom.	Sg.		gamall	gömul	gamalt
		Pl.		gamlir	gamlar	gömul
	Acc.	Sg.		gamlan	gamla	gamalt
		Pl.		gamla	gamlar	gömul
	Dat.	Sg.		gömlum	gamalli	gömlu
		Pl.		gömlum	gömlum	gömlum
	Gen.	Sg.		gamals	gamallar	gamals
		Pl.		gamalla	gamalla	gamalla
Wk.	Nom.	Sg.		gamli	gamla	gamla
		Pl.		gömlu	gömlu	gömlu
	Obl.	Sg.		gamla	gömlu	gamla
		Pl.		gömlu	gömlu	gömlu
Comp.		Sg.		eldri	eldri	eldra
		Pl.		eldri	eldri	eldri
Su. St.	Nom.	Sg.		elztur	elzt	elzt
		Pl.		elztir	elztar	elzt
	Acc.	Sg.		elztan	elzta	elzt
		Pl.		elzta	elztar	elzt
	Dat.	Sg.		elztum	elztri	elztu
		Pl.		elztum	elztum	elztum
	Gen.	Sg.		elzts	elztrar	elzts
		Pl.		elztra	elztra	elztra
Wk.	Nom.	Sg.		elzti	elzta	elzta
		Pl.		elztu	elztu	elztu
	Obl.	Sg.		elzta	elztu	elzta
		Pl.		elztu	elztu	elztu

Shows u-shift, syncope, and suppletion in comparative and superlative. Notice that masculine nominative singular and rstem are formed by doubling the final l in the castem, similar to the procedure in seinn. This process, +r:l, has roughly the same environment as +r:n.

Syncope is a process which causes a suffix vowel to be deleted. It is generally triggered by adding another suffix which begins with a vowel, therefore it is often present in the vstems. All syncopated vowels were originally in suffixes, and if they happen to be u, the root vowel is never a, even in the u-less forms.

There is another restriction on the application of syncope: only the suffixes

al ul il ar ur ir an un in

show it. (Note: ur results from u-epenthesis and ir occurs only in a few nouns). This goes along with the general trend toward simplification of the environment for application of rules as much as possible. The process could be generalized in SML as

[{a,u,i}=0]{l,r,n}#

Notice that when a vowel is deleted by syncope, it introduces opacity into the paradigm, as did the reduction in i-shift above. That is, without lexical information it is impossible to insert the correct vowel uniquely (in most cases). Furthermore, let us suppose that syncope is represented as a single process, as in the above SML rule, and as a result, various paradigms showing syncope of different vowels but identical in other ways are combined. In this case, one of the basic requirements of an FPG analysis would not be met: it is essential that paradigms be defined in such a way that given a surface form and its category and the paradigm to which it belongs, any of the other forms in the paradigm can be uniquely derived. But if, for example, we have gamli, and know that it is the weak masculine nominative singular of paradigm X, and paradigm X uses the generalized syncope rule we have been using, then we must try three astems: gamul-, gamil-, and gamal-. There would be no way to decide which is correct. Therefore, we will use three different rule-names to keep them distinct: +aC:C, +iC:C, and

+uC:C. Note again that +r:ur looks very much like a special case of -uC:u.

3.2.1.9. Cstems in -il: mikill. In addition to syncope, these disyllabic adjectives have some peculiarities not shared in other classes. However, they are very common in use:

mikill "much, large, big, great" (adjective)

				Masc.	Fem.	Neut.
Po. St.		Nom.	Sg.	mikill	mikil	mikiđ
			Pl.	miklir	miklar	mikil
		Acc.	Sg.	miklan	mikla	mikiđ
			Pl.	mikla	miklar	mikil
		Dat.	Sg.	miklum	mikilli	miklu
			Pl.	miklum	miklum	miklum
		Gen.	Sg.	mikils	mikillar	mikils
			Pl.	mikilla	mikilla	mikilla
	Wk.	Nom.	Sg.	mikli	mikla	mikla
			Pl.	miklu	miklu	miklu
		Obl.	Sg.	mikla	miklu	mikla
			Pl.	miklu	miklu	miklu
Comp.			Sg.	meiri	meiri	meira
			Pl.	meiri	meiri	meiri
Su. St.		Nom.	Sg.	mestur	mest	mest
			Pl.	mestir	mestar	mest
		Acc.	Sg.	mestan	mesta	mest
			Pl.	mesta	mestar	mest
		Dat.	Sg.	mestum	mestri	mestu
			Pl.	mestum	mestum	mestum
		Gen.	Sg.	mests	mestrar	mests
			Pl.	mestra	mestra	mestra
	Wk.	Nom.	Sg.	mesti	mesta	mesta
			Pl.	mestu	mestu	mestu
		Obl.	Sg.	mesta	mestu	mesta
			Pl.	mestu	mestu	mestu

Obviously suppletive in comparative and superlative; less obvious is the irregular masculine accusative singular in -inn. The neuter singular nominative and accusative are formed by changing the final -l of the castem to -đ. This is regular for castems ending in -il-. One other word goes like mikill: lítill "small", but in the latter, the vowel shift +í:i applies after all instances of +iC:C:

líttill > -r:l -r +iC:C +í:i +a > litla .

3.2.1.10. Heiðinn. The last type of true adjective to be covered here is another example of the old suffix class of disyllabic adjectives:

heiðinn "heathen" (adjective)

				Masc.	Fem.	Neut.
Po. St.	Nom.	Sg.		heiðinn	heiðin	heiðið
		Pl.		heiðnir	heiðnar	heiðin
	Acc.	Sg.		heiðnan	heiðna	heiðið
		Pl.		heiðna	heiðnar	heiðin
	Dat.	Sg.		heiðnum	heiðinni	heiðnu
		Pl.		heiðnum	heiðnum	heiðnum
	Gen.	Sg.		heiðins	heiðinnar	heiðins
		Pl.		heiðinna	heiðinna	heiðinna
Wk.	Nom.	Sg.		heiðni	heiðna	heiðna
		Pl.		heiðnu	heiðnu	heiðnu
	Obl.	Sg.		heiðna	heiðnu	heiðna
		Pl.		heiðnu	heiðnu	heiðnu
Comp.		Sg.		heiðnari	heiðnari	heiðnara
		Pl.		heiðnari	heiðnari	heiðnari
Su. St.	Nom.	Sg.		heiðnastur	heiðnust	heiðnast
		Pl.		heiðnastir	heiðnastar	heiðnust
	Acc.	Sg.		heiðnastan	heiðnasta	heiðnast
		Pl.		heiðnasta	heiðnastar	heiðnust
	Dat.	Sg.		heiðnustum	heiðnastri	heiðnustu
		Pl.		heiðnustum	heiðnustum	heiðnustum
	Gen.	Sg.		heiðnasts	heiðnastrar	heiðnasts
		Pl.		heiðnastra	heiðnastra	heiðnastra
Wk.	Nom.	Sg.		heiðnasti	heiðnasta	heiðnasta
		Pl.		heiðnustu	heiðnustu	heiðnustu
	Obl.	Sg.		heiðnasta	heiðnustu	heiðnasta
		Pl.		heiðnustu	heiðnustu	heiðnustu

Shows syncope and doubling for the rstem. Note that the neuter nominative singular is formed by changing the final -n of the cstem to -ð, similar to the parallel procedure in mikill. Again, this is regular for cstems ending in -in.

3.2.1.11. talinn. This class, while similar in both form and function to the rest of the adjectives, is actually a verbal form--the past participle. Although the comparative and superlative forms might not be in use, it it possible to give them according to rule. In fact, these paradigms would probably be subnetworks in the verbal paradigms, derived by borrowing from the adjectives:

talinn "told" (adjective--past part. of telja "tell")

				Masc.	Fem.	Neut.
Po. St.		Nom.	Sg.	talinn	talin	talið
			Pl.	taldir	taldar	talin
		Acc.	Sg.	taldan	talda	talið
			Pl.	talda	taldar	talin
		Dat.	Sg.	töldum	talinni	töldu
			Pl.	töldum	töldum	töldum
		Gen.	Sg.	talins	talinnar	talins
			Pl.	talinna	talinna	talinna
	Wk.	Nom.	Sg.	taldi	talda	talda
			Pl.	töldu	töldu	töldu
		Obl.	Sg.	talda	töldu	talda
			Pl.	töldu	töldu	töldu
Comp.			Sg.	*taldari	*taldari	*taldara
			Pl.	*taldari	*taldari	*taldari
Su. St.		Nom.	Sg.	*taldastur	*töldust	*taldast
			Pl.	*taldastir	*taldastar	*töldust
		Acc.	Sg.	*taldastan	*taldasta	*taldast
			Pl.	*taldasta	*taldastar	*töldust
		Dat.	Sg.	*töldustum	*taldastri	*töldustu
			Pl.	*töldustum	*töldustum	*töldustum
		Gen.	Sg.	*taldasts	*taldastrar	*taldasts
			Pl.	*taldastra	*taldastra	*taldastra
	Wk.	Nom.	Sg.	*taldasti	*taldasta	*taldasta
			Pl.	*töldustu	*töldustu	*töldustu
		Obl.	Sg.	*taldasta	*töldustu	*taldasta
			Pl.	*töldustu	*töldustu	*töldustu

Note: comparative & superlative unattested.

This form is actually the past participle of telja, a weak verb. It is declined like heiðinn, but notice that the vstems have replaced -in- with d. Kress identifies this as being part of a process of dental assimilation. A similar alternation exists in the formation of neuter nominative singulars. The names for processes such as these are +lt:d̂, +in:d, and so on. The analysis of adjectives given here is incomplete, as it does not contain examples of each type of dental assimilation.

3.3. FPG Analysis of Adjectives. The preceding section presented examples of the principal adjective paradigms. In this section, networks are described which can ultimately be combined to form a complex FPG network for the examples. The plan of attack is to summarize the facts given above in terms of subnetworks which are internally transparent and which are shared by all the paradigms. When this is done, the other paradigms will be described essentially in terms of their similarities and differences with respect to these subnetworks.

In setting up the analysis, an implicit assumption is that stem-changing processes are more radical than suffixation. This seems intuitively correct, and it is consistent with Slobin's (1973) Operating Principle A: "Pay attention to the ends of words". However, it is still an arbitrary decision.

3.3.1. Stem relations. Although there are many stem forms used in the adjectives, there is a basic organization of these stems which is present in most of them. The most basic form, for our purposes here, will be the positive weak masculine singular oblique form. This form always ends in an a, so it will be called the aform. Removing the a from the aform results in the astem, which is abstract, as it may not be a surface form. From this astem the other basic stem forms are derived, using various rules depending on the paradigm, often including identity among all of the basic stem forms. A graphical representation of one basic stem relations is:

```
aform <--[+a]-- astem <----- castem -----> rstem
                  |            |
                  |            |
                  v            v
                ustem        custem
```

This form is most appropriate for syncopating forms, such as gamall or heiðinn. Note that the astem is derived from the castem. This corresponds with the notion that syncope is a process of deletion rather than of insertion. Some other adjectives would reverse that arc and derive castem from astem, for example nýr, which has an epenthetic j in astem but not in castem. The difference is not relevant to the data structures produced by the interpreter, only to the human reader of the grammar. It is important to remember that the castem is not 'the underlying form' for these adjectives. Instead, the whole network is present, and any form can be produced from any other form.

Some adjectives have a slightly different stem pattern, including i-shifted stem forms (only in comparative or superlative):

```
                istem -----> cistem
                  ^
                  |
                  |
aform <--[+a]-- astem -----> castem -----> rstem
                  |
                  |
                  v
                ustem -----> custem
```

An example with this pattern is fagur. Note that in this paradigm it is easier to derive custem (fögur) from ustem (fögr) than from castem (fagur). This is related to the problem of defining the +a:ö rule. As it is most easily set up so as to affect only the last vowel in the input string, it would complicate matters to have to skip over the epenthetic u in fagur.

3.3.2. Shared Subnetworks. Here are some relations in common among all adjectives which don't have to do with stem form changes per se:

```
DEFINE adj_equ
        EQUATE (po st fem nom sg) (po st neut nom pl)
               (po st neut acc pl)
        EQUATE (po st neut nom sg) (po st neut acc sg)
        EQUATE (po st masc gen sg) (po st neut gen sg)
        EQUATE (po st fem sg acc) (po st masc pl acc)
               (wk masc sg obl) (wk fem sg nom)
               (wk neut sg nom) (wk neut sg obl)
        EQUATE (po st fem pl nom) (po st fem pl acc)
        EQUATE (wk fem sg obl) (wk masc pl)
               (wk fem pl) (wk neut pl)
        EQUATE (po st masc sg dat) (po st masc pl dat)
               (po st fem pl dat) (po st neut pl dat)
        EQUATE (po st masc pl gen) (po st fem pl gen)
               (po st neut pl gen)
        EQUATE (comp masc sg) (comp fem sg) (comp masc pl)
               (comp fem pl) (comp neut pl)
```

The next body of shared material is a group of suffixes which is added to what we have labeled the astem. The FPG specification for this subnetwork is

```
DEFINE adj_anet
        DERIVE (po st fem sg acc) FROM astem VIA +a
        DERIVE (po st masc sg acc)
               FROM (po st fem sg acc) VIA +n
        DERIVE (po st fem pl nom)
               FROM (po st fem sg acc) VIA +r
        DERIVE (wk masc sg nom) FROM astem VIA +i
        DERIVE (po st masc pl nom)
               FROM (wk masc sg nom) VIA +r
```

Another closely related subnetwork which is common to all the regular paradigms is formed on the ustem and custem:

```
DEFINE adj_unet
        EQUATE custem (po st fem sg nom)
        DERIVE (wk fem sg obl) FROM ustem VIA +u
        DERIVE (po st masc sg dat)
                FROM (wk fem sg obl) VIA +m
```

If u-shift of the astem is not possible, astem and ustem will be equated. The third shared subnetwork consists of those categories formed by adding suffixes to the rstem:

```
DEFINE adj_rnet
        DERIVE (po st fem sg dat) FROM rstem VIA +i
        DERIVE (po st masc pl gen) FROM rstem VIA +a
        DERIVE (po st fem sg gen) FROM (po st masc pl gen) VIA +r
```

In some cases, namely for adjectives whose cstem ends in rr, the rstem categories are equated with one or both of the other two stems. There are two suffixes which are added to the comparative stem:

```
DEFINE adj_comp
        DERIVE (comp masc sg) FROM (compstem) VIA +i
        DERIVE (comp neut sg) FROM (compstem) VIA +a
```

As noted in the discussion of seinn above, it is sometimes the case that this comparative subnetwork is equated to the rstem network.

3.3.3. The Superlative: Nesting. Obviously, there is a great deal in common among all the paradigms in the superlative, but there is also a great deal in common between the superlative and the positive. In fact, with the exception of the possibility for forming additional levels of comparative or superlative, the superlative declension behaves just like a positive one with similar phonological shape. In particular, it uses the above subparadigms, attaching the remaining forms at various points.

There are two subnetworks in the superlative. Both of them can be borrowed from gulur:

```
DEFINE adj_sup
        BORROW po FROM gulur
                AND SPLIT BETWEEN astem ustem
                AND MAP po TO sup
                AND MAP astem TO supstem
                AND MAP ustem TO supustem
                AND DELETE castem rstem
```

This declaration must follow all declarations of positive forms for gulur. If there is no separate u-shifted superlative stem, then supstem and supustem are equated.

In summary, a speaker of Icelandic would know all of this information about formal relations without knowing anything except that some form is an adjective. No phonological or other lexical information is needed. In order to be able to refer to this information below, the following macro is defined:

```
DEFINE adj_shared
        INCLUDE adj_equ
        INCLUDE adj_anet
        INCLUDE adj_unet
        INCLUDE adj_rnet
        INCLUDE adj_comp
        INCLUDE adj_sup
```

3.3.4. Remaining Categories. These shared subnetworks, when combined in various ways, account for all but a few forms. These forms vary primarily due to phonological differences in the castem, from which they are formed.

The first form is masculine singular nominative. Its most common realization is derivation from the castem via +r +r:ur as in gulur and in the superlative. However, it can be identical with the rstem, as in seinn, mikill, and others when +l:r or +n:r are applicable; it can result from +r:ur, as in fagur, and can be identical with the castem, as in dýr.

The second of these forms is the masculine singular genitive. Its only variation results from the fact that two consecutive s's do not occur after a consonant, so nothing is added to stems ending in consonant+s.

The third varying form is the neuter singular nominative-accusative. This form involves the addition of a dental of some sort, plus various kinds of assimilation. Compare for example gulur ~ gult, hár ~ hátt, mikill ~ mikið, and the superlative, which has a stem ending in consonant+t.

Finally, there is the irregular masculine singular accusative in mikill.

3.3.5. A combined adjective network. The following combines all of the above subnetworks into a large bulged network, and adds information necessary to derive the stems and the additional categories.

```
BEGIN
        gulur svalur fagur hár nýr dýr
        seinn gamall mikill heiðinn talinn
--
        FOR gulur svalur
                DERIVE (po st masc nom sg)
                        FROM castem VIA +r +r:ur
        FOR fagur
                DERIVE (po st masc nom sg)
                        FROM castem VIA +r:ur
        FOR hár nýr
                DERIVE (po st masc nom sg)
                        FROM castem VIA +r
        FOR d´yr
                EQUATE (po st masc nom sg) castem
        FOR seinn hei´dinn talinn
                DERIVE (po st masc nom sg)
                        FROM castem VIA +r +r:n
        FOR gamall mikill
                DERIVE (po st masc nom sg)
                        FROM castem VIA +r +r:l
--
        FOR gulur svalur fagur dýr gamall
                DERIVE (po st neut nom sg)
                        FROM castem VIA +t
        FOR h´ar n´yr
                DERIVE (po st neut nom sg)
                        FROM castem VIA +tt
        FOR mikill
                DERIVE (po st neut nom sg)
                        FROM castem VIA +t +lt:ð
        FOR hei´dinn seinn talinn
                DERIVE (po st neut nom sg)
                        FROM castem VIA +t +nt:ð
--
        DERIVE (po st masc gen sg) FROM castem VIA +s
--
        FOR gulur svalur
                DERIVE rstem FROM castem VIA +r
        FOR fagur h´ar n´yr d´yr
                DERIVE rstem FROM (po masc nom sg) VIA +r
        FOR seinn gamall mikill hei´dinn talinn
                EQUATE rstem (po masc nom sg)
--
        FOR gulur hár nýr dýr seinn
                mikill heiðinn
                        EQUATE astem ustem
        FOR svalur talinn fagur
                DERIVE ustem FROM astem VIA +a:ö
        FOR gamall
                DERIVE ustem FROM astem VIA +a-a:ö-u
```

```
--
        FOR fagur
                DERIVE astem from castem VIA +r:ur
        FOR gulur svalur hár dýr seinn
                EQUATE astem castem
        FOR n´yr
                DERIVE astem FROM castem VIA +j
        FOR gamall
                DERIVE astem FROM castem VIA +aC:C
        FOR hei´dinn mikill
                DERIVE astem FROM castem VIA +iC:C
        FOR talinn
                DERIVE astem FROM castem VIA +in:d
--
        FOR gulur hár nýr dýr
                seinn mikill heiðinn
                        EQUATE custem castem
        FOR svalur gamall talinn
                DERIVE custem FROM castem VIA +a:ö
        FOR fagur DERIVE custem FROM ustem VIA +r:ur
--
        FOR gulur svalur dýr heiðinn *talinn
                DERIVE compstem FROM astem VIA +a +r
        FOR fagur
                EQUATE istem compstem
                AND DERIVE compstem FROM cistem VIA +r
        FOR hár
                DERIVE compstem FROM cistem VIA +rr
        FOR nýr dýr
                DERIVE compstem FROM castem VIA +rr
        FOR seinn
                EQUATE rstem compstem
        FOR gamall
                DERIVE compstem FROM astem VIA +gaml:eldr
        FOR mikill
                DERIVE compstem FROM astem VIA +mikl:meir
--
        FOR fagur DERIVE istem FROM astem VIA +a:e
        FOR hár DERIVE istem FROM astem VIA +á:æ
--
        FOR fagur DERIVE cistem FROM istem VIA +r:ur
        FOR hár DERIVE EQUATE istem cistem
--
        FOR gulur svalur fagur nýr dýr seinn
                heiðinn *talinn
                        DERIVE supstem FROM astem VIA +a +s +t
        FOR dýr
                DERIVE supstem FROM castem VIA +s +t
        FOR fagur hár
                DERIVE supstem FROM cistem VIA +s +t
        FOR gamall
```

```
            DERIVE supstem FROM astem VIA +gaml:elzt
      FOR mikill
            DERIVE supstem FROM astem VIA +mikl:mest
--
      FOR gulur fagur nýr dýr seinn heiðinn
            DERIVE supustem FROM supstem VIA +a:u
      FOR svalur talinn
            DERIVE supustem from supstem VIA +a-a:ö-u
      FOR fagur (Note) hár dýr (Note) gamall mikill
            EQUATE supstem supustem
      INCLUDE adj_shared
```

Note: it was realized at a late date that the treatment of abundance in fagur and dýr was ambiguous--this version actually would allow *fagrastum and *dýrastum, which are obviously not possible forms.

3.4. Noun Paradigms. There are 16 categories which I will include in the complete noun paradigm. These are: singular-plural number; nominative-accusative-dative-genitive case; and definite-unmarked determination. Because of the importance of root-vowel changes in the system, I will consistently use mnemonic labels for various stem forms which recur in various paradigms, even though for many paradigms these stem forms are undifferentiated. The distribution of these stem forms is similar to that for adjectives. For the more common nouns, there are 5 stems. The cstem is that stem form usually used for suffixes beginning with consonants; the vstem is the stem form usually used for suffixes beginning with vowels (but the vstem does not usually take suffixes directly); the astem is for suffixes beginning with -a; the istem is for suffixes beginning with -i; the ustem is for suffixes beginning with -u. In the simplest noun stem-patterns, all five are identical in form. However, in order to emphasize relatedness among paradigms or subparadigms which differ only in root changes, the arcs connecting even identical stem forms are most often best represented by this configuration:

```
                    = istem
cstem = vstem (
                    = astem = ustem .
```

The following paradigm definitions will take advantage of this structure, most often to introduce a root change, such as u-shift:

```
                    = istem
cstem = vstem (
                    = astem ---[+u-shift]---> ustem .
```

or to add some epenthetic element, such as pre-vocalic glide:

```
                              = istem
cstem ---[+j]---> vstem (
                              = astem = ustem
```

Occasionally the structure will be modified from this basic form, but when this happens, a special comment will be used to mark it. One last preliminary remark on the stem forms: although it is historically true that, for example, suffixing (e.g.) -i was associated with certain stem form modifications, this is no longer true in many cases, since many suffixes have been reduced and root changes morphologized.

One portion of the noun paradigms which can be considered as shared to a great extent is the definite endings. These endings are related historically to the postposed free definite article hinn. When it is postposed, certain morphological changes result, but there is relatively little variation from word to word within a gender. There is one of these which is shared among all nouns, regardless of gender and phonological shape:

```
DERIVE dpd FROM dp VIA -m +num
```

The genitive plural is also shared among all genders, but it has two variants:

```
DERIVE gpd FROM gp VIA +nna
DERIVE gpd FROM gp VIA -a +nna
```

There are actually only 14 of these patterns used in the nouns included here. They will be introduced and defined separately as macros.

Due to the fact that there is much more variation in the noun paradigms than in the adjectives, no attempt will be made to join together the paradigms as was done for the adjectives. Instead, each example will be treated independently as to its peculiarities, but the analysis will emphasize similarities among the individual paradigms. The nouns are grouped roughly by traditional formal classes and by gender, but this grouping is violated when convenient. To save space and make for easier reading and much easier typing, the category labels are abbreviated: ap means (acc pl nondef), etc. The nondefinite forms are left unmarked; definite forms end in ′d′: apd means (acc pl def). The italicized form in the left margin is the citation form, which here will always correspond to the nominative singular nondefinite (ns). In addition, all of the forms for each noun is given in four lines of comments in the following order:

```
; ns as ds gs
; np ap dp gp
; nsd asd dsd gsd
; npd apd dpd gpd
```

This definition is the most common masculine definite forms:

```
DEFINE def_hestur
    DERIVE nsd FROM ns VIA +inn
    DERIVE asd FROM as VIA +inn
    DERIVE dsd FROM ds VIA +num
    DERIVE gsd FROM gs VIA +ins
    DERIVE npd FROM np VIA +nir
    DERIVE apd FROM ap VIA +na
    DERIVE dpd FROM dp VIA -m +num
    DERIVE gpd FROM gp VIA +nna

BEGIN hestur "horse" M
    ; hestur hest hesti hests
    ; hestar hesta hestum hesta
    ; hesturinn hestinn hestinum hestsins
    ; hestarnir hestana hestunum hestanna
    EQUATE vstem cstem istem astem
    EQUATE astem ustem
    DERIVE ns FROM cstem VIA +u +r
    DERIVE ds FROM istem VIA +i
    DERIVE gs FROM cstem VIA +s
    DERIVE ap FROM astem VIA +a
    DERIVE np FROM ap VIA +r
    DERIVE dp FROM astem VIA +u +m (Note 1)
    DERIVE gp FROM astem VIA +a (Note 2)
    INCLUDE def_hestur
```

Note 1: dp is split into +u +m because the +num of the definite suffix regularly replaces +m. Note 2: gp merges with ap here, but for later paradigms needs to be kept distinct.

```
BEGIN hattur "hat" M
    ; hattur hatt hatti hatts
    ; hattar hatta höttum hatta
    ; hatturinn hattinn hattinum hattsins
    ; hattarnir hattana höttunum hattanna
    BORROW ALL FROM hestur
            AND SPLIT BETWEEN astem ustem
    DERIVE ustem FROM astem VIA +a:ö (Note)
```

Note: This rule effects only dp and dpd.

```
BEGIN mór "peat, heath" M
    ; mór mó mó mós
    ; móar móa móum móa
    ; mórinn móinn mónum mósins
    ; móarnir móana móunum móanna
    BORROW ALL nondef BUT ns ds FROM hestur
    EQUATE istem ds
    DERIVE ns FROM cstem VIA +r
    INCLUDE def_hestur

DEFINE def_skór
    DERIVE nsd FROM ns VIA +inn
    DERIVE asd FROM as VIA +inn
    DERIVE dsd FROM ds VIA +num
    DERIVE gsd FROM gs VIA +ins
    DERIVE npd FROM np VIA +nir
    DERIVE apd FROM ap VIA +na
    DERIVE dpd FROM dp VIA -m +num
    DERIVE gpd FROM gp VIA -a +nna

BEGIN skór "shoe" M
    ; skór skó skó skós
    ; skór skó skóm skóa
    ; skórinn skóinn skónum skósins
    ; skórnir skóna skónum skónna
    BORROW ALL FROM mör EXCEPT dp as np
    EQUATE as ap
    EQUATE ns np
    DERIVE dp FROM cstem VIA +m
    INCLUDE def_skór

BEGIN söngur "song" M
    ; söngur söng söng(vi) söngs
    ; söngvar söngva söngvum söngva
    ; söngurinn sönginn söng(vi)num söngsins
    ; söngvarnir söngvana söngvunum söngvanna
    BORROW ALL FROM hestur
            AND SPLIT BETWEEN cstem vstem
    DERIVE vstem FROM cstem VIA +v
    EQUATE as ds (Note)
```

Note: söngur has two forms for ds: söng and söngvi.

```
BEGIN læknir "doctor" M
    ; læknir lækni lækni læknis
    ; læknar lækna læknum lækna
    ; læknirinn lækninn lækninum læknisins
    ; læknarnir læknana læknunum læknanna
```

```
    BORROW ALL FROM mör
            AND SPLIT BETWEEN cstem vstem
    DERIVE cstem FROM vstem VIA +i
```

Note: there is an alternate paradigm for <u>læknir</u> which goes like <u>hver</u>, below (except for the double plurals).

```
BEGIN hver "geyser" M
    ; hver hver hver hvers
    ; hverar hvera hverum hvera
    ; hverinn hverinn hvernum hversins
    ; hverarnir hverana hverunum hveranna
    BORROW ALL FROM hestur BUT ns ds
    EQUATE cstem ns ds
    BORROW istem ap np FROM smiður (Note)
```

Note: There are alternate plural forms for <u>hver</u>.

```
BEGIN stóll "chair" M
    ; stóll stól stóli stóls
    ; stólar stóla stólum stóla
    ; stóllinn stólinn stólinum stólsins
    ; stólarnir stólana stólunum stólanna
    BORROW ALL nondef BUT ns FROM hestur
    DERIVE ns FROM cstem VIA +r +r:n
    INCLUDE def_hestur

BEGIN himinn "heaven" M
    ; himinn himin himni himins
    ; himnar himna himnum himna
    ; himinninn himininn himninum himinsins
    ; himnarnir himnana himnunum himnanna
    BORROW ALL nondef FROM hestur BUT ns
            AND SPLIT BETWEEN vstem cstem
    BORROW ns FROM stóll
    DERIVE vstem FROM cstem VIA +iC:C
    INCLUDE def_hestur

BEGIN akur "field" M
    ; akur akur akri akurs
    ; akrar akra ökrum akra
    ; akurinn akurinn akrinum akursins
    ; akrarnir akrana ökrunum akranna
    BORROW ALL nondef FROM hattur BUT ds ns
            AND SPLIT BETWEEN vstem cstem
    EQUATE cstem ns
```

```
    DERIVE cstem FROM vstem VIA +r:ur
    INCLUDE def_hestur

BEGIN jökull "glacier" M
    ; jökull jökul jökli jökuls
    ; jöklar jökla jöklum jökla
    ; jökullinn jökulinn jöklinum jökulsins
    ; jöklarnir jöklana jöklunum jöklanna
    BORROW ALL FROM himinn
            AND SPLIT BETWEEN vstem cstem
    DERIVE vstem FROM cstem VIA +uC:C

BEGIN ás "ace" M
    ; ás ás ás áss
    ; ásar ása ásum ása
    ; ásinn ásinn ásnum ássins
    ; ásarnir ásana ásunum ásanna
    BORROW ALL nondef FROM hver BUT ds
    BORROW as ds FROM hestur
    INCLUDE def_hestur

BEGIN karl "(old) man" M
    ; karl karl karl karls
    ; karlar karla körlum karla
    ; karlinn karlinn karlnum karlsins
    ; karlarnir karlana körlunum karlanna
    BORROW ALL nondef FROM ás BUT ustem dp
    BORROW astem ustem dp FROM hattur
    INCLUDE def_hestur

BEGIN dagur "day" M
    ; dagur dag degi dags
    ; dagar daga dögum daga
    ; dagurinn daginn deginum dagsins
    ; dagarnir dagana dögunum daganna
    BORROW ALL FROM hattur
            AND SPLIT BETWEEN vstem istem
    DERIVE istem FROM vstem VIA +a:e

BEGIN gaffall "fork" M
    ; gaffall gaffal gaffli gaffals
    ; gafflar gaffla göfflum gaffla
    ; gaffallinn gaffalinn gafflinum gaffalsins
    ; gafflarnir gafflana göfflunum gafflanna
    BORROW ALL FROM himinn
            AND SPLIT BETWEEN vstem AND cstem
    DERIVE vstem FROM cstem VIA +aC:C
    SPLIT BETWEEN astem ustem
    BORROW astem ustem dp FROM hattur
```

```
BEGIN ketill "kettle" M
    ; ketill ketil katli ketils
    ; katlar katla kötlum katla
    ; ketillinn ketilinn katlinum ketilsins
    ; katlarnir katlana kötlunum katlanna
    BORROW ALL FROM gaffall
            AND SPLIT BETWEEN cstem AND vstem
    DERIVE vstem FROM cstem VIA +iC:C -a:e

BEGIN kristall "crystal" M
    ; kristall kristall kristalli kristals
    ; kristallar kristalla kristöllum kristalla
    ; kristallinn kristallinn kristallinum kristalsins
    ; kristallarnir kristallana kristöllunum kristallanna
    BORROW nondef ALL BUT gs FROM karl
    DERIVE gs FROM cstem VIA -l +s (Note)
    INCLUDE def_hestur
```

Note: kristall is obviously a borrowed word, and its cstem is -ll, not -l as for gaffall. However, the second -l is removed before the -s of gs.

```
BEGIN skógur "woods, birch copse" M
    ; skógur skóg skógi skógs,skógar
    ; skógar skóga skógum skóga
    ; skógurinn skóginn skóginum skógsins,skógarins
    ; skógarnir skógana skógunum skóganna
    BORROW ALL FROM hestur
    EQUATE gs np (Note)
```

Note: alternate gs form. Alternate gsd botched.

```
BEGIN grautur "porridge, cereal" M
    ; grautur graut grauti grautar
    ; grautar grauta grautum grauta
    ; grauturinn grautinn grautinum grautarins
    ; grautarnir grautana grautunum grautanna
    BORROW nondef ALL FROM hestur BUT gs
    EQUATE np gs
    INCLUDE def_hestur

BEGIN bær "farm" M
    ; bær bæ bæ bæjar
    ; bæir bæi bæjum bæja
    ; bærinn bæinn bænum bæjarins
```

```
    ; bæirnir bæina bæjunum bæjanna
    BORROW ALL nondef FROM veggur BUT ns
    DERIVE ns FROM cstem VIA +r
    INCLUDE def_hestur

BEGIN ljár "scythe" M
    ; ljár ljá ljá ljáar
    ; ljáir ljái ljá(u)m ljáa
    ; ljárinn ljáinn ljánum ljáarins
    ; ljáirnir ljáina ljá(u)num ljánna
    BORROW nondef ALL BUT ns FROM smiður
    DERIVE ns FROM cstem VIA +r
    INCLUDE def_skór
```

Note: also blær "soft wind, breeze"

```
BEGIN guð "god" M
    ; guð guð guð guðs
    ; guðir guði guðum guða
    ; guðinn guðinn guðnum guðsins
    ; guðirnir guðina guðunum guðanna
    BORROW ALL nondef BUT ns FROM smiður
    EQUATE cstem ns
    INCLUDE def_hestur
```

Note: also gris "pig", her "army".

```
BEGIN veggur "wall" M
    ; veggur vegg vegg veggjar
    ; veggir veggi veggjum veggja
    ; vegggurinn vegginn veggnum veggjarins
    ; veggirnir veggina veggjunum veggjanna
    BORROW ALL FROM hlutur
            AND SPLIT BETWEEN cstem AND vstem
    DERIVE vstem FROM cstem VIA +j
    BORROW ALL FROM hlutur
            AND SPLIT BETWEEN vstem AND astem (Note)
    DERIVE astem FROM vstem VIA +j (Note)
```

Note: these forms are consistent with the orthography, which regularly reduces +ji to +i after k- or g-. The other forms are consistent with the phonology, where the /j/ is pronounced throughout. This occurs in various other forms; from here on we'll go with the phonology when there is a difference.

```
BEGIN gestur "guest" M
    ; gestur gest gesti gests
    ; gestir gesti gestum gesta
    ; gesturinn gestinn gestinum gestsins
    ; gestirnir gestina gestunum gestanna
    BORROW ALL nondef FROM hestur
            AND SPLIT BETWEEN astem ap
    DERIVE ap FROM istem VIA +i
    INCLUDE def_hestur

BEGIN smiður "smith" M
    ; smiður smið smið smiðs
    ; smiðir smiði smiðum smiða
    ; smiðurinn smiðinn smiðnum smiðsins
    ; smiðirnir smiðina smiðunum smiðanna
    BORROW ALL nondef FROM gestur BUT ds
    EQUATE as ds
    INCLUDE def_hestur

BEGIN leikur "play" M
    ; leikur leik leik leiks
    ; leikir leiki leikjum leikja
    ; leikurinn leikinn leiknum leiksins
    ; leikirnir leikina leikjunum leikjanna
    BORROW ALL FROM smiður
            AND SPLIT BETWEEN cstem vstem
    DERIVE vstem cstem VIA +j

BEGIN dalur "valley" M
    ; dalur dal dal dals
    ; dalir dali dölum dala
    ; dalurinn dalinn dalnum dalsins
    ; dalirnir dalina dölunum dalanna
    BORROW ALL FROM smiður
            AND SPLIT BETWEEN astem ustem
    BORROW ustem astem FROM hattur


BEGIN staður "place" M
    ; staður stað stað staðar
    ; staðir staði stöðum staða
    ; staðurinn staðinn staðnum staðarins
    ; staðirnir staðina stöðunum staðanna
    BORROW ALL FROM hlutur
            AND SPLIT BETWEEN astem ustem
    DERIVE ustem FROM astem VIA +a:ö
```

```
BEGIN hlutur "thing" M
     ; hlutur hlut hlut hlutar
     ; hlutir hluti hlutum hluta
     ; hluturinn hlutinn hlutnum hlutarins
     ; hlutirnir hlutina hlutunum hlutanna
     BORROW ALL nondef BUT gs FROM smiður
     DERIVE gs FROM astem FROM +a +r
     INCLUDE def_hestur

BEGIN fatnaður "clothing" M
     ; fatnaður fatnað fatnaði fatnaðar
     ; fatnaðir fatnaði fötnuðum fatnaða
     ; fatnaðurinn fatnaðinn fatnaðinum fatnaðarins
     ; fatnaðirnir fatnaðina fötnuðunum fatnaðanna
     BORROW ALL nondef FROM staður BUT ds
             AND SPLIT BETWEEN astem ustem
     DERIVE ustem FROM astem VIA +a-a:ö:u
     EQUATE ap ds
     INCLUDE def_hestur

BEGIN söfnuður "congregation" M
     ; söfnuður söfnuð söfnuði safnaðar
     ; söfnuðir söfnuði söfnuðum safnaða
     ; söfnuðurinn söfnuðinn söfnuðinum safnaðarins
     ; söfnuðirnir söfnuðina söfnuðunum safnaðanna
     BORROW ALL FROM fatnaður
             AND SPLIT BETWEEN astem vstem (Note)
     EQUATE vstem ustem
```

Note: this alters the basic pattern of relations among stem forms given above; it results in a pattern like:

```
                        =  istem
    cstem = vstem (
                        = ustem <--[+u-shift]-- astem
```

so that all but astem forms are u-shifted.

```
BEGIN köttur "cat" M
     ; köttur kött ketti kattar
     ; kettir ketti köttum katta
     ; kötturinn köttinn kettinum kattarins
     ; kettirnir kettina köttunum kattanna
     BORROW ALL FROM fatnaður
             AND SPLIT BETWEEN vstem istem
             AND BETWEEN vstem cstem
             AND BETWEEN astem ustem
```

```
    DERIVE ustem FROM astem VIA +a:ö
    DERIVE istem FROM vstem VIA +a:e
    EQUATE cstem ustem (Note)
```

Note: this is another modification of the basic stem configuration; both ustem and cstem forms are u-shifted if possible. This configuration is the basis for the next few paradigms as well.

```
BEGIN fjörður "fjord" M
    ; fjörður fjörð firði fjarðar
    ; firðir firði fjörðum fjarða
    ; fjörðurinn fjörðinn firðinum fjarðarins
    ; firðirnir firðina fjörðunum fjarðanna
    BORROW ALL FROM köttur
            AND SPLIT BETWEEN vstem astem
    DERIVE astem FROM vstem VIA +breaking

BEGIN háttur "mode" M
    ; háttur hátt hætti háttar
    ; hættir hætti háttum hátta
    ; hátturinn háttinn hættinum háttarins
    ; hættirnir hættina háttunum háttanna
    BORROW ALL FROM köttur
            AND SPLIT BETWEEN ustem astem
    EQUATE astem ustem

BEGIN spónn "spoon" M
    ; spónn spón spæni spóns
    ; spænir spæni spónum spóna
    ; spónninn spóninn spæninum spónsins
    ; spænirnir spænina spónunum spónanna
    BORROW ALL nondef FROM háttur BUT ns
    BORROW cstem ns FROM stóll
    INCLUDE def_hestur

BEGIN sonur "son" M
    ; sonur son syni sonar
    ; synir syni sonum sona
    ; sonurinn soninn syninum sonarins
    ; synirnir synina sonunum sonanna
    BORROW ALL nondef FROM  háttur BUT gs
            AND SPLIT BETWEEN vstem istem
    BORROW astem gs FROM hlutur
    DERIVE istem FROM vstem VIA +o:y
    INCLUDE def_hestur
```

```
DEFINE def maður
    DERIVE nsd FROM ns VIA +inn
    DERIVE asd FROM as VIA +inn
    DERIVE dsd FROM ds VIA +num
    DERIVE gsd FROM gs VIA +ins
    DERIVE npd FROM np VIA +i +r +nir
    DERIVE apd FROM ap VIA +i +na
    DERIVE dpd FROM dp VIA -m +num
    DERIVE gpd FROM gp VIA +nna

BEGIN maður "man" M
    ; maður mann manni manns
    ; menn menn mönnum manna
    ; madurinn manninn manninum mannsins
    ; mennirnir mennina mönnunum mannanna
    EQUATE vstem astem
    EQUATE vstem cstem
    DERIVE istem FROM vstem VIA +a:e
    DERIVE ustem FROM astem VIA +a:ö
    EQUATE as cstem
    EQUATE istem ap np (Note 3)
    DERIVE ns FROM cstem VIA +r +nnr:ðr +r:ur (Note 2)
    BORROW astem gp ustem dp FROM hattur
    BORROW cstem gs FROM hestur
    INCLUDE def_maður
```

Note 1: maður is different enough to spell out certain relations which might otherwise be expressed as borrowings. Note 2: this rule (mannr -> maðr) is idiosyncratic. Note 3: this is unusual in that the npd and apd are regular in shape while np and ap have merged with the istem.

```
BEGIN fótur "foot" M
    ; fótur fót fæti fótar
    ; fætur fætur fótum fóta
    ; fóturinn fótinn fætinum fótarins
    ; fæturnir fæturna fótunum fótanna
    BORROW ALL nondef FROM maður EXCEPT ns ds ap np
           AND SPLIT BETWEEN astem ustem
    EQUATE astem ustem
    BORROW cstem ns FROM háttur
    BORROW istem ds FROM háttur
    DERIVE np ap FROM istem VIA +u +r
    INCLUDE def_hestur
```

```
BEGIN faðir "father" M
     ; faðir føður føður føður
     ; feður feður feðrum feðra
     ; faðirinn føðurinn føðurnum føðurins
     ; feðurnir feðurna feðrunum feðranna
     BORROW vstem astem ustem FROM hattur
     DERIVE ns FROM astem VIA +r:ir (Note 2)
     DERIVE cstem FROM ustem VIA +r:ur
     EQUATE cstem as ds gs
     BORROW vstem istem FROM køttur
     DERIVE ap np FROM istem VIA +r:ur
     DERIVE gp FROM istem VIA +a (Note 3)
     DERIVE dp FROM istem VIA +u +m (Note 3)
     INCLUDE def_hestur
```

Note 1: the three family nouns faðir, broðir, and moðir share the same inflections (except moðir, being feminine, uses standard feminine definites). Also, the root structure is very unusual because of the two stem vowels -i- and -u-. Note 2: the i epenthesis rule is unique to these nouns. Note 3: adding gp and dp to istem rather than to astem and ustem is another unusual aspect of these nouns.

```
BEGIN fingur "finger" M
     ; fingur fingur fingri fingurs
     ; fingur fingur fingrum fingra
     ; fingurinn fingurinn fingrinum fingursins
     ; fingurnir fingurna fingrunum fingranna
     BORROW ALL nondef BUT ds ns gs FROM faðir
             AND SPLIT BETWEEN astem AND ustem
                     AND BETWEEN istem AND vstem
     EQUATE astem ustem
     EQUATE vstem istem
     EQUATE ns as
     BORROW istem ds FROM hestur
     BORROW cstem gs FROM hestur
     INCLUDE def_hestur

BEGIN vetur "winter" M
     ; vetur vetur vetri vetrar
     ; vetur vetur vetrum vetra
     ; veturinn veturinn vetrinum vetrarins
     ; veturnir veturna vetrunum vetranna
     BORROW ALL nondef BUT gs FROM fingur
     DERIVE gs FROM gp VIA +r
```

```
    INCLUDE def_hestur

DEFINE def_kinn
    DERIVE nsd FROM ns VIA +in
    DERIVE asd FROM as VIA +ina
    DERIVE dsd FROM ds VIA +inni
    DERIVE gsd FROM gs VIA +innar
    DERIVE npd FROM np VIA +nar
    DERIVE apd FROM ap VIA +nar
    DERIVE dpd FROM dp VIA -m +num
    DERIVE gpd FROM gp VIA +nna

BEGIN kinn "cheek" F
    ; kinn kinn kinn kinnar
    ; kinnar kinnar kinnum kinna
    ; kinnin kinnina kinninni kinnarinnar
    ; kinnarnar kinnarnar kinnunum kinnanna
    BORROW vstem istem astem ustem gp dp
            FROM hestur
    EQUATE cstem ustem (Note 1)
    EQUATE cstem ns as ds
    DERIVE gs np ap FROM gp VIA +r
    INCLUDE def_kinn
```

Note: most feminines show u-shift when possible in what would be called cstem for masculines.

```
BEGIN tíð "time" F
    ; tíð tíð tíð tíðar
    ; tíðir tíðir tíðum tíða
    ; tíðin tíðina tíðinni tíðarinnar
    ; tíðirnar tíðirnar tíðunum tíðanna
    BORROW ALL nondef FROM kinn BUT np ap
    DERIVE np ap FROM istem VIA +i +r
    INCLUDE def_kinn

BEGIN skel "shell" F
    ; skel skel skel skeljar
    ; skeljar skeljar skeljum skelja
    ; skelin skelina skelinni skeljarinnar
    ; skeljarnar skeljarnar skeljunum skeljanna
    BORROW ALL FROM kinn
            AND SPLIT BETWEEN cstem ustem
    DERIVE ustem FROM cstem VIA +j (Note)
```

Note: all of the vowel-stems have the -j-, which is dropped if no vowel follows.

```
BEGIN stöð "station" F
    ; stöð stöð stöð stöðvar
    ; stöðvar stöðvar stöðvum stöðva
    ; stöðin stöðina stöðinni stöðvarinnar
    ; stöðvarnar stöðvarnar stöðvunum stöðvanna
    BORROW ALL FROM kinn
            AND SPLIT BETWEEN cstem ustem
    DERIVE ustem FROM cstem VIA +v (Note)
```

Note: compare skel above.

```
DEFINE def_kerling
    DERIVE nsd FROM ns VIA +in
    DERIVE asd FROM as VIA +na
    DERIVE dsd FROM ds VIA +nni
    DERIVE gsd FROM gs VIA +nnar
    DERIVE npd FROM np VIA +nar
    DERIVE apd FROM ap VIA +nar
    DERIVE dpd FROM dp VIA -m +num
    DERIVE gpd FROM gp VIA +nna
```

```
BEGIN Þórey "pers. name" F
    ; Þórey Þóreyju Þóreyju Þóreyjar
    ; Þóreyjar Þóreyjar Þóreyjum Þóreyja
    ; Þóreyin Þóreyjuna Þóreyjunni Þóreyjarinnar
    ; Þóreyjarnar Þóreyjarnar Þóreyjunum Þóreyjanna
    BORROW ALL nondef FROM skel BUT as ds
    DERIVE as ds FROM ustem VIA +u
    INCLUDE def_kerling
```

Note: also Guðný, also a personal name.

```
BEGIN kerling "old woman" F
    ; kerling kerlingu kerlingu kerlingar
    ; kerlingar kerlingar kerlingum kerlinga
    ; kerlingin kerlinguna kerlingunni kerlingarinnar
    ; kerlingarnar kerlingarnar kerlingunum kerlinganna
    BORROW ALL FROM kinn BUT as ds
    DERIVE as ds FROM ustem VIA +u
    INCLUDE def_kerling
```

Note: words in -ing, an old diminutive suffix.

```
DEFINE def_á
    DERIVE nsd FROM ns VIA +in
    DERIVE asd FROM as VIA +na
    DERIVE dsd FROM ds VIA +nni
    DERIVE gsd FROM gs VIA +innar
    DERIVE npd FROM np VIA +nar
    DERIVE apd FROM ap VIA +nar
    DERIVE dpd FROM dp VIA -m +num
    DERIVE gpd FROM gp VIA -a +nna

BEGIN á "river" F
    ; á á á ár
    ; ár ár ám áa
    ; áin ána ánni árinnar
    ; árnar árnar ánum ánna
    BORROW ns as ds gp FROM kinn
    DERIVE gs np ap FROM cstem VIA +r
    BORROW dp cstem FROM skór
    BORROW gp dp FROM skór
    INCLUDE def_á

BEGIN frú "Mrs." F
    ; frú frú frú frúar
    ; frúr frúr frúm frúa
    ; frúin frúna frúnni frúarinnar
    ; frúrnar frúrnar frúnum frúnna
    BORROW ALL FROM á BUT gs gsd
    BORROW gs gsd gp gpd FROM kinn

BEGIN alin "ell" F
    ; alin alin alin álnar
    ; álnir álnir álnum álna
    ; alinin alinina alininni álnarinnar
    ; álnirnar álnirnar álnunum álnanna
    BORROW ALL FROM tíð
            AND SPLIT BETWEEN cstem ustem
    DERIVE ustem FROM cstem VIA +iC:C +áln:aln (Note)

Note: idiosyncratic rule; no other examples.


BEGIN heiði "heath" F
    ; heiði heiði heiði heiðar
    ; heiðar heiðar heiðum heiða
    ; heiðin heiðina heiðinni heiðarinnar
    ; heiðarnar heiðarnar heiðunum heiðanna
    BORROW ALL nondef FROM kinn BUT ns as ds
    DERIVE ns as ds FROM ustem VIA +i
    INCLUDE def_kerling
```

```
BEGIN Hildur "pers. name" F
    ; Hildur Hildi Hildi Hildar
    ; Hildar Hildar Hildum Hilda
    ; Hildurin Hildina Hildinni Hildarinnar
    ; Hildarnar Hildarnar Hildunum Hildanna
    BORROW ALL FROM heiði BUT ns nsd
    BORROW cstem ns FROM hestur
    BORROW ns nsd FROM kinn


DEFINE def_djörfung
    DERIVE nsd FROM ns VIA +in
    DERIVE asd FROM as VIA +ina
    DERIVE dsd FROM ds VIA +nni
    DERIVE gsd FROM gs VIA +innar
    DERIVE npd FROM np VIA +nar
    DERIVE apd FROM ap VIA +nar
    DERIVE dpd FROM dp VIA -m +num
    DERIVE gpd FROM gp VIA +nna

BEGIN djörfung "courage" F
    ; djörfung djörfung djörfungu djörfungar
    ; djörfungar djörfungar djörfungum djörfunga
    ; djörfungin djörfungina djörfungunni djörfungarinnar
    ; djörfungarnar djörfungarnar djörfungunum djörfunganna
    BORROW ALL FROM kerling BUT as
    EQUATE as ns
    INCLUDE def_djörfung

DEFINE def_lifur
    DERIVE nsd FROM astem VIA +in
    DERIVE asd FROM astem VIA +ina
    DERIVE dsd FROM astem VIA +nni
    DERIVE gsd FROM gs VIA +innar
    DERIVE npd FROM np VIA +nar
    DERIVE apd FROM ap VIA +nar
    DERIVE dpd FROM dp VIA -m +num
    DERIVE gpd FROM gp VIA +nna

BEGIN lifur "liver" F
    ; lifur lifur lifur lifrar
    ; lifrar lifrar lifrum lifra
    ; lifrin lifrina lifrinni lifrarinnar
    ; lifrarnar lifrarnar lifrunum lifranna
    BORROW ALL nondef FROM kinn
            AND SPLIT BETWEEN ustem cstem
    DERIVE cstem FROM ustem VIA +r:ur
    INCLUDE def_lifur
```

```
BEGIN verzlun "trade" F
    ; verzlun verzlun verzlun verzlunar
    ; verzlanir verzlanir verzlunum verzlana
    ; verzlunin verzlunina verzluninni verzlunarinnar
    ; verzlanirnar verzlanirnar verzlununum verzlananna
    BORROW ALL nondef FROM tíð BUT gs
           AND SPLIT BETWEEN astem AND ustem
    DERIVE ustem FROM astem VIA +a:u
    DERIVE gs FROM ustem VIA +a +r
    INCLUDE def_kinn

BEGIN pöntun "order" F
    ; pöntun pöntun pöntun pöntunar
    ; pantanir pantanir pöntunum pantana
    ; pöntunin pöntunina pöntuninni pöntunarinnar
    ; pantanirnar pantanirnar pöntununum pantananna
    BORROW ALL FROM verzlun
           AND SPLIT BETWEEN astem ustem
    DERIVE ustem FROM astem VIA +a-a:ö-u

BEGIN gjöf "gift" F
    ; gjöf gjöf gjöf gjafar
    ; gjafir gjafir gjöfum gjafa
    ; gjöfin gjöfina gjöfinni gjafarinnar
    ; gjafirnar gjafirnar gjöfunum gjafanna
    BORROW ALL FROM tíð
           AND SPLIT BETWEEN astem ustem
    DERIVE ustem FROM astem VIA +a:ö

BEGIN sól "sun" F
    ; sól sólu sólu sólar
    ; sólir sólir sólum sóla
    ; sólin sóluna sólunni sólarinnar
    ; sólirnar sólirnar sólunum sólanna
    BORROW ALL nondef FROM tíð BUT as ds
    DERIVE as ds FROM ustem VIA +u
    INCLUDE def_kinn

BEGIN steik "roast" F
    ; steik steik steik steikar,steikur
    ; steikur steikur steikum steika
    ; steikin steikina steikinni steikarinnar,steikurinnar
    ; steikurinnar steikurinnar steikunum steikanna
    BORROW ALL nondef FROM tíð BUT ap np
    DERIVE gs ap np FROM istem VIA +r +r:ur (Note)
    INCLUDE def_kinn
```

Note: two gs forms (steikur ~ steikar).

```
BEGIN mörk "pint" F
    ; mörk mörk mörk markar,merkur
    ; merkur merkur mörkum marka
    ; mörkin mörkina mörkinni markarinnar,merkurinnar
    ; merkurnar merkurnar mörkunum markanna
    BORROW ALL FROM steik
            AND SPLIT BETWEEN istem vstem
            AND SPLIT BETWEEN astem ustem
    DERIVE istem FROM vstem VIA +a:e
    DERIVE ustem FROM astem VIA +a:ö

BEGIN bók "book" F
    ; bók bók bók bókar
    ; bækur bækur bókum bóka
    ; bókin bókina bókinni bókarinnar
    ; bækurnar bækurnar bókunum bókanna
    BORROW ALL nondef FROM steik BUT gs
            AND SPLIT BETWEEN istem vstem
    DERIVE istem FROM vstem VIA +ó:æ
    DERIVE gs FROM gp VIA +r
    INCLUDE def_kinn

BEGIN nótt "night" F
    ; nótt nótt nótt nætur
    ; nætur nætur nóttum nótta
    ; nóttin nóttina nóttinni næturinnar
    ; næturnar næturnar nóttunum nóttanna
    BORROW ALL nondef FROM bók EXCEPT np ap
    DERIVE np ap FROM istem VIA -t +r +r:ur
    INCLUDE def_kinn
```

Note: geminate consonant simplification (cf. kristall above).

```
BEGIN brú "bridge" F
    ; brú brú brú brúar
    ; brýr brýr brúm brúa
    ; brúin brúna brúnni brúarinnar
    ; brýrnar brýrnar brúnum brúnna
    BORROW ALL nondef BUT np ap dp FROM bók
    DERIVE np ap FROM istem VIA +r
    BORROW dp cstem FROM skór
    INCLUDE def_á

BEGIN kýr "cow" F
    ; kýr kú kú kýr
    ; kýr kýr kúm kúa
    ; kýrin kúna kúnni kýrinnar
```

```
    ; kýrnar kýrnar kúnum kúnna
    BORROW ALL nondef FROM brú BUT ns gs
    EQUATE ns gs np
    INCLUDE def_á

BEGIN móðir "mother" F
    ; móðir móður móður móður
    ; mæður mæður mæðrum mæðra
    ; móðirin móðurina móðurinni móðurinnar
    ; mæðurnar mæðurnar mæðrunum mæðranna
    BORROW ALL nondef FROM bróðir
    INCLUDE def_kinn

BEGIN mús "mouse" F
    ; mús mús mús músar
    ; mýs mýs músum músa
    ; músin músina músinni músarinnar
    ; mýsnar mýsnar músunum músanna
    BORROW ALL nondef BUT np ap FROM bók
    EQUATE ns np ap
    INCLUDE def_kinn


DEFINE def_kyn
    DERIVE nsd FROM ns VIA +ið
    DERIVE asd FROM as VIA +ið
    DERIVE dsd FROM ds VIA +nu
    DERIVE gsd FROM gs VIA +ins
    DERIVE npd FROM np VIA +in
    DERIVE apd FROM ap VIA +in
    DERIVE dpd FROM dp VIA -m +num
    DERIVE gpd FROM gp VIA +nna

BEGIN kyn "kin" N
    ; kyn kyn kyni kyns
    ; kyn kyn kynjum kynja
    ; kynið kynið kyninu kynsins
    ; kynin kynin kynjunum kynjanna
    BORROW ALL nondef BUT ds ns np ap FROM leikur
    EQUATE ns as
    BORROW ds istem FROM hestur
    BORROW ustem np ap FROM borð
    INCLUDE def_kyn


DEFINE def_tré
    DERIVE nsd FROM ns VIA +ð
    DERIVE asd FROM as VIA +ð
    DERIVE dsd FROM ds VIA +nu
```

```
    DERIVE gsd FROM gs VIA +ins
    DERIVE npd FROM np VIA +n
    DERIVE apd FROM ap VIA +n
    DERIVE dpd FROM dp VIA -m +num
    DERIVE gpd FROM gp VIA +nna

BEGIN tré "tree" N
    ; tré tré tré trés
    ; tré tré trjám trjá
    ; tréð tréð trénu trésins
    ; trén trén trjánum trjánna
    EQUATE cstem ns as ds np ap
    BORROW cstem gs FROM hestur
    DERIVE astem FROM cstem VIA +é:á (Note)
    BORROW astem ustem dp gp FROM skór
    INCLUDE def_tré
```

Note: unusual rule (<u>é</u> -> <u>já</u>) also found in: <u>hné</u> "knee", <u>hlé</u> "lee, shelter", and <u>fé</u> "livestock, sheep, property".

```
BEGIN borð "table" N
    ; borð borð borði borðs
    ; borð borð borðum borða
    ; borðið borðið borðinu borðsins
    ; borðin borðin borðunum borðanna
    BORROW ALL nondef FROM hestur BUT ns np ap
    EQUATE as ns
    EQUATE ustem np ap
    INCLUDE def_kyn

BEGIN böl "calamity" N
    ; böl böl bölvi böls
    ; böl böl bölum bölva
    ; bölið bölið bölvinu bölsins
    ; bölin bölin bölunum bölvanna
    BORROW ALL nondef FROM borð BUT gp
    DERIVE gp FROM astem VIA +v +a
    INCLUDE def_kyn
```

Note: old ds was <u>bölvi</u>, but according to Kress, this is now gone.

```
BEGIN barn "child" N
    ; barn barn barni barns
    ; börn börn börnum barna
    ; barnið barnið barninu barnsins
    ; börnin börnin börnunum barnanna
```

```
    BORROW ALL nondef FROM hattur BUT ns np ap
    EQUATE ns as
    EQUATE ustem np ap
    INCLUDE def_kyn
```

Note: borð is to barn as hestur is to hattur.

```
DEFINE def_hreiður
    DERIVE nsd FROM astem VIA +ið
    DERIVE asd FROM astem VIA +ið
    DERIVE dsd FROM ns VIA +nu
    DERIVE gsd FROM ns VIA +ins
    DERIVE npd FROM astem VIA +in
    DERIVE apd FROM astem VIA +in
    DERIVE dpd FROM ns VIA -m +num
    DERIVE gpd FROM ns VIA +nna

BEGIN hreiður "nest" N
    ; hreiður hreiður hreiðri hreiðurs
    ; hreiður hreiður hreiðrum hreiðra
    ; hreiðrið hreiðrið hreiðrinu hreiðursins
    ; hreiðrin hreiðrin hreiðrunum hreiðranna
    BORROW ALL nondef BUT np ap FROM akur
            AND SPLIT BETWEEN astem ustem
    EQUATE astem ustem
    EQUATE cstem np ap
    INCLUDE def_hreiður

BEGIN daður "flirting" N
    ; daður daður daðri daðurs
    ; daður daður döðrum daðra
    ; daðrið daðrið daðrinu daðursins
    ; daðrin daðrin döðrunum daðranna
    BORROW ALL nondef BUT ap np FROM akur
    DERIVE np ap FROM astem VIA r:ur
    INCLUDE def_hreiður
```

Note: also þvaður gossip.

```
DEFINE def_tími
    DERIVE nsd FROM ns VIA +nn
    DERIVE asd FROM as VIA +nn
    DERIVE dsd FROM ns VIA +num
    DERIVE gsd FROM ns VIA +ns
    DERIVE npd FROM np VIA +nir
    DERIVE apd FROM ap VIA +na
    DERIVE dpd FROM ns VIA -m +num
```

```
    DERIVE gpd FROM ns VIA +nna

BEGIN tími "time" M
    ; tími tíma tíma tíma
    ; tímar tíma tímum tíma
    ; tíminn tímann tímanum tímans
    ; tímarnir tímana tímunum tímanna
    BORROW np ap dp gp FROM hestur
    DERIVE ns FROM astem VIA +i
    DERIVE ns as ds FROM astem VIA +a
    INCLUDE def_tími

BEGIN afi "grandfather" M
    ; afi afa afa afa
    ; afar afa öfum afa
    ; afinn afann afanum afans
    ; afarnir afana öfunum afanna
    BORROW plu FROM hattur
    BORROW sg FROM tími
    INCLUDE def_tími

BEGIN dómari "judge" M
    ; dómari dómara dómara dómara
    ; dómarar dómara dómurum dómara
    ; dómarinn dómarann dómaranum dómarans
    ; dómararnir dómarana dómurunum dómaranna
    BORROW ALL FROM tími
            AND SPLIT BETWEEN astem ustem
    DERIVE ustem FROM astem VIA +a:u
    INCLUDE def_tími

BEGIN bakari "baker" M
    ; bakari bakara bakara bakara
    ; bakarar bakara bökurum bakara
    ; bakarinn bakarann bakaranum bakarans
    ; bakararnir bakarana bökurunum bakaranna
    BORROW ALL FROM tími
            AND SPLIT BETWEEN astem ustem
    DERIVE ustem FROM astem VIA +a-a:ö-u
    INCLUDE def_tími

BEGIN nemandi "pupil" M
    ; nemandi nemanda nemanda nemanda
    ; nemendur nemendur nemendum nemenda
    ; nemandinn nemandann nemandanum nemandans
    ; nemendurnir nemendurna nemendunum nemendanna
    BORROW sg FROM tími
    DERIVE istem FROM astem VIA +a:e
    DERIVE ap np FROM istem VIA +r +r:ur
    DERIVE dp FROM istem VIA +u +m
    DERIVE gp FROM istem VIA +a
```

```
    INCLUDE def_tími

DEFINE def_lilja
    DERIVE nsd FROM ns VIA +n
    DERIVE asd FROM as VIA +na
    DERIVE dsd FROM ns VIA +nni
    DERIVE gsd FROM ns VIA +nnar
    DERIVE npd FROM np VIA +nar
    DERIVE apd FROM ap VIA +nar
    DERIVE dpd FROM ns VIA -m +num
    DERIVE gpd FROM ns VIA +nna

BEGIN lilja "lily" F
    ; lilja lilju lilju lilju
    ; liljur liljur liljum lilja
    ; liljan liljuna liljunni liljunnar
    ; liljurnar liljurnar liljunum liljanna
    BORROW gp astem ustem dp FROM hestur
    EQUATE gp ns
    DERIVE as ds gs FROM ustem VIA +u
    DERIVE ap np FROM ustem VIA +r +r:ur
    INCLUDE def_lilja

BEGIN amma "grandmother" F
    ; amma ömmu ömmu ömmu
    ; ömmur ömmur ömmum amma
    ; amman ömmuna ömmunni ömmunnar
    ; ömmurnar ömmurnar ömmunum ammanna
    BORROW ALL FROM lilja
            AND SPLIT BETWEEN astem ustem
    DERIVE ustem FROM astem VIA +a:ö

BEGIN tunga "tongue" F
    ; tunga tungu tungu tungu
    ; tungur tungur tungum tungna
    ; tungan tunguna tungunni tungunnar
    ; tungurnar tungurnar tungunum tungnanna
    BORROW ALL FROM lilja BUT gp
    DERIVE gp FROM astem VIA +n +a
    INCLUDE def_lilja

BEGIN saga "story" F
    ; saga sögu sögu sögu
    ; sögur sögur sögum sagna
    ; sagan söguna sögunni sögunnar
    ; sögurnar sögurnar sögunum sagnanna
    BORROW ALL BUT gp FROM amma
    BORROW gp astem FROM tunga
    INCLUDE def_lilja
```

```
BEGIN lygi "lie" F
    ; lygi lygi lygi lygi
    ; lygar lygar lygum lyga
    ; lygin lygina lyginni lyginnar
    ; lygarnar lygarnar lygunum lyganna
    BORROW ALL BUT sing def FROM kinn
    DERIVE ns as ds gs FROM istem VIA +i
    INCLUDE def_lilja

BEGIN æfi "life" F
    ; æfi æfi æfi æfi
    ; æfir æfir æfum æfa
    ; æfin æfina æfinni æfinnar
    ; æfirnar æfirnar æfunum æfanna
    BORROW plu FROM tíð
    BORROW sg FROM lygi
    INCLUDE def_lilja

DEFINE def auga
    DERIVE nsd FROM ns VIA +ð
    DERIVE nsd FROM ns VIA +ð
    DERIVE nsd FROM ns VIA +nu
    DERIVE nsd FROM ns VIA +ns
    DERIVE nsd FROM ns VIA +n
    DERIVE nsd FROM ns VIA +n
    DERIVE nsd FROM ns VIA -m +num
    DERIVE nsd FROM ns VIA +nna

BEGIN auga "eye" N
    ; auga auga auga auga
    ; augu augu augum augna
    ; augað augað auganu augans
    ; augun augun augunum augnanna
    EQUATE astem ustem
    DERIVE ns as ds gs FROM astem VIA +a
    BORROW astem gp FROM tunga
    DERIVE np ap FROM ustem VIA +u
    BORROW ustem dp FROM hestur
    INCLUDE def_auga

BEGIN hjarta "heart" N
    ; hjarta hjarta hjarta hjarta
    ; hjörtu hjörtu hjörtum hjartna
    ; hjartað hjartað hjartanu hjartans
    ; hjörtun hjörtun hjörtunum hjartnanna
    BORROW ALL FROM auga
            AND SPLIT BETWEEN astem ustem
    DERIVE ustem FROM astem VIA +a:ö
```

CHAPTER 4

CODA

The preceding chapters describe a new theory of inflectional morphology, define a syntax for representing the theoretical structures, and apply the formalism to a body of data from Modern Icelandic. They emphasized the linguistic processes and relations in the specific paradigms discussed. However, I feel that it is important that linguistic theories be applicable to a wider range of problems than just manipulation of abstract variables, however interesting this may be. If the linguistic theory is to help solve human problems, then it must be a theory about human beings.

Human beings are not abstract entities. The use of abstract constructs to describe them is probably always doomed from the outset to failure. However, researchers, being human themselves, are able to see beyond the abstractions which they propose to a human reality which is formed as part of the process of perceiving and manipulating the abstract symbols. Therefore, however imperfect such abstract systems must be in the objective sense, it is possible to use them to communicate the essential emotional experience of insight about some domain from one person to another.

Another fundamental attribute of researchers as human beings seems to be the search for consistency. There is an emotion known to every researcher which is felt when abstract concepts are joined together in novel ways to create new abstract systems. Seemingly, the new system is no longer only the juxtaposition of the original ideas; instead it is something new and different in its own right, more highly valued than any of its precedents.

This process of insightful synthesis leads to a methodology. First, develop a theory which seems to describe and explain some abstracted domain. Then, when the mental connections between the abstract notions of the system and the abstracted domain are reasonably well developed, apply the same theory to a new domain. Make whatever changes are required for consistency. These changes should be minimal, and if possible should actually improve the efficacy of the theory for the original domain. Continue forever.

The first portion of this chapter applies this methodology to FPG in three ways. First, the results of a small experiment are reported which appear to support FPG in contrast to certain alternatives. Second, an attempt is made to extend FPG to historical language change. Third, some speculations are made concerning use of FPG in syntax. The final section of the chapter is a brief summary of the dissertation.

4.1. Some preliminary experimental results. Over the last decade or so, investigations of language behavior have been carried out to test hypotheses in linguistic morphology both by linguists (Zimmer 1969; Ohala 1974; Ohala and Ohala 1975; Fromkin 1975; Lukatela, Mandić, Gligorijević, Kostić, Savić, and Turvey 1978; Bradley 1980) and psychologists (Anisfeld 1969; Gibson and Guinet 1971; Jarvella and Snodgrass 1974; MacKay 1976, 1978; Rubin and Becker 1979; Stanners, Neiser, Hernon, and Hall 1979; Stanners, Neiser, and Painton 1979; Taft and Forster 1975). However there is a general reluctance on the part of non-experimental linguists to incorporate this data into formal theories of grammar (Derwing 1979). One reason for this is that the relation between a theory of grammar and a set of language behavior data is sometimes controversial or at least not clearcut (Fromkin 1975). As Wirth (1975) suggests, the way to remedy this is to develop models of language which make specific predictions about both linguistic competence and performance.

The results presented in this section describe a small experiment which was carried out to get some feel for the extent to which the predictions of a network model such as FPG (based on surface similarities) would jibe with performance data. For purposes of comparison, two straw man models are set up. A pair of competence models are converted into performance models by the most obvious and simple methods (such as using generative rules as production rules), then determining how the result might be modified to correspond to facts about language behavior. This approach has been used before (e.g. Ohala 1974; MacKay 1976) although it has been criticized (Fromkin 1975) for not relating directly to the theoretical issues involved in the ´converted´ models. The experimental data given here is not to be taken as evidence one way or another for the competence models, but instead as evidence suggesting the kind of model which would be required in a psychological extension of FPG. It should be kept in mind that the linguistic models referred to here are isolated portions of theories developed in very different contexts, and this experiment is not intended to ´prove´ or to ´disprove´ these models.

4.1.1. Two straw men. There are two fairly well-known models for inflectional morphology which have been proposed in the last decade. The first model, which I will refer to as the list model (LM), was proposed by Halle (1973). In this model, there is a component of the grammar which contains a list of all fully inflected forms organized into paradigms (a paradigm being a generalized set of surface forms which are related to one another in a semantically predictable way and which ordinarily share certain morphological similarities). Insertion of a lexical item into the sentence matrix in this model means that the entire paradigm is inserted. Later rules then eliminate all but the one appropriate form. Similar models (i.e. the lexicon contains a list of inflected forms) have been developed by Jackendoff (1975) and Vennemann (1974). The second model, which I will refer to as the syntactic model (SM), is discussed by Anderson (1977) among others. In this model, inflections are derived from (possibly) abstract elements which are lexical. There is no list of inflected forms; paradigmatic relations are indirectly represented.

Now consider the difference in the process of derivation of an inflected form in FPG and SM. In SM, there must be a constant linear sequence of intermediate forms, each connected to the next via a rule of some sort. In FPG, however, while linear connections are easily represented, it is possible for there to be different paths between two inflectional categories for different paradigms. This means that it is possible for a given derivation to be faster in one paradigm than in another. In both of these models rules are assumed to be ordered in time. For present purposes, I will ignore models wherein rules are not ordered.

Note that FPG cannot be distinguished from a fourth model where all of the surface forms are listed in the lexicon, but are accessed via derivational rules rather than as elements in a list or table. This kind of model is essentially what is proposed in Leben and Robinson (1977). The gist of their theory is that phonological rules are applied to surface forms in order to determine whether they can be legitimately related to other surface forms. Just enough of the phonology is undone to find a match. In this experiment, FPG and upside down phonology would make similar predictions. (In their paper, they propose an experiment similar to this one, but do not actually carry it out).

Assuming that users of language actually traverse these paths (or look up forms in lists) in real time, it should be possible to choose among the three models by measuring the difference in time it takes to perform some task involving paradigmatic relations. LM might be supposed to predict that since transforming one form into another involves searching an ordered list the time to perform the transformation is dependent only on the relative positions of the forms in the list; that is, on their paradigmatic categories. SM, on the other hand, would predict that a constant linear sequence of rules would have to be undone (undoing rules is part of the conversion from competence model to performance model), then another similar sequence applied. No variations in rule ordering from paradigm to paradigm would be allowed. That is, there is only one path between surface forms, always leading through the single abstract underlying form. Finally, FPG predicts that it is possible for optimized paths to be available for each paradigm, resulting in more efficient performance.

Several experimental paradigms have been reported in the psychological literature which are relevant to this study. These experiments concern the problem of organization of the internal lexicon, specifically as to whether certain categories of words have separate lexical entries or are derived via morphological rules from lexical items. It is assumed in most of these studies that lexical items are stored and retrieved as lists ordered with most frequently occurring items first (Glanzer and Ehrenreich 1979, Bradley 1980). But there are certain results which suggest that lexical relations might have a more complex structure (Wickelgren 1976).

A number of studies use the lexical decision task on visually presented forms (e.g. Jastrzembski and Stanners 1975; Lukatela et al. 1978). The lexical decision task is an experimental procedure in which subjects are asked to decide as quickly as possible whether or not a given test item is a word. The usual dependent variables are response latency and errors. It is assumed that shorter positive response latencies are associated with items having their own lexical entries. Longer positive response latencies are associated with relatively infrequent items. Longer negative response latencies are associated with items which appear to be morphologically related to existing lexical items.

Another set of studies use a production task where subjects are given a word auditorily and asked to say a related word (MacKay 1976, 1978). Dependent variables are response latency and error category. Longer response latencies are associated with longer derivations. Errors are similar to hypothetical intermediate stages in the derivations.

Jarvella and Snodgrass (1974) use a verification task to infer the extent to which speakers have direct access to the stem morpheme presumed to underlie inflected forms. The task was to decide as quickly as possible whether two forms share the same stem morpheme. They found that spelling differences are more important than pronunciation differences in visually presented items. This experimental paradigm seems best suited for contrasting the three models described above.

While English inflectional morphology is not a particularly good source of test items, it is possible to choose from the set of strong verbs items which provide a means of comparing the three models. Three paradigmatically related verb forms will be considered, namely the infinitive (I), the past tense (PT) and the past participle (PP) for two classes of English strong verbs. The first class contains verbs which pattern like tear/tore/torn; the second class patterns like blow/blew/blown. The difference between the two classes (at a very superficial level of analysis) is that in the former, the main vowel of I is distinct from that of both PT and PP, while in the latter, the main vowel of PT is distinct from that of I and PP. The task is to verify that both of a given pair of words chosen from these three categories are members of the same paradigm. The three models described above (preformance versions of competence models) make different predictions about response latency in this task. The next section gives these predictions in detail.

4.1.2. Predictions for LM. All paradigms are stored in ordered lists. Position in the list determines the paradigmatic category of a form. (This is the simplest performance conversion. Other conversions would include randomly ordered tagged lists, content-addressable memory, parallel-access lists, and so on. See Vennemann (1974) for a discussion.) The subject searches the list item by item until both forms are found. Assume that the list is stored in the order I PT PP (any ordering which is the same for all paradigms could be used without changing the essential characteristic that verification time is proportional to position in the list), and that each comparison takes one increment of time. Then the following relative latencies should obtain:

5	blew/blown (PT/PP)	tore/torn (PT/PP)
4	blow/blown (I/PP)	tear/torn (I/PP)
3	blow/blew (I/PT)	tear/tore (I/PT)

For example, to verify tear/torn, the following comparisons would be made: tear::tear; torn::tear, torn::tore, torn::torn. The numbers in this case represent the number of comparisons needed, for convenience assuming that the required forms are the first three in the paradigm list.

4.1.3. Predictions for SM. All forms must be derived from an underlying abstract form. For convenience, assume that the underlying form in each case is identical to I. (This is a simplification, but I know no reason why any other assumption would affect the present hypothesis.) In order to verify two forms, the derivations of both forms must be undone until both reach the underlying form. Assume that there are two rules involved, one which changes the main vowel and the other which adds the -(e)n suffix. In this case, the derivational paths can be represented as (underlying form in caps):

```
BLOW -[+vowel]-> blew
BLOW -[+suffix]-> blown
blew -[-vowel]-> BLOW -[+suffix]-> blown

TEAR -[+vowel]-> tore
TEAR -[+vowel]-> tore -[+suffix]-> torn
tore -[-vowel]-> TEAR -[+vowel]-> tore -[+suffix]-> torn
```

If it takes one time increment to undo or to apply a rule, the following relative latencies[2] are expected:

3		tore/torn (PT/PP)
2	blew/blown (PT/PP)	tear/torn (I/PP)
1	blow/blew, blow/blown (I/PT, I/PP)	tear/tore (I/PT)

In this table, the numbers represent the number of processes required in order to convert the items into comparable forms.

4.1.4. Predictions for FPG. Forms are derived from each other more or less as for SM, but different paradigms can differ as to the paths taken. Assuming that the two rules mentioned above are used, we would see the same paths as for SM with one exception:

```
blow -[+vowel]-> blew
blow -[+suffix]-> blown
blew -[-vowel]-> blow -[+suffix]-> blown

tear -[+vowel]-> tore
tear -[+vowel]-> tore -[+suffix]-> torn
tore -[+suffix]-> torn
```

Under the assumption that applying or undoing a rule takes one time increment, this model leads to the following predictions:

2	blew/blown (PT/PP)	tear/torn (I/PP)
1	blow/blew, blow/blown (I/PT, I/PP)	tear/tore, tore/torn (I/PT, PT/PP)

The following is a preliminary study of a technique for determining the real time distance between points in hypothetical inflectional space. It uses the above verbal types and measures response latencies associated with various inflectional transformations.

4.1.5. Subjects. Three female high school students and one male college student participated in the experiments and were paid for their services.

4.1.6. Apparatus. Instructions and stimuli were delivered to the subjects via MIME-I CRT computer terminals operating at 9600 baud. The difference between the longest and the shortest pairs in terms of display time was less than 16 milliseconds. Subjects responded by pressing a key on the keyboard. The computer recorded which key was pressed and the response time (in 1/60 seconds) from the time that the last letter of the stimulus was written out. Experiments were run in a dark, quiet, isolated booth.

4.1.7. Instructions. The following instructions were displayed on the CRT screen before each block of trials:

> This experiment measures how quickly you can make judgments about English verb forms. On each trial you will be shown two words which may or may not be forms of the same verb. If you think the two are forms of the same verb so signify by pressing the ´J´ key. If you think the two words cannot be derived from the same verb press the ´F´ key. For example, given the pair of words
>
> ran ren
>
> you would push the ´F´ key, because they cannot come from the same verb. But on the other hand given the words
>
> come came
>
> you would press the ´J´ key because they can be derived from the same verb. To facilitate fast responding, please lightly rest your right index finger on ´J´ and your left index finger on ´F´. When you are ready to begin, please press either ´J´ or ´F´. Every few trials, the program will pause to allow you to rest.

In addition, the string "F=NO" was displayed in the upper-left portion of the screen and the string "J=YES" was displayed in the upper-right portion. After every 10 trials, the the program paused until the subject was ready to continue. Otherwise, all timing was dependent on the program and the response latencies. Before beginning the actual experiment, the two example pairs ran/ren and come/came were presented but the responses were not recorded.

4.1.8. The Verbs. Ten verbs of each type were used. The first set were all of the type blow/blew/blown (the vowel of I is the same as the vowel of PP) and the second set were all of the type tear/tore/torn (the vowel of PT is the same as the vowel of PP). In addition to the three principal parts, a false form (FF) for each verb was created by adding -(e)n to either I or PT, whichever one did not contain the main vowel used in PP (for example, blow/blew/blown/blewn and tear/tore/torn/tearn). This provided a set of 80 verb forms. The verbs used were: (set one) blow, draw, fall, give, grow, know, see, shake, take, throw; (set two) bite, choose, forget, freeze, speak, steal, swear, tear, wear, weave.

4.1.9. Design. Stimuli consisted of pairs of verb forms from the two sets as follows: I/I I/PT I/PP I/FF PT/PT PT/PP PT/FF PP/PP PP/FF FF/FF (each form paired with each other form). Both orders were used (that is, both I/PT and PT/I) unless not possible (as for I/I). In a block of trials, all of the resulting 320 pairs were presented once, in an individual random sequence.

4.1.10. Results. Negative responses were irrelevant to the particular comparisons needed here and so were not analyzed. Furthermore, trials with unusually long response latencies (greater than 2 seconds) were discarded (about 2% of all trials). Reaction times for all subjects and all blocks were combined to get an overall average response time for "yes" responses to the key pairs I/PT I/PP and PT/PP for the two classes of verbs:

0.88	blew/blown (PT/PP) 10.9%	
0.85		tear/torn (I/PP 6.7%
0.82		tore/torn (PT/PP) 4.0%
0.80	blow/blown (I/PP) 8.6%	
0.77		tear/tore (I/PT) 3.4%
0.72	blow/blew (I/PT) 3.4%	

In addition, planned comparisons revealed blow/blew $\neq$ tear/tore ($t(3) = 1.88$, n.s.), blow/blown $<$ tear/torn ($t(3) = 3.04$, $p < .05$), and blew/blown $>$ tore/torn ($t(3) = 3.53$, $p < .05$). There was no speed/accuracy trade-off, since there were relatively more errors associated with longer latencies, the only exception being blow/blown ($r = .75$, $p < .05$).

Post-hoc comparisons were also made within categories. It was found that tear/tore = tore/torn ($t(3) = 1.73$, n.s.), but that all of the other comparisons were significant: tear/torn > tore/torn ($t(3) = 3.06$, $p < .05$); tear/torn > tear/tore ($t(3) = 4.25$, $p < .05$); blow/blown > blow/blew ($t(3) = 6.57$, $p < .005$); blew/blown > blow/blown ($t(3) = 6.63$, $p < .005$); blew/blown > blow/blew ($t(3) = 7.42$, $p < .005$).

4.1.11. Comparison of the Models. The results clearly support FPG over LM and SM. LM is not consistent with the results because it predicts that the sole determiner of latencies between forms is the storage order in the paradigm list. Thus there should be no difference between the two classes of verbs. The significant differences blow/blown < tear/torn and blew/blown > tore/torn indicate that there is a difference and therefore the model is not supported by the data.

SM is also not consistent with the data. It predicts that tore/torn > blew/blown, since the vowel rule must be undone for both tore and torn in order to get back to the underlying form tear. In fact, there is a significant difference in the opposite direction (tore/torn < blew/blown). This indicates that the model is not supported by the data.

FPG is supported by the data. It predicts the difference tore/torn < blew/blown since the two paradigms use paths of different lengths for the derivations. The path for tore/torn does not go through the underlying form, while the one for blew/blown must.

Notice that the fact that tear/tore = blow/blew indicates that there is no significant difference between the two categories as a result of such factors as semantic differences or basic lexical access time. This implies that the differences found for the other categories are a result of the inflectional processes.

4.1.12. Implications. These results suggest that inflectional paradigms are stored in some form which exhibits certain maze-like characteristics. Specifically, there are sequences of rules connecting surface forms which are not required to pass through hypothetical underlying forms. Notice that this is not an argument against underlying forms. It is quite possible for such forms to exist in FPG. The argument is only that alternate paths through the maze defined by paradigm categories and inflectional processes must be allowed. Obviously, LM and SM are not to be taken seriously as models for inflectional morphology, either psychological or otherwise. These imaginary creations are only the most obvious reaction-time models which can be built from current generative models of morphology. It would be possible to apply the insightful synthesis methodology to a generative model of inflectional morphology and construct a system which can account for reaction time data such as those reported here. For example, it would be possible to maintain a variant of LM where the elements of the list are ordered in some other way. If the tear forms are ordered tore-tear-torn and the blow forms are ordered blow-blew-blown, then the observed results would be predicted. However, if this is done then some means other than simple order in the list must be used to determine the inflectional category (besides, what would determine the ordering?). If instead of ordering the elements of the paradigm in a list the forms are connected via rules as suggested in Vennemann (1972), then it should be possible to arrange things in a manner consistent with the results.

4.1.13. Problems and Future Research. The present report is based on data from only four subjects. Since all four performed similarly, the expected differences were supported at statistically significant levels. However, it would be worthwhile to try to replicate the results of this preliminary experiment with a larger subject group. Another issue has to do with interpreting data from visually presented forms. The relation between this type of language behavior and spoken language is not well known. There is evidence in favor of a model of reading which involves conversion of written language into a phonemic form (Spoehr 1978) based on longer lexical decision latencies associating with longer phonological codes, but there is also evidence to the contrary (Taft 1979) based on longer lexical decisions when blank space is inserted at orthographic rather than phonological boundaries. The way to resolve this source of interference is to gather additional data testing the same hypothesis but using auditory presentation.

There is another problem due to the fact that there are very few compound inflectional processes in English (that is, cases where the concept of variation among paradigms is relevant). In fact, the only one that I know of is the set of strong verbs used in this study. In addition, nearly all of the existing verbs of these two types were used. (Verbs were excluded if alternate forms are in common use, such as get/got/got versus get/got/gotten.) Therefore it is difficult to generalize from these results.

For this reason it would be desirable to perform future research using subjects who are speakers of a more suitable language. If this were the case it would be possible to find a large set of test items and randomly sample from it in a particular block, a situation more easily allowing generalizations to be made about the results. In languages with more complex inflectional processes it would be possible to derive reasonable estimates of the process latency associated with rules in a putative analysis. This would allow the analysis to to be changed to mirror more closely the actual cognitive structures and processes of the language user. In turn, this would allow the development of a universal theory of inflectional paradigms to be developed which corresponds to cognitive reality. In particular, it would be quite interesting to test the large bodies of inflectional relations given in Chapter 3 for Icelandic using this procedure.

4.2. FPG Applied to Historical Linguistics. A very interesting part of the general puzzle of language change, and one which has not generally been focused directly upon by historical linguists, is the problem of change in the internal structure of inflectional paradigms. (Generally the individual forms are traced through the years--the relations of the form to the paradigm are secondary). This type of change would include migration from one paradigm to another, groups of relations drifting from one paradigm into another, and loss of complexity within and among paradigms. Perhaps this study has not been at the center of research on paradigm change because of the lack of a way to characterize explicitly all of the inflectional relations among the members of a paradigm so that they can be directly compared and manipulated. Since FPG could provide a way to do this, this section explores two problems in inflectional change as a tentative proposal for a new methodology in historical linguistics.

4.2.1. Old and Early Middle English Strong Nouns. A particularly clean example of paradigm regularization comes from data in Moore (1966), in particular changes in the strong noun declensions. Moore gives the forms of certain of these nouns at three stages of their development: the Old English (OE) period, from the earliest writings up to the time of the Norman Conquest; Early Middle English (EME), up to about the 12th century; and Late Middle English (LME), at the period of Chaucer. The forms at the LME period were so reduced in complexity as to be less interesting (their paradigm was essentially the same as for Modern English), but the change from OE to EME is worth inspection.

We will illustrate the developments by use of five example nouns forms: dōm "judgment", ende "end", lufu "love", hwīl "period of time", lim "limb", Here are the paradigms for OE:

(4.2.1.1)

		Masculine		Feminine		Neuter
Sg	Nom	dōm	ende	lufu	hwīl	lim
	Acc	dōm	ende	lufe	hwīle	lim
	Dat	dōme	ende	lufe	hwīle	lime
	Gen	dōmes	endes	lufe	hwīle	limes
Pl	Nom	dōmas	endas	lufa	hwīla	limu
	Acc	dōmas	endas	lufa	hwīla	limu
	Dat	dōmum	endum	lufum	hwīlum	limum
	Gen	dōma	enda	lufa	hwīla	lima

FPG analysis of this system would exploit the rather obvious similarities among the genders and categories. The most striking, other than the extensive syncretism, is the formation of dative singular, dative plural, and genitive plural, which not only is constant throughout the paradigms, but which can be thought of as sort of a backbone for the other categories:

```
           --[+u]--> --[+m]--> dp
          (
    stem ----[+e]--> ds
          (
           --[+a]--> gp
```

Using this structure, the five OE examples can be defined as follows:

```
DEFINE oe_backbone
            DERIVE dp FROM stem VIA +u +m
            DERIVE ds FROM stem VIA +e
            DERIVE gp FROM stem VIA +a

BEGIN dōm "judgment" M
            INCLUDE oe_backbone
            EQUATE stem ns as
            DERIVE np ap FROM gp VIA +s
            DERIVE gs FROM ds VIA +s

BEGIN ende "end" M
            BORROW ALL BUT ns as FROM dōm
            EQUATE ds ns as
```

```
BEGIN lim "limb" N
            BORROW ALL BUT np ap FROM dōm
            DERIVE np ap FROM stem VIA +u

BEGIN lufu "love" F
            INCLUDE oe_backbone
            EQUATE ds as gs
            EQUATE gp np ap
            DERIVE ns FROM stem VIA +u

BEGIN hwīl "while, period of time" F
            BORROW ALL BUT ns FROM lufu
            EQUATE stem ns
```

Notice that the difference between the two masculine examples and between the two feminine ones is a shift of the same categories, although to different locations in the paradigm. Also, the network implied by the feminine forms is simpler than that for the masculine and neuter, since the arcs with +s are not used in the feminine networks.

There were two primary changes in pronunciation between OE and EME which are relevant to the declensions. Unstressed vowels in endings were reduced to e, thought to be pronounced as schwa. This would have a major effect on the declensions given above, since the backbone of the OE system is essentially a three-way division based on suffixation of different unstressed vowels: +a +e, and +u. The other change was that word-final nasals were reduced to n. This effects only the dative plural ending.

Here are the paradigms for EME:

(4.2.1.3)

		Masculine		Feminine		Neuter
Sg	Nom	doom	ende	luve	hwile	lim
	Acc	doom	ende	luve	hwile	lim
	Dat	doome	ende	luve	hwile	lime
	Gen	doomes	endes	luve	hwile	limes
Pl	Nom	doomes	endes	luve	hwile	lime
	Acc	doomes	endes	luve	hwile	lime
	Dat	doomen	enden	luven	hwilen	limen
	Gen	doome	ende	luve	hwile	lime

Most of the collapsing of categories is due to the simplification of the stem vowels. This would have the effect of combining the three subnetworks distinguished by the three vowels into a single one. This would take place automatically by the tree pruning convention. (All arcs leaving a given node which have the same rule are collapsed; nodes with no arcs leaving and with no labels are pruned). In addition, one of the feminine paradigms is lost. This results in the following system:

```
DEFINE eme backbone
          DERIVE dp FROM stem VIA +e +m
          DERIVE ds FROM stem VIA +e
          DERIVE gp FROM stem VIA +e

BEGIN doom "judgment" M
          INCLUDE eme_backbone
          EQUATE stem ns as
          DERIVE np ap FROM gp VIA +s
          DERIVE gs FROM ds VIA +s

BEGIN ende "end" M
          BORROW ALL BUT ns as FROM dōm
          EQUATE ds ns as

BEGIN lim "limb" N
          BORROW ALL BUT np ap FROM dōm
          DERIVE np ap FROM stem VIA +e

BEGIN luve "love" F
          INCLUDE eme_backbone
          EQUATE ds as gs
          EQUATE gp np ap
```

```
            DERIVE ns FROM stem VIA +e

BEGIN hwile "while, period of time" F
            BORROW ALL FROM lufu
```

This is written out in such a way as to emphasize its similarities with the OE system. It could have been written a little more simply by combining lines with the same rule instead of letting the interpreter collapse them by tree pruning.

It is interesting to compare Moore's account of the change in the ns form of _hwile_ with the one given here. He states that by process of analogy with other feminines, the paradigm for _hwile_ and others in its class were modified. In FPG, the more natural generalization is that nouns in the _hwile_ class were reclassified and moved into the other feminine paradigm when their own paradigm was deleted. There is not much difference in the actual result, but in the analogy account, each individual member of the old _hwile_ class would have had to have been reclassified individually, while in the more holistic FPG account, the two paradigms were simply replaced with one. Eventually, all five of these paradigms were replaced with one single noun paradigm, in LME.

4.2.2. Old Norse and Modern Icelandic. Sometimes words change from one paradigm to another. An interesting question is whether single forms are units of change or whether the entire paradigm of a word changes as a unit.

4.2.2.1. Idiosyncratic Paradigms. In fact, this approach is useful in studying sound change per se, especially if sound change is seen as always being the cause of paradigm change. There are many examples where some particular word uses a modified paradigm without any transferal to other paradigms. For example, refer to the np and npd of maður "man" given in Chapter 3. This variation (equating np to ap, but including the standard derivation for np in the path from ap to npd: menn ap & np; mennirnir npd, not *menninir) is used in no other paradigms in the language. For ON menninir is the form given in most modern-day grammars (such as Gordon 1957).

4.2.2.2. Modification of Paradigms. There are clear cases where all members of a certain paradigm change a single form or group of forms. An example from Icelandic is the group sometimes called u-stem nouns, including köttur "cat" and fjörður "fjord". In Old Norse, ap for these words was derived from the ustem via +u; in Modern Icelandic, the paradigm is altered by deriving ap from the istem via +i (collapsing with the np branch (+i +r) and also conforming to a nearly complete generalization about masculine strong nouns that np is derived from ap via +r). This type of change essentially deletes an earlier paradigm pattern from the language and replaces it with a modified version of itself. However, it is different from the type of change seen above in English where most paradigms in the language were rewritten as a result of (or simultaneously with) sound change, because only one paradigm was affected by the change. Furthermore, it is unlikely that the change in the u-stems could be attributed to phonology, especially since the suffixing of +u to the ustem exists in several other paradigms today, such as the weak adjective plurals.

This particular example is important because it seems to be a rather clear case of analogy or generalization. However, it is an analogy based not on surface similarities between one form and another but on similarities between one subparadigm and another. That is, the subnetwork for ap in the older paradigm is deleted and replaced with a corresponding, more general, subnetwork. This change suggests that paradigmatic relations are important as units in language change.

4.2.2.3. Paradigm Crossover. A very interesting kind of change takes place when a word shifts from one paradigm to another. There are many examples of this kind of change. Some cause corresponding changes in syntax, such as grammatical gender change. I have no ideas about why these changes take place, but FPG should be able to provide some insight into the process of the change. There is an interesting example in Icelandic of such a change which is not yet complete. The masculine noun leikur "play" began as an a-stem (like hestur), according to Cleasby et al. (1957) and Iversen (1937). However, it is now classified as an i-stem (like bekkur or gestur, according to Kress, although Einarsson refers to as a mixture of his class one (a-stem) and class two (i-stem). It has many compounds, which vary in paradigm patterns: dansleikur "dance", tónleikur "concert, music", fimleikar "gymnastics" (plu only), skopleikur "comedy" (mixed); hljómleikur "concert, music" (class one only); einleikur "(instrumental) solo", gjörvuleikur "accomplishment, talent" (class two only). These classifications are according to Einarsson, who is the only writer to put enough grammatical information in his word list to be used. He does not specify what is meant by 'mixed', unfortunately, so it is not clear whether all forms are used by all speakers or whether there are dialectal differences (the more probable, in my opinion). It is very clear, however, that the variation in the compounds can exist for individual speakers. For example, in Einar Pálsson's Icelandic in Easy Stages, Vol. 2 (Reykjavík: Mímir 1977), in the reading passage for lesson 72, söngleikir is given as np of söngleikur "musical" (class 2), while on the same page tónleika is given as ap of tónleikur (class 1). In any case, there seems to be a smear of transition as the word moves from class one to class two. Either that, or the compounding process which originally created the words is now defunct and the whole words are randomly moving around between the two paradigms, a rather unsatisfying explanation.

It is possible that this process of paradigm change is similar to the process of change by redundant innovation described in Justus (1978). In this kind of change, an innovation which is initially redundant eventually takes the place of its predecessor. The important aspect of this process is that there must be some period of time where both variants coexist, until the newer one wins out over the older one. This coexistence of variants during a transitional period seems quite similar to the modern situation in Icelandic for the -leikur words.

This kind of change is quite different from the change in the u-stems. The only form effected seems to be leikur and its compounds. The paradigms themselves remain the same. The interesting thing about this is that again, whole paradigms rather than derivations of individual forms seem to function as units during the process of change.

There is another interesting thing about the situation. If inflectional processes are to be triggered by abstract phonological segments in the lexical entry, then you end up with two very different lexical forms for the -leikur in class one compounds versus class two compounds, which surely seems counterintuitive.

4.3. Syntax. In normal linguistic usage, the term paradigm is used not only to refer to inflectional paradigms, as in this dissertation, but also to refer to other linguistic systems. In general, anything which can be set up in a table of two or more contrasting degrees can be called a paradigm. In part, it was this extended usage to which the name Finite Paradigm Grammar corresponds, although the focus of this work is on the inflectional paradigm. This section explores in a very tentative way how the ideas presented above might be extended into the domain of syntax.

4.3.1. Periphrastic forms. Many languages often use periphrasis instead of morphology. These periphrastic forms can function more or less like inflected forms, with two major differences. In the first place, it doesn't seem reasonable to suppose that there would be a lexical entry corresponding to each periphrastic paradigm. If there were, there would be an incredible explosion in the number of lexical entries needed in the system. Take for example this adaptation of a partial paradigm given by Quirk et al. (1972) for complex finite verb phrases in Modern English:

may have examined
may be examining
may be examined
has been examining
has been examined
is being examined
may have been examining
may have been examined
may be being examined
has been being examined
may have been being examined

There are four different words involved in this paradigm: _be may have_ and _examine_, and two of them are only examples of words which might fit the paradigm. Thus there would have to be lexical entries not only for _may-examine_, but for _shall-examine_, _could-examine_, and so on. The other major difference between periphrastic formations such as these is that since they are made up of more than one word, they can be reordered according to syntactic paradigms:

The specimen has been examined too closely.
Has the specimen been examined too closely?

4.3.2. Phrasal and Clausal Paradigms. It is also possible to organize higher-level sentence or textual material into paradigms as well. Some examples of such material might be

I hit him.
He was hit by me.
--
I see the dog right over there.
He sees the dog right over there.
I see the dogs right over there.
He sees the dogs right over there.
--
You are busy.
You are not busy.
Are you busy?

There are well known problems with dealing with certain sentence types as more 'basic' than others. This problem is avoided in FPG by virtue of the bidirectionality of its rules. On the other hand, a basic problem with an FPG network of sentence types would be that it says nothing directly about the interpretation of the sentences in any absolute sense (although it could relate a sentence to some abstract representation of 'meaning' easily enough). That is, there is no starting point. (Remember that in the inflectional system the citation form always provides a reference point).

4.3.3. Recursive FPG Networks.

FPG has a natural means of dealing with the nonlexical periphrastic or syntactic paradigm, which is quite similar to the way that a standard ATN grammar might work if it were bidirectional. The trick is that the rules on the arcs in the paradigm can refer to other paradigms in the system. This is very similar to the recursive 'seek' or 'push' arc in ATN-like grammars. Consider the following fragment of a possible FPG syntax in Icelandic:

```
BEGIN fake_np
        DERIVE plu FROM sing VIA +npplu
        DERIVE def FROM indef VIA +npdef .
```

This would handle forms such as

```
goður maður "(a) good man"
goði maðurinn "the good man"
goðir menn "(some) good men"
goða mennirnir "the good men"
```

if the rules +npplu and +npdef were defined appropriately. The rule +npplu would examine its input string and attempt to parse it as a noun phrase. If this succeeded, the next step would be to map each of the items in it to its corresponding plural form by accessing the inflectional net. The inflectional system would classify each input as to its category and attempt to derive the requested category. If all succeeded, then the transformation would have succeeded. The +npdef rule would operate in a similar way, but its request to the inflectional system would be to map the noun to its corresponding definite form and the adjective to its corresponding weak form. The higher-level paradigms would have to know about possible sequences of inflected categories.

What about node labels? Clearly, it would be insane to attempt to come up with a list of all the sentence types in a language and arrange them in a network. However, there is another approach. When a linguist studies a sentence or other unit, he becomes aware of many overlapping categories of ′things′ which are in the unit. For example, the sentence

Jack and Jill went up the hill

contains a conjoined NP which is the subject of the verb; the verb is inflected for past tense; the prepositional phrase marks goal of movement; the hill is not new information--the speaker assumes that it is either known or easily figured out by the hearer, and so on. These various components of the sentence are not necessarily arranged in any neat system of strata or in a hierarchy; instead they represent essentially different ways of looking at the sentence or some part of it. As the linguist studies the sentence, his point of view shifts over time. He recognizes one sort of perceptual object, then another, then another. The complex field consisting of all the memories of recognizing the objects is built up in the linguists′ mind. This process is very similar to the classical gestalt view of general perception (e.g. Köhler 1940). The point is that it would be possible to classify these perceptual objects and to organize them into FPGs, together with rules for transforming certain of them into others. Now, from a certain point of view, only some of the properties of the object under examination would be relevant. Thus, from the point of view of single-word inflectional morphology it would be irrelevant that the above example sentence comes from a nursery rhyme, yet this is a basic perceptual experience which comes from reading the sentence. Similarly, the fact that the subject np is conjoined may or may not have anything to do with the inflection of the verb, depending on the language and the sentence. But the overall system must be set up in such a way that the various paradigmatic systems can interact.

So that is the end of this speculative excursion. The basic idea is that a set of networks could be set up which would be able to access each other recursively. The system would be very redundant, like the human mind. The basic object in the system would not be the word, but the globs of relations among abstract objects represented as finite paradigms.

4.4. Conclusions. In this dissertation I have developed a theory of inflectional morphology which is relevant to various applications of linguistic theory, especially those based on the use of computers. The theory which was developed took the form of a set of finite networks interconnecting surface forms and intermediate forms via arcs with rules on them. The approach is superior to other approaches with which I am familiar because it does not make the error of perceiving language production and recognition as separate processes. Instead, the theory enforces a set of constraints which results in all of the system being bidirectional. A side effect of this is that many of the generalizations about phonological processes made in unidirectional theories cannot be carried over too well into FPG, because they result in ambiguity when the rules are reversed. Distinctive feature theory is actually one of the first constructs to be found wanting in this way. The reason for this is that in order to reverse a deletion rule without unbounded ambiguity, it is necessary that the entire set of items which can be deleted be listed, and it is usually the case that this is more difficult to do when distinctive feature are used than for lists of alternatives.

On the other hand, I am not happy with the linear input language used in this dissertation. The reason for this is that since the system was developed for use on a computer system which can represent any arbitrary relational subnetwork, it is very hard to develop a notational approach which is both readable by humans and which is sufficient to represent the required data objects. In several places in the analysis, the notation required very round-about means to be used. The approach which I would like to develop for future work would be built around a graphics system capable of displaying portions of the network in a pictorial representation. Information would be entered into the network using cursor addressing, and the network would be updated immediately. It would also be possible to have interactive verification of correctness for any particular paradigm, by generating the list of forms. If this approach were used, there would be no printable version of the FPG grammar. Instead, the machine representation of the system would be the grammar.

Another contribution made by this dissertation is a detailed synchronic analysis of the major classes of nouns and adjectives in Modern Icelandic. While it is not complete, it should be possible to build on it until eventually a complete analysis will result. Even in its present form, the analysis should be useful for learners of the language, and since it is a computer-based analysis, it should be possible to apply it to lemmatization, parsing, and translation of Icelandic. Since the differences between the written forms of the ancient and the modern languages are relatively minor, only small adjustments would be needed to create a similar analysis of Old Norse inflections.

A very exciting potentiality is that raised by the experimental results reported earlier in this chapter. I feel that it would be very interesting to develop a methodology whereby various aspects of an informants linguistic performance, including lexical decision tasks such as above, were available at all times during the analysis of a set of data. This information would essentially act as a guide as to which of the various alternative analyses corresponds most closely to the psychologically real factors of the informant. For example, a good way to test the Icelandic data would be to exhaustively test all pairs of items from each paradigm with native speakers. From this data an optimal network relating the forms could be constructed and the FPG analysis would then be constructed on that model.

Another innovation of this dissertation has to do with the concept of rule ordering. While traditional theories tend to see rule ordering as something outside the lexicon, FPG suggests that rule ordering is implicit in the structure of the inflectional paradigms, and therefore tied closely to lexical entries. Instead of using rule features or abstract phonological forms in the lexicon to guide an item through the set of rules, each item actually 'knows' all of the possible derivational paths which it can follow.

In the discussion of language change, it was shown that it is often the case that the shape of linguistic paradigms is what is changed, so that individual forms are often not of central importance. FPG allows whole paradigms or subparadigms to be represented and manipulated as single structures. This results in a shift of attention in the analysis of language change, especially with respect to those phenomena which have been analyzed as being due to analogical change. In particular, it was shown at least in a few cases that analogy seems to be based on similarities between portions of paradigms rather than on similarities between isolated surface forms. This, taken together with the experimental results, suggests that the finite paradigm is a psychologically real element of linguistic structure.

BIBLIOGRAPHY

Allen, John, _Anatomy of LISP_, McGraw-Hill, New York (1978).

Allen, Margaret Reece, _Morphological Investigations_, PhD Dissertation, University of Connecticut (1978).

Anderson, Stephen R., _West Scandinavian Vowel Systems and the Ordering of Phonological Rules_, Unpublished MIT doctoral dissertation (1969).

Anderson, Stephen R., "Icelandic u-umlaut and breaking in a generative grammar," pp. 13-30 in _Studies for Einar Haugen_, ed. E. S. Firchow, K. Grimstad, N. Hasselmo, and W. A. O'Neil, Mouton, The Hague (1972).

Anderson, Stephen R., "On the formal descripton of inflection," in _CLS 13_ (1977).

Anderson, Stephen R., "Where's morphology," _UCLA Occasional Papers_ Vol. 4, pp. 1-40 (1981).

Anisfeld, Moshe, "Psychological evidence for an intermediate stage in a morphological derivation," _J. verb. Learn. verb. Behav._ Vol. 8, pp. 191-195 (1969).

Anttila, Raimo, _Analogy_, Mouton, The Hague (1977).

Aronoff, Mark, _Word Formation in a Generative Grammar_, Linguistic Inquiry Monographs, Cambridge Mass (1976).

Bates, Madeleine, Syntactic Processing in the BBN Speech Understanding System, Harvard PhD Dissertation (1975).

Bierwisch, Manfred, "Syntactic features in morphology: general problems of so-called pronominal inflection in German," pp. 239-270 in To Honor Roman Jakobson Vol. I, Mouton, The Hague (1967).

Bobrow, Daniel G. and Terry Winograd, "An overview of KRL, a knowledge representation language," Xerox (1976).

Boot, M., "Homography and lemmatization in Dutch texts," ALLC Bulletin Vol. 8(2), pp. 175-189 (1980).

Bradley, Dianne, "Lexical representation of derivational relation," pp. 37-55 in Juncture: A Collection of Original Papers, ed. Mark Aronoff and Mary-Louise Kean, Anma Libri, Saratoga, California (1980). (Volume 7 in Studia Linguistica et Philologica, editor Alphonse Juilland)

Chomsky, Noam, Aspects of the Theory of Syntax, MIT Press, Cambridge, Mass. (1965).

Chomsky, Noam, "Deep structure, surface structure, and semantic interpretation," pp. 183-216 in Semantics: an Interdisciplinary Reader in Philosophy, Linguistics, and Psychology, ed. D. D. Steinberg and L. A. Jakobovits, University Press, Cambridge (1971).

Chomsky, Noam and Morris Halle, The Sound Pattern of English, Harper and Row, New York (1968).

Cleasby, Richard, Gudbrand Vigfusson, and William Craigie, An

Icelandic-English Dictionary, Clarendon Press, Oxford (1957). 2nd Edition

Dawson, J. L., "Suffix removal and word conflation," ALLC Vol. 2(3), pp. 33-46 (1974).

Derwing, Bruce L., "Psycholinguistic evidence and linguistic theory," in Perspectives in Experimental Linguistics, ed. Gary Prideaux, John Benjamins B. V., Amsterdam (1979). (Papers from the University of Alberta Conference of Experimental Linguistics at Edmonton, 13-14 October 1978)

Eggers, Hans, "Das Lemmatisierungssystem SALEM des Sonderforschungsbereichs 90 ´Elektronische Sprachforschung´ in Saarbrücken," ALLC Bulletin Vol. 9(2), pp. 9-15 (1981).

Einarsson, Stefán, Icelandic, The Johns Hopkins University Press, Baltimore (1945).

Eliasson, Stig, "Unstable vowels in Swedish: syncope, epenthesis, or both?," pp. 174-188 in Studies for Einar Haugen, ed. E. S. Firchow, K. Grimstad, N. Hasselmo, and W. A. O´Neil, Mouton, The Hague (1972).

Friedman, Joyce, "Computational Linguistics," pp. 718,756 in A Survey of Linguistic Science, ed. William Orr Dingwall, Linguistics Program/University of Maryland, College Park, Md (1971).

Fromkin, Victoria A., "When does a test test a hypothesis, or, What counts as evidence," pp. 43-64 in Testing Linguistic Hypotheses, ed. David Cohen and Jessica R. Wirth, Hemisphere Publishing Company, Washington (1975).

Garnes, Sara, "Breaking in Icelandic: the loss of a phonological rule," CLS Vol. 1, p. 385 (1971).

Geens, D., "Computational recognition of English syntactic structures," Cahiers de lexicologie Vol. 35(2) (1979).

Gibson, Eleanor J. and Lynne Guinet, "Perception of inflections in brief visual presentations of words," J. verb. Learn. verb. Behav. Vol. 10, pp. 182-189 (1971).

Gilbert, Leon and Rudolf Hirschmann, "Computer-aided indexes and concordances to Early German: a critical study," ALLC Bulletin Vol. 8(3), pp. 249-262 (1981).

Glanzer, Murray and S. L. Ehrenreich, "Structure and search of the internal lexicon," J. verb. Learn. verb. Behav. Vol. 18, pp. 381-398 (1979).

Gordon, E. V., An Introduction to Old Norse, The Oxford University Press, Oxford (1957).

Griggs, S. and C. M. Rulon, English Verb Inflection: A Generative View, Janua Linguarum S.M. 211 (1974).

Halle, Morris, "Prolegomena to a theory of word formation," Linguistic Inquiry Vol. 4(1), pp. 3-16 (Winter 1973).

Hann, Michael L., "Principles of automatic lemmatisation," Review of

Applied Linguistics Vol. 23, pp. 3-22 (1974).

Haugen, Einar, "On diagramming vowel systems," pp. 433-440 in Studies by Einar Haugen, ed. E. S. Firchow, K. Grimstad, N. Hasselmo, and W. A. O'Neil, Mouton, The Hague (1972a).

Haugen, Einar, "The phonemics of Modern Icelandic," pp. 355-389 in Studies by Einar Haugen, ed. E. S. Firchow, K. Grimstad, N. Hasselmo, and W. A. O'Neil, Mouton, The Hague (1972b).

Hellberg, Staffan, "Computerized lemmatization without the use of a dictionary," Computers and the Humanities Vol. 6, pp. 209-212 (1972).

Hooper, Joan B., "Rule morphologization in natural generative phonology," pp. 160-170 in Papers from the Parasession on Natural Phonology, ed. Anthony Bruck, Robert A. Fox, Michael W. LaGaly, Chicago Linguistic Society, Chicago (1974).

Horvath, Julia, "On the status of vowel patterns in Modern Hebrew: morphological rules and lexical representations," UCLA Occasional Papers Vol. 4, pp. 228-261 (1981).

Iversen, Ragnvald, Norrøn Grammatikk, H. Aschehoug & Co., Oslo (1937).

Jackendoff, Ray, "Morphological and semantic regularities in the lexicon," Language Vol. 51(3), p. 639 (1975).

Jarvella, Robert J. and Joan Gay Snodgrass, "Seeing ring in rang and retain in retention: on recognizing stem morphemes in printed words," J. verb. Learn. verb. Behav. Vol. 13, pp. 590-598 (1974).

Jastrzembski, James E. and Robert F. Stanners, "Multiple word Meanings and lexical search speed," J. verb. Learn. verb. Behav. Vol. 14, pp. 534-537 (1975).

Jóhannesson, Alexander, Die Suffixe im Isländischen, Max Niemeyer Verlag, Halle (Saale) (1927).

Johnson, Rowland, General Language Processor, Berkeley PhD Thesis (1976).

Justus, Carol F., "Syntactic change: evidence for restructuring among coexistent variants," JIES Vol. 6, pp. 107-132 (1978).

Kaplan, Ronald M., "A general syntactic processor," pp. 193-242 in Natural Language Processing, ed. R. Rustin, Algorithmics Press, New York (1973).

Kay, Martin, "The MIND system," pp. 155-188 in Natural Language Processing, ed. R. Rustin, Algorithmics Press, New York (1973).

Kay, Martin, "Syntactic processing and functional sentence perspective," pp. 12-15 in Theoretical Issues in Natural Language Processing, ed. Bonnie Nash-Webber and Roger Schank (1975).

Kiefer, Ferenc, "Morphology in generative grammar," pp. 265-280 in The Formal Analysis of Natural Languages, ed. M. Gross, M. Halle, and M-P. Schützenberger, Mouton, The Hague (1973).

Köhler, Wolfgang, Dynamics in Psychology, Liveright, New York (1940).

Köhler, Wolfgang, _Gestalt Psychology_, Liveright, New York (1947).

Kress, Bruno, _Laut- und Formenlehre des Isländischen_, Max Niemeyer Verlag, Halle (Saale) (1963).

Labov, William, _Sociolinguistic Patterns_, University of Pennsylvania Press, Philadelphia (1972).

Lakoff, George, "On generative Semantics," pp. 232-296 in _Semantics: an Interdisciplinary Reader in Philosophy, Linguistics, and Psychology_, ed. D. D. Steinberg and L. A. Jakobovits, University Press, Cambridge (1971).

Lakoff, George, "Linguistic gestalts," in _CLS 13_ (1977).

Lakoff, George and Henry Thompson, "Introducing cognitive grammar," pp. 295-313 in _BLS I_ (1975).

Leben, William R. and Orrin W. Robinson, "'Upside-down' phonology," _Language_ Vol. 53(1), pp. 1-20 (1977).

Lotz, J., "Relational and numerical characteristics of language paradigms," pp. 355-361 in _Papers in Computational Linguistics_ Vol. 91, ed. F. Papp and G. Szépe, Mouton (1978).

Lukatela, G., Z. Mandić, B. Gligorijević, A. Kostić, M. Savić, and M. T. Turvey, "Lexical decision for inflected nouns," _Haskins Laboratories Status Report on Speech Research_ Vol. 53(1), pp. 65-72 (1978).

MacKay, Donald G., "On the retrieval and lexical structure of verbs," _J._

verb. Learn. verb. Behav. Vol. 15, pp. 169-182 (1976).

MacKay, Donald G., "Derivational rules and the internal lexicon," J. verb. Learn. verb. Behav. Vol. 17, pp. 61-71 (1978).

MacWhinney, Brian, The Acquisition of Morphophonology, Child Development Monographs (1978).

Marcus, Mitchell P., A Theory of Syntactic Recognition for Natural Language, MIT Press, Cambridge, MA (1980).

Matthews, P. H., Inflectional Morphology, University Press, Cambridge (1972).

Matthews, P. H., Morphology, University Press, Cambridge (1974).

McCarthy, John, Paul W. Abrahams, Daniel J. Edwards, Timothy P. Hart, and Michael I. Levin, LISP 1.5 Programmer's Manual, MIT Press, Cambridge, Massachusetts (1965).

McCawley, James D., "English as a VSO language," Language Vol. 46, pp. 286-300 (1970).

Moore, Samuel, Historical Outlines of English Sounds and Inflections, George Wahr Publ. Co., Ann Arbor (1966).

Ohala, John J., "Experimental historical phonology," in Historical Linguistics II. Theory and Description in Phonology (Proceedings of the First International Conference on Historical Linguistics, Edinburgh, 2-7 September 1973), ed. J. M. Anderson and C. Jones, North Holland, Amsterdam (1974).

Ohala, John J. and Manjari Ohala, "Testing hypotheses regarding the psychological reality of morpheme structure conditions," Xerox (1975).

Petrick, S. R., A recognition procedure for transformational grammars, MIT PhD dissertation (1965).

Pike, Kenneth L., "Theoretical implications of matrix permutation in Fore (New Guinea)," Anthropological Linguistics Vol. 5(8) (1963).

Postal, Paul M., "On so-called ´pronouns´ in English," in Modern Studies in English, ed. D. A. Reibel and P. S. Rosenbaum, Prentice-Hall, Englewood Cliffs, NJ (1969).

Quirk, Randolph, Sidney Greenbaum, Geoffrey Leech, and Jan Svartvik, A Grammar of Contemporary English, Seminar Press, New York (1972).

Rosenberg, Richard S., "Artificial intelligence and linguistics: a brief history of a one-way relationship," pp. 379-392 in BLS 1 (1975).

Ross, John Robert, "On declarative sentences," in Readings in English Transformational Grammar, ed. R. A. Jacobs and P. S. Rosenbaum, Ginn, Waltham, MA (1970).

Rubin, Gary S. and Curtis A. Becker, "Morphological structure and its effect on visual word recognition," J. verb. Learn. verb. Behav. Vol. 18, pp. 757-767 (1979).

Sågvall, Anna-Lena, A System for Automatic Inflectional Analysis Implemented for Russian, Almqvist and Wiksell, Stockholm (1975).

Shenaut, Gregory K., Presupposition and negation as nondiscrete semantic and syntactic variables, Michigan State University MA thesis (1975a).

Shenaut, Gregory K., "Valves: plumbing the presuppositional depths (or, what's a plug like you doing in a hole like this?)," pp. 498-513 in CLS 11 (1975b).

Siegel, Dorothy, Topics in English Morphology, Garland Press (1974 MIT PhD Diss.), New York (1979).

Slobin, Dan I., "Cognitive prerequisites for the development of grammar," pp. 175-208 in Studies of Child Language Development, ed. C. A. Ferguson and D. I. Slobin, Holt, Rinehart, and Winston, New York (1973).

Spencer, N. J., "Differences between linguists and non-linguists in intuitions of grammaticality-acceptability," J. Psycholinguistic Res. Vol. 2(2), pp. 83-98 (1973).

Spoehr, Kathryn T., "Phonological encoding in visual word recognition," J. verb. Learn. verb. Behav. Vol. 17, pp. 127-141 (1978).

Stanners, Robert F., James J. Neiser, William P. Hernon, and Roger Hall, "Memory representation for morphologically related words," J. verb. Learn. verb. Behav. Vol. 18, pp. 399-412 (1979).

Stanners, Robert F., James J. Neiser, and Scott Painton, "Memory representation for prefixed words," J. verb. Learn. verb. Behav. Vol. 18, pp. 733-743 (1979).

Taft, Marcus, "Lexical access via an orthographic code: the basic orthographic syllabic structure (BOSS)," J. verb. Learn. verb. Behav. Vol. 18, pp. 21-39 (1979).

Taft, Marcus and Kenneth I. Forster, "Lexical storage and retrieval of prefixed words," J. verb. Learn. verb. Behav. Vol. 14, pp. 638-647 (1975).

Vennemann, Theo, "Rule inversion," Lingua Vol. 29, pp. 209-242 (1972).

Vennemann, Theo, "Words and syllables in natural generative grammar," pp. 346-374 in Papers from the Parasession on Natural Phonology, ed. Anthony Bruck, Robert A. Fox, Michael W. LaGaly, Chicago Linguistic Society, Chicago (1974).

Weissman, Clark, LISP 1.5 Primer, Dickenson, Belmont (California) (1967).

Wickelgren, Wayne A., "Network strength theory of storage and retrieval dynamics," Psych. Rev. Vol. 83(6), pp. 466-478 (1976).

Wilbur, Ronnie Bring, "When is a phonological rule not a phonological rule? The morphology of Sierra Miwok," pp. 385-395 in Natural Phonology, CLS (1974).

Winograd, Terry, Understanding Natural Language, Academic Press, New York (1972).

Winston, Patrick Henry and Berthold Klaus Paul Horn, LISP, Addison-Wesley, Reading. Massachusetts (1981).

Wirth, Jessica, "Logical considerations in the testing of linguistic hypotheses," pp. 211-220 in Testing Linguistic Hypotheses, ed. David Cohen and Jessica R. Wirth, Hemisphere Publishing Company, Washington (1975).

Woods, William A., "Procedural semantics for a question-answering system," pp. 457-471 in Proc. AFIPS Fall Joint Comp. Conf. Vol. 33(1), MDI Publ, Wayne, Pa (1968).

Woods, William A., "Transition network grammars for natural language analysis," CACM Vol. 13(10), pp. 591-606 (1970).

Woods, W. A., "An experimental parsing system for transition network grammars," pp. 111-154 in Natural Language Processing, ed. R. Rustin, Algorithmics Press, New York (1973).

Zimmer, Karl, "Psychological correlates of some Turkish morpheme structure conditions," Language Vol. 45(2), pp. 309-321 (1969).