THE RESOLUTION OF AMBIGUITIES AND THE CORRECTION OF ERRORS IN THE AUTOMATIC TRANSCRIPTION OF PALANTYPE

by

ALEXANDER W. BOOTH

A Thesis
presented to the
Council for National Academic Awards
in partial fulfillment of the
requirements for the degree of
DOCTOR OF PHILOSOPHY

Leicester Polytechnic, May 1982

in collaboration with

The National Physical Laboratory

ABSTRACT

The computer transcription of machine shorthand code into English text has been a subject of research for more than 20 years. The emphasis during this period has been to develop systems which, by connecting shorthand machines to a computer, are able to output English words only fractions of a second after they have been spoken. In previous years the British system of machine shorthand, Palantype, has been investigated for use in such applications as transcription of House of Commons proceedings, law court proceedings, computer typesetting and more recently for the automatic subtitling of television to benefit hearing impaired viewers. In most of these applications the quality of English text has been impaired by a combination of the problems associated with computer transcription of Palantype code and the goal of real time transcription. Although the speed and efficiency of computers have increased dramatically over the last 20 years, the goals of "real time" transcription and high quality output text have been difficult to achieve because of the errors and ambiguities present in speech itself, and those introduced by the recording process. In the present research these problems have been examined in detail for dictionary-based Palantype transcription systems, and new methods have been developed to attempt to overcome them.

These 'new' methods have not had their development hampered by attempting to work within the restrictions imposed by a "real time" goal. The major contributions made to the knowledge of this subject can be summarised as follows:-

1. A partial solution to the word reconstitution problem

2. An analysis of output from Palantype transcription systems and operator errors

3. The development of new error correcting methods which in some cases include syntax analysis

4. The development of a system which is capable of producing higher quality output text (not in real time) than has previously been achieved by a Palantype transcription system

5. The specification of a framework of a parser particularly suited to parsing the output from Palantype transcription systems

6. A conclusion which states that a Palantype transcription system to produce very high quality text output requires the organisation of and access to many knowledge sources in a similar way to speech recognition systems

ACKNOWLEDGMENTS

I am indebted to many people for many things including technical advice and feedback, provision of Palantype input data, proof reading of this thesis, patience and understanding, to name but a few. I would like to express my gratitude to all those people for their much appreciated efforts.

E.A.Edmonds, W.L.Price, The team of researchers at Southampton University, The Palantype Organisation, Miss I.Beard and other Palantypists, Leicester Polytechnic, The S.E.R.C., The B.B.C., S.P.Guest, S.A.R.Scrivener, P.A.Leadbetter, P.Innocent, N.J.Thompson, J.Connolly, and not least of all Gillian M. and Tamara J. Booth (my wife and daughter).

CONTENTS

Chapter 1

INTRODUCTION

1.1 SCOPE

The computer transcription of machine shorthand code into English text has been a subject of research for more than 20 years. The emphasis during this period has been to develop systems which, by connecting shorthand machines to a computer, are able to output English words only fractions of a second after they have been spoken.

It is reasonably well known that mechanical shorthand machines can be used to record verbatim proceedings of conferences, committees and law courts etc. The British system for achieving this is known as Palantype (Beard,1978) and the American system is called Stenotype (Stenograph, 1976). The output from the shorthand machine is a kind of phonetic code, and is not readily intelligible to the uninitiated reader. For example, the word "believe" is represented in Palantype by "P+LI.F+". At first glance, there seems to be little correspondence between the two, but when one learns that in Palantype "P+" stands for "B", "F+" for "V" and "I." for the double "E" sound (ee), one finds that "BLEEV" is phonetically similar to "believe".

After a recording session, an operator normally transcribes his own output into English words. Although this is not a difficult task, it is very time consuming; typically 4 to 5 times longer than the recording stage. Therefore, if an operator has been recording for several hours in the law courts, for example, it may be two or three days before transcripts of the proceedings are available.

Besides being a valuable time saver for Palantype operators, there are several other applications for computer transcription systems; the most recent one being the automatic generation of television subtitles to benefit deaf and hard of hearing viewers. In most of these applications (which are described in chapter 2), the quality of English text has been impaired by a combination of the problems associated with computer transcription of Palantype code and the goal of real time transcription. Although the speed and efficiency of computers have increased over the last 20 years, the goal of "real time" transcription has been difficult to achieve because of the errors and ambiguities present in speech itself, and those introduced by the recording process. These problems have been examined in detail and in some cases 'new' methods have been developed to attempt to overcome them.

These 'new' methods, described in this thesis, for transcribing Palantype code into English text have not had their development hampered by attempting to work within the restrictions imposed by a "real time" goal. They have been developed, in the short term to reduce to a minimum the effect of errors and ambiguities, and in the long term to provide a framework for a system which incorporates several knowledge sources, in a similar way to a model of linguistic performance,to help resolve errors and ambiguities in Palantype transcription systems.

1.2 BACKGROUND

1.2.1 Machine shorthand

Machine shorthand is very different from conventional forms of hand written shorthand and although not as widely practised, is a direct alternative to them. Palantype is the British system of machine shorthand, (equivalent to the American Stenograph system), and is used for recording verbatim proceedings of conferences, committees, law courts, etc. It employs a small portable machine with a keyboard of 29 keys. The keys are laid out in three zones (fig.1A), 12 consonant keys on the left operated by the fingers of the left hand, 12 consonant keys on the right operated by the fingers of the right hand, and 5 vowel keys in the centre of the keyboard operated by the 2 thumbs.

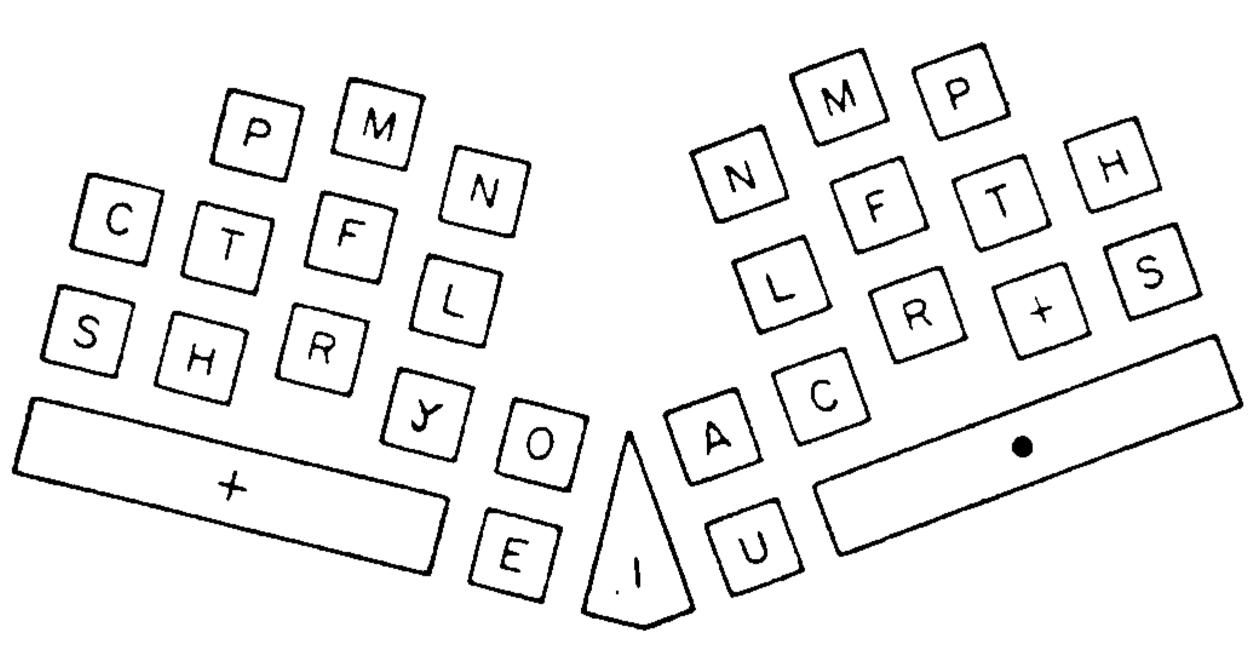


Fig. 1A Layout of Palantype Keyboard

Any set of keys may be depressed simultaneously to form what is known as a chord, and each chord pressed is printed on a paper roll in such a way that the order of the keys across the keyboard is preserved, i.e. when the left hand 'S' is present in a chord, it appears in a discrete position on the printed roll. In phonetic terms a Palantype chord usually corresponds fairly closely to the spoken syllable, for example, the word 'capillary' is represented by the chords 'CA' 'PIL' and 'RI', and the word 'funnel' by the chord 'FUNL'.

It will have been noted that the keyboard does not contain a key for each letter of the alphabet. The way in which the missing consonants are represented in Palantype is by depressing 2 or 3 keys simultaneously (along with any other keys which might be in the chord). For example, to obtain a representation for B, D, G, V and Z, the '+' key is pressed simultaneously with P, T, C, F or S respectively. Fig.1B shows the form in which the output appears on a printed roll. In fig.1B, the orthographic equivalent is given so that the reader may gain some insight into the coding conventions used in Palantype. A description of the major coding conventions is given in Chapter 2.

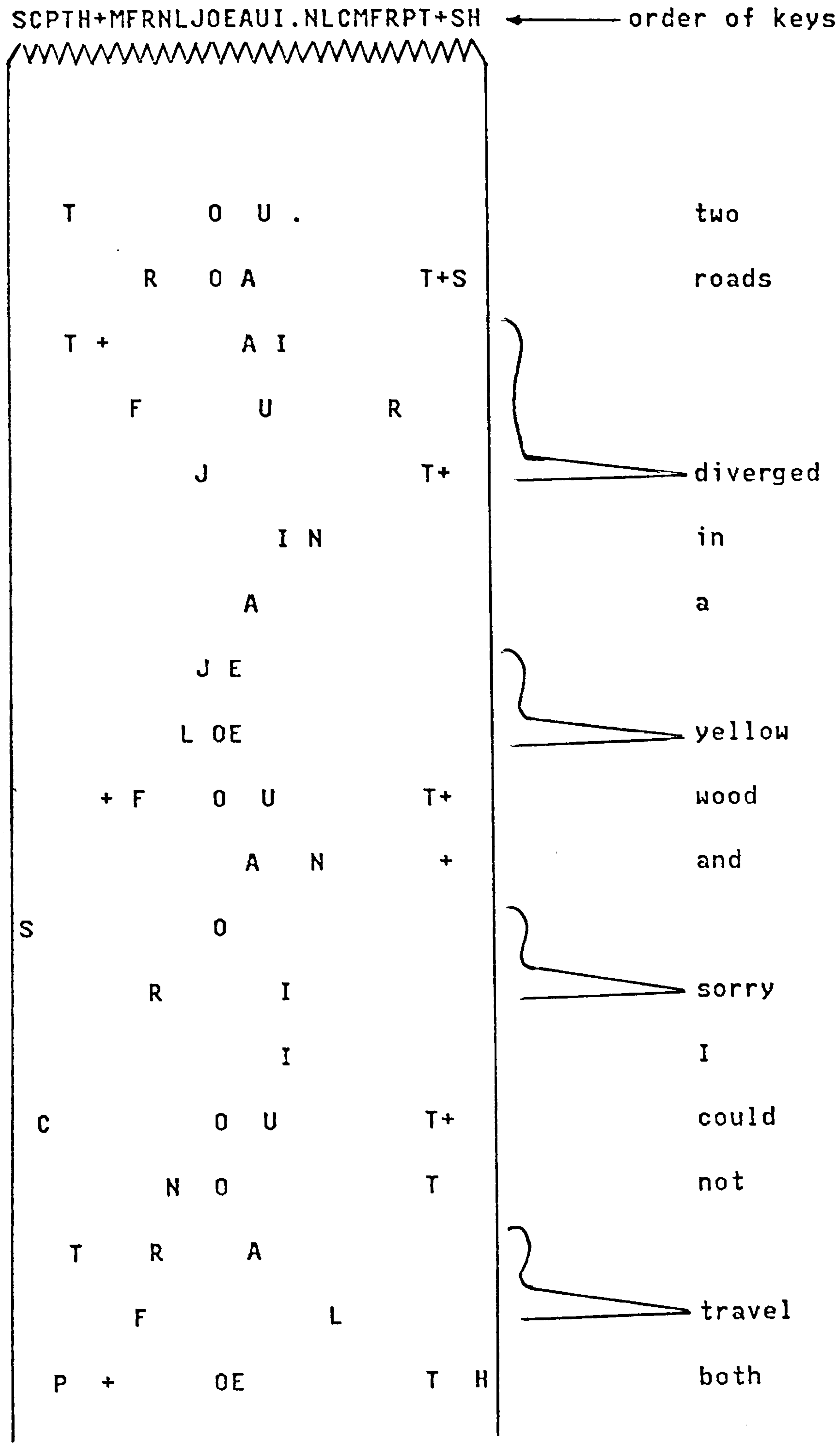


Fig. 1B Palantype roll with orthographic equivalent

1.2.2 Manual transcription

The output from the Palantype machine is not readily intelligible to the uninitiated reader (as with hand written shorthand), and is taken away at the end of a recording session to be transcribed into plain language. Operators have no difficulty in transcribing shorthand code into English text because human transcribers can draw (usually subconsciously) on many knowledge sources, e.g. articulatory, phonetic, lexical, syntactic and semantic, to help them resolve errors and ambiguities present in speech itself and those introduced by the recording process. One disadvantage of the manual system is that a transcript of several hours proceedings may not be ready for several days, during which time the operator who is performing the transcription is not available for further recording.

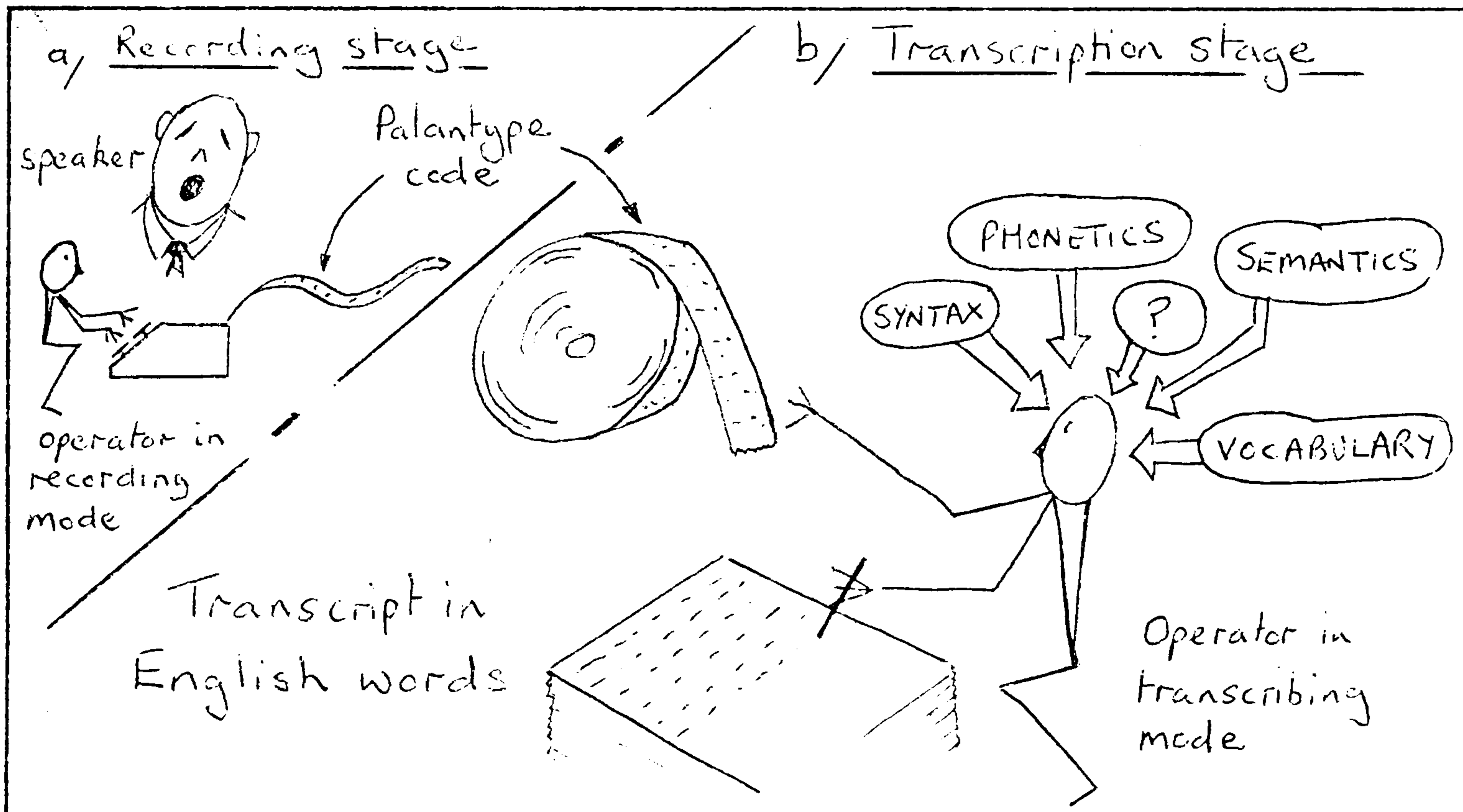


Fig. 1C Manual Transcription System

1.2.3 Computer transcription

There have been many different kinds of computer transcription systems. In those applications where the goal has been "real time" transcription, e.g. visual display for deaf people, there has been little or no processing time available for syntax or context analysis. This is because most of the available processing time is spent transforming the Palantype code into English text. In " real time" systems, an aim has been to make the recording and transcription stages concurrent.

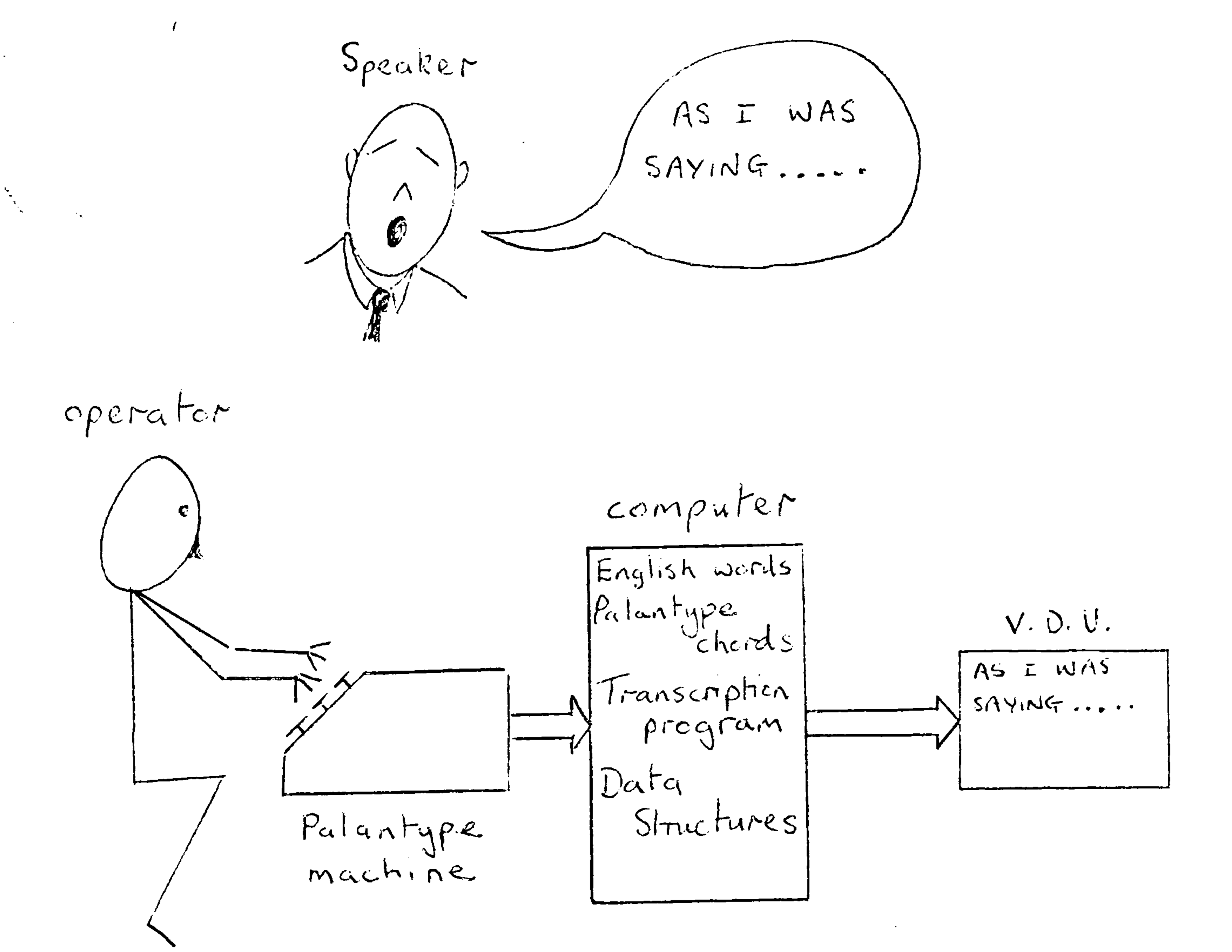


Fig. 1D Concurrent Recording and Transcription Stages

The limited amount of syntax and context checking which can be done in real time (Szanser,1971), is described in Chapter 2.

In those applications where the goal has not been real time transcription, e.g. transcripts of proceedings, an emphasis has been put on post transcription editing facilities rather than automatic correcting routines.

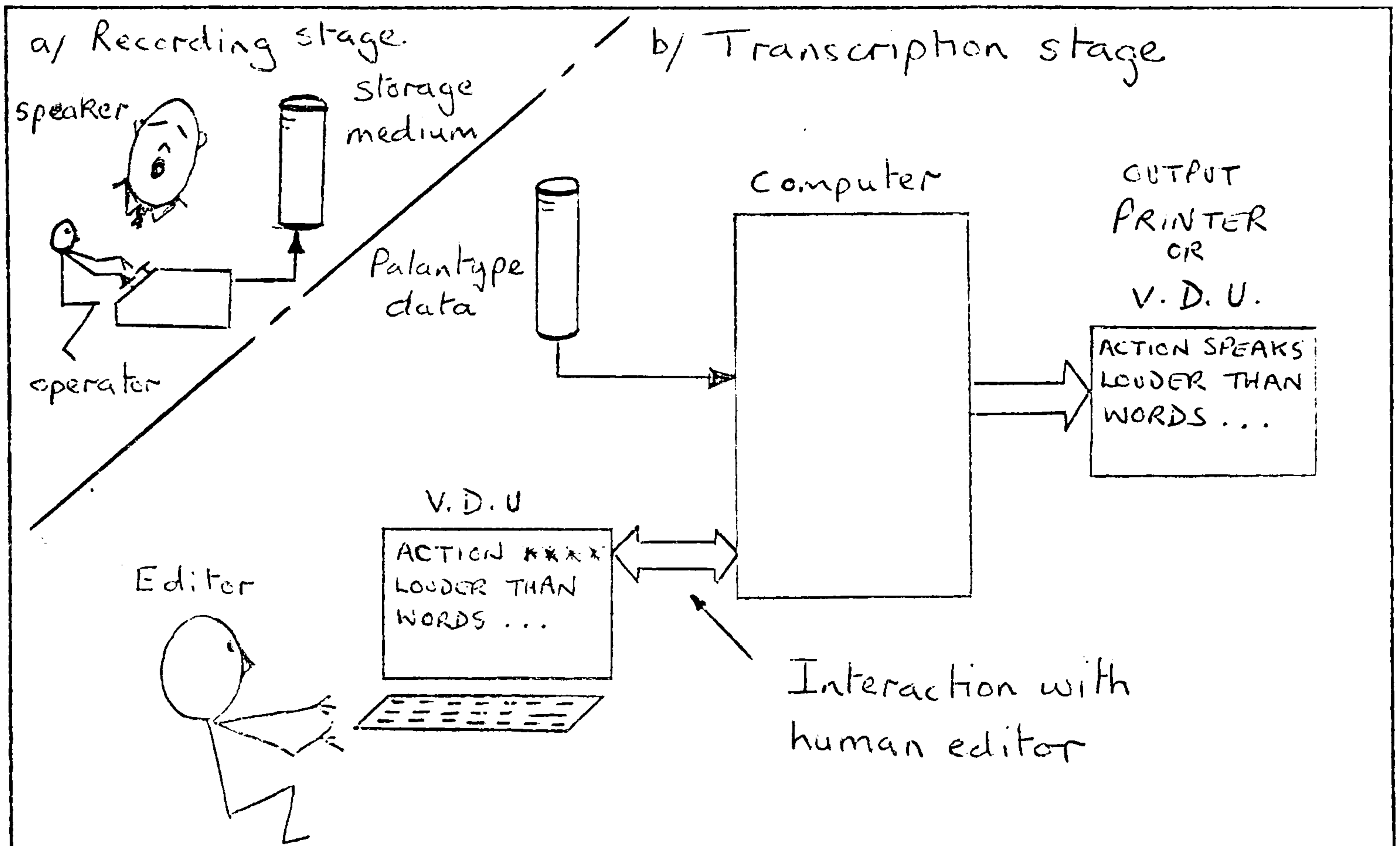


Fig. 1E Separate Recording and Transcription Stages

An example of a computer transcription system which attempted to output English text in real time is the one developed at Leicester Polytechnic (Booth & Barnden,1979). The work that went into developing the first Leicester system became the foundation for the author's research, and therefore it is worth describing briefly the organisation of the transcription system at Leicester.

1.2.4 The transcription system at Leicester

At Leicester, an English dictionary of over 75,000 words and their Palantype equivalents are stored on disk. This dictionary was provided by the National Physical Laboratory and was used by them on an earlier project (Price, 1971). Table 1 shows the first few words of the dictionary with their Palantype equivalents.

English words	corresponding Palantype chords		
a	/A		
aback	/A	P+/AC	
abacus	/A	P+/A	C/US
abandon	/A	P+/AN	T+/N
abandons	/A	P+/AN	T+/NS

TABLE 1

Note that in table 1, the symbols to the left of the slash (/) denote those keys operated by the left hand, and the symbols to the right of the slash denote those keys operated by the right hand.

An electronic interface between a Palantype machine and a Burroughs 6700 computer was designed and built (a task performed by the author), so that chords pressed on the Palantype keyboard could be processed by programs on the computer. A Burroughs 6700, which is a large computer, i.e. 1.25 Mbytes of electronic memory and 250 Mbytes of hard disk store, was used only because it was convenient. However, the transcription system is not complex and was small enough to be implemented in another version on a micro-computer (Thomas,1980) with a Palantype machine modified to be ergonomically acceptable to the BBC (Hawkins,1980). The transcription program was capable of transcribing Palantype chords into English words and displaying them on a visual display unit at speeds which were adequate to keep up with fast speech and the speediest of shorthand machine operators. Fig.1F shows a block diagram of the first Leicester system, and fig.1G is a typical example of the output produced by that system.

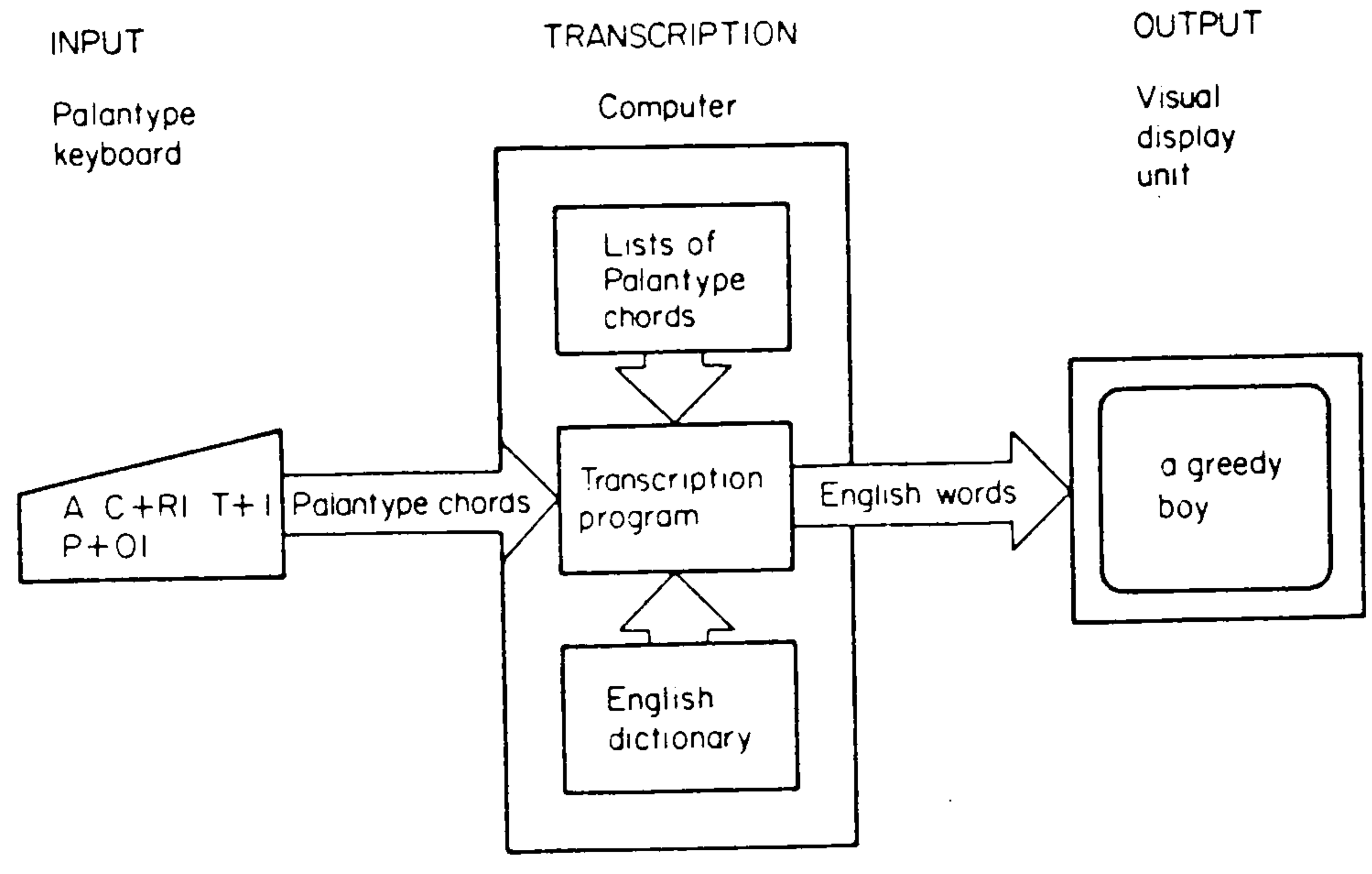


Fig. 1F The Transcription System at Leicester

It is clear from examining the text in fig.1G that the Leicester system had several shortcomings; some of these were due to the problems associated with shorthand code to English text transcription. In the next section these problems are identified and the chapters which describe the author's attempt at resolving them are indicated.

Many people ask about subtitling on T V and I get bags of letters a bat that sup ject, why not more of them. But abroad castors sometimes argue that the general public dont want to see subtitles on their programmes. Apart from that, it takes a very long time to subtitle a programme. But the B B C has een working with a couple of universities developing anew way of subtitling to overcome these objections. Now its a ri ec sa tn development, and all tho we mustn't raise false hops because its still experimental and therefore we can't show it now, it is based n what you saw earlier in the programme. It would use just one Palantype operator linked to a computer, and that food produce written English instantly on the screen. It could be used fr live programmes wi that a great deal of preparation. To cope with the other prop lem, subtitles would not have to appear on every ptis screen, own li of those who wanted to see them.   So that food remove the jen rl objection to subtitles.   That its possibility bl because of another B B C invention cold see facs which you will be able to see den stray ted in next weeks programme which was also recorded in nineteen seven five pfor these x pear ments on subtling began.

Fig. 1G   Example of Output from the Leicester System

1.3 THE PROBLEMS OF TRANSCRIBING PALANTYPE CODE INTO ENGLISH TEXT

Although many hurdles have been overcome to facilitate the transcription of Palantype code into English text by computer, some problems still remain.

1.3.1 Operator keying mistakes

The major factor affecting the quality of English text produced by Palantype transcription systems is the number of keying mistakes made by the machine operator. Recording verbatim means that the operator has sometimes to work at speeds in excess of 200 words per minute. It was shown (Price,1971) that as the recording rate increased, so did the operator's error rate. The transcription program at Leicester on encountering erroneous chords which failed to match up with English words, resorted to subjecting them to a simple pseudo phonetic translation procedure, so that for example, the chord 'P+RI.' would be output as 'BREA'. This procedure is described in a later chapter.

The present research has tried to keep the need for such translation to a minimum by attempting to correct automatically any erroneous chords. A description of tools developed to do this can be found in chapter 4, and the correction procedures in which the tools were used are described in chapter 6 and chapter 8.

1.3.2 Word boundaries

Transcribing machine shorthand code into English text is made more difficult by the fact that there is nothing in a string of chords to indicate where one word ends and the next one begins, i.e. there is no explicit 'end of word' marker to guide the transcription process. Although the transcription program which produced the output in figure 1G produces output consisting wholly of English words from error-free input almost all the time, examples can be contrived which would cause it to fail. An algorithm which always produces output consisting wholly of English words from error-free input has an important implication for research into error correction, and that is as follows. Once the algorithm outputs anything other than English words, it is known that there is at least one error in the input string. It is only when the presence of an error has been detected that procedures can be invoked to correct it. Therefore, some research has been done on developing an algorithm which always produced output consisting of English words from error-free input. This research is described in Chapter 3.

The word boundary problem can be viewed as two separate problems. The first one has already been mentioned; that of producing output consisting wholly of English words from error-free input. The second problem arises because, even when the output consists wholly of English words, the words have not necessarily been partitioned at the correct word boundaries. For example, the chords /A N/EU S/I T/IN, can be transcribed equally well in three different ways:- "a new sitting", "anew sitting" or "a newsy tin". Chapter 3 describes an algorithm which is capable of generating all alternative output strings for a given input string.

1.3.3 Homophonic ambiguities

As Palantype is a phonetic shorthand, words which sound the same are Palantyped identically. For example, the words 'for', 'four' and 'fore' are represented in Palantype by the single chord 'F/OR' . Of the 75,000 words in the dictionary used by the Leicester system, almost 1,000 are one of a pair or triad of homonyms. In the Leicester system, the designers were faced with one of two options; outputting all the alternative homonyms separated by a stroke (/), or outputting the word which was felt to be most commonly occurring in everyday spoken English. Consider, 'The for/four/fore ladies sat down for/four/fore tea' or 'The for ladies sat down for tea'. The latter option was chosen at Leicester, but it is clear that the resulting text is not satisfactory. Chapter 9 describes a method for resolving homophonic ambiguities using syntax analysis.

1.3.4 Keyboard design

One cause of operator keying mistakes might be the design of the keyboard, both ergonomically and physically. It was shown (Price,1971) that the design of the Palantype keyboard was ergonomically poor since for example, the weakest fingers (commonly the third and fourth of the left hand) have the most work to do. No effort has been made in the present research to improve this aspect of the transcription system.

With an emphasis on preventing errors rather than correcting them, Downton et al (1979) have had some success by improving the physical operation of the Palantype keyboard with the introduction of electronic components. Some work has also taken place at the B.B.C. (Hawkins & Robinson, 1979) in developing an electronic Palantype keyboard. No work of this kind has taken place in the present research.

1.3.5 Incomplete lexicon

No matter how large the dictionary becomes, the operator will always encounter words which are not in it. Consequently, when the chords which represent those words are pressed, the transcription program will fail to find a match for them and will therefore assume that there is an error in the input string, even though there may not be. Special program features to help reduce the effect that such words have on the output text are described in Chapter 5.

1.3.6 Error and ambiguity present in speech

As the Palantype operator is recording verbatim with little or no time for on-line editing, errors or ambiguities in speech are transferred through the transcription system to impair the quality of the output text. Inaccurate or hasty articulation can cause words or parts of words to sound quite different from what was intended. Also, a very common 'error' in everyday spoken English is that sentences (and sometimes words) are left unfinished. These factors have been considered in the goal of defining a 'grammar' particularly suited to parsing the output from Palantype transcription systems.

1.4 THE AIMS OF THE PRESENT RESEARCH

Having considered the problems associated with Palantype code to English text transcription it was decided that the aims of the research would be as follows:-

1. To investigate the work done previously in the field and also the work currently being undertaken.

2. To develop an algorithm which produces output consisting wholly of English words from error-free Palantype input.

3. To reduce to a minimum the effect of operator error by developing detection and correction tools to use in automatic correction procedures.

4. To define grammatical rules which would take into consideration the unusual 'structures' found in the output of Palantype

transcription systems, e.g. incomplete sentences, incomplete or missing words, minimal punctuation and homophones etc. Parsing the output from Palantype transcription systems poses a different set of problems to parsing "book type" English text because of the factors mentioned above. The problem is not only one of defining rules to cope with these many anomalies but also what to do when multiple errors are encountered and what actual 'unit' or part of the output string should be submitted for syntax analysis. The rules should be expressed in such a way as to 'drive' an already existing parser.

5. To incorporate syntax analysis in procedures to correct errors and resolve ambiguities. When an attempt has been made to correct an erroneous chord, syntax analysis will be used to test whether that chord forms part of a word which forms part of a grammatically correct phrase. In the resolution of homophonic ambiguities, syntax analysis will be used to test which of the alternative homophones form part of a grammatically correct phrase.

6. To investigate speech recognition systems and elicit any useful information in the way of parsing techniques and the organisation of knowledge sources in models of linguistic performance.

The programs which have been written and the way in which they interact shall now be described in outline.

1. A program to perform word boundary reconstitution which is also capable of producing output consisting wholly of English words from error-free input, and is capable of producing all alternative output strings for a given input string.

2. A subroutine which produces a list of chords which are a given 'distance', in terms of keys pressed (see Chapter 4), from a chord passed to the subroutine by a correction procedure.

3. A subroutine which orders the list in point 2 based on criteria elicited from an analysis of operator errors.

4. A program to correct errors by detecting the presence of an error in the input string, assuming a chord to be in error, 'calling' the subroutines in 2 and 3 and substituting the erroneous chord with the chord most likely intended by the operator. This program uses only the chords in the input string which precede the erroneous chord to help it correct the error.

5. The same as 4 except that this program looks ahead of the erroneous chord by up to five chords in the input string to help it correct the error.

6. The same as 5 except that this program uses syntax analysis to decide whether the attempted correction is acceptable, i.e. whether the corrected chord forms part of a word which forms part of a grammatically correct phrase.

7. A program which uses syntax analysis to determine which of a list of alternative homophones form part of a grammatically correct phrase.

8. As well as defining the rules for an existing table driven parser, certain subroutines had to be written which were called by the parser.

Chapter 2

THE AUTOMATIC TRANSCRIPTION OF MACHINE SHORTHAND : A REVIEW

2.1 A BRIEF HISTORY OF MACHINE SHORTHAND

There have been many systems of machine shorthand, but the first, which was French, was used as early as 1827 and this pre-dated typing by 40 years (see Beard,1978). The American system of machine shorthand, Stenograph, incorporates a machine that was originally developed by a shorthand reporter, W.S. Ireland, in 1914. It was patented in 1920. The British system of machine shorthand, called Palantype, was adapted from the French Grandjean system and was patented in 1939, although the production of Palantype machines did not commence until after the second world war. For an account of the history and progress of the Palantype machine see Newell,1978.

2.2 A COMPARISON OF PALANTYPE AND STENOTYPE

Palantype and Stenotype are similar in as much as they both employ a portable machine consisting of a keyboard and a paper roll on which the output is printed by a set of mechanical print hammers. Also, they are both intended for speech to be recorded syllable-by-syllable by pressing a number of keys simultaneously. Any such depression is known as a "chord" in Palantype and a "stroke" in Stenotype. Where they differ, and significantly so, is in the layout of the keys and the recording conventions used. Figures 2A and 2B show the layout of the Palantype and Stenotype keyboards respectively. The Palantype keyboard has 29 keys, whereas the Stenotype keyboard has only 24 keys; both machines have a set of initial consonants keys on the left, final consonant keys on the right, and a set of vowel keys in the centre of the keyboard. It can be seen that on neither keyboard is the full alphabet represented, and therefore to obtain the missing letters (and sounds, since the spelling of syllables is phonetically derived), coding conventions have to be used. Figures 2C and 2D are taken from Arnott et al, 1979, and show the major coding conventions used in Palantype and Stenotype. It is fairly clear that Stenotype has the more complex conventions of the two. For a comprehensive study of the differences between Palantype and Stenotype systems, see Arnott et al.

The remainder of this chapter describes various transcription systems and attempts at resolving some of the problems stated in the previous chapter. Although work on the transcription of Palantype has implications for Stenotype systems and vice-versa, the two are reported separately here for reasons of clarity and readability.

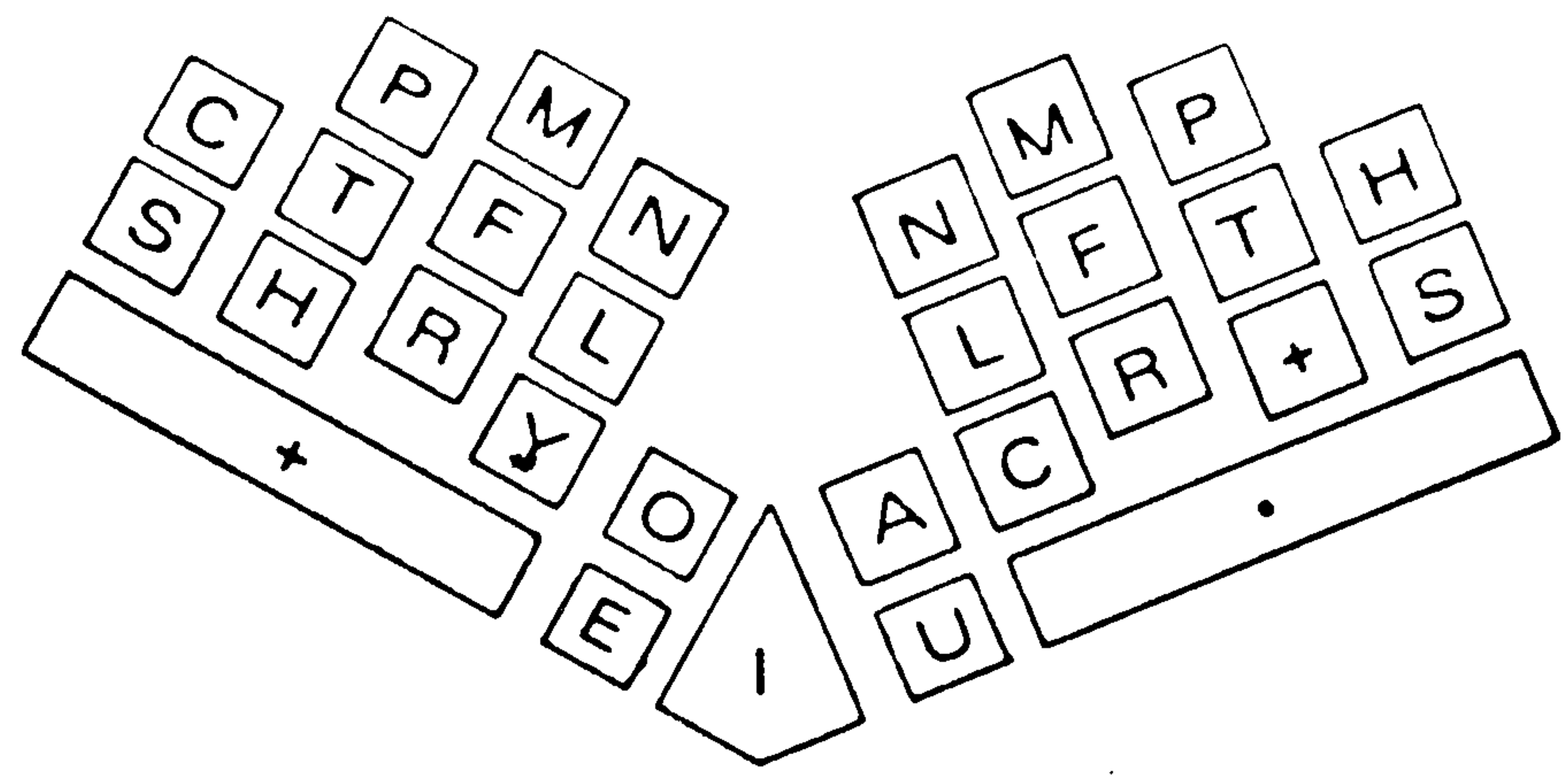


Fig. 2A Layout of Palantype Keyboard

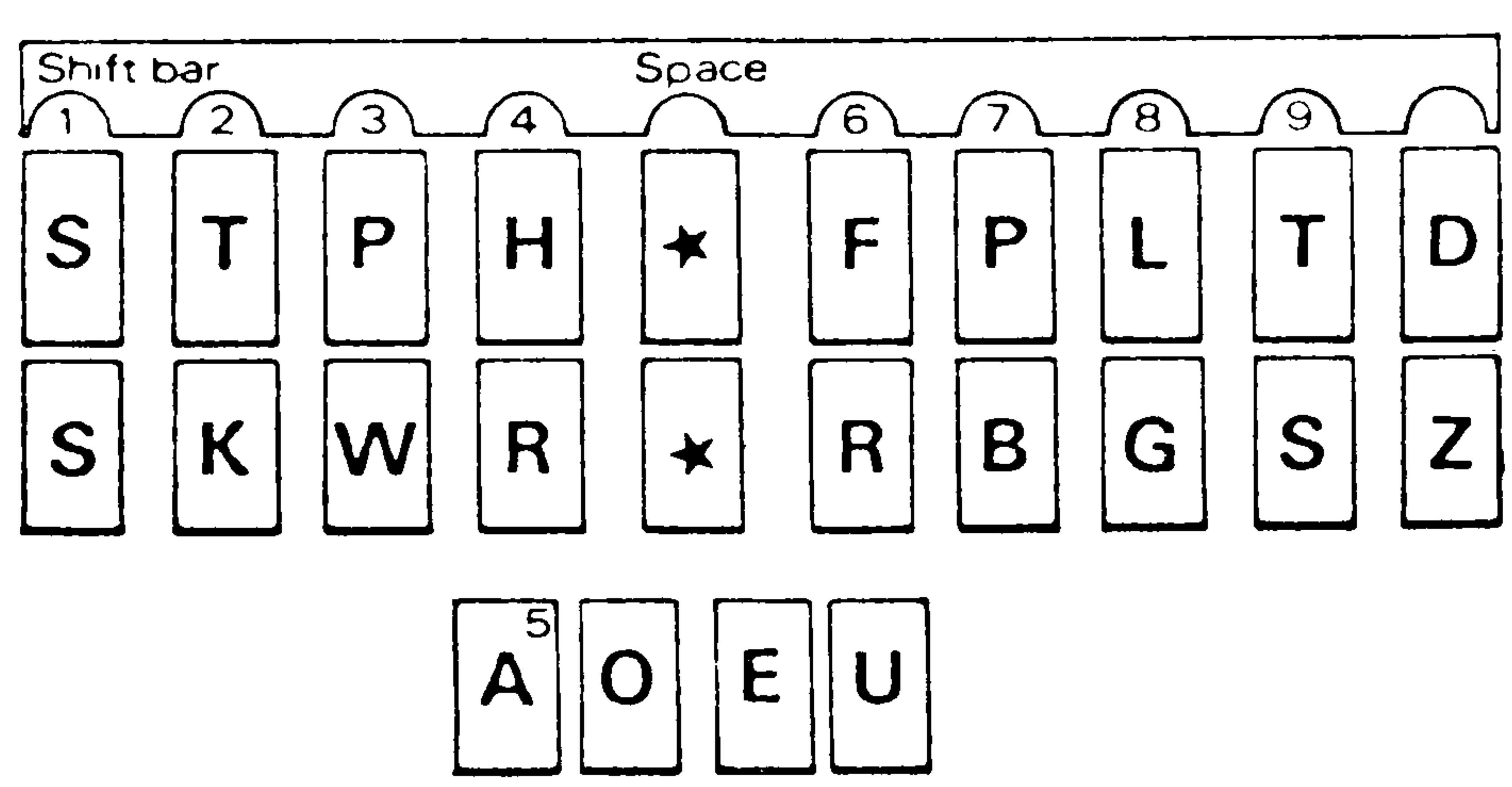


Fig. 2B Layout of Stenotype Keyboard

Initial Letters Palantype	Equivalent Sound	Final Letters Palantype	Equivalent Sound
C+	GATE	C+	DOG
MF	VOW	F+	LOVE
P+	BOW	P+	WEB
T+	DASH	T+	GLAD
Y+	JET	N+	WING
+F	WET		
HF	WHEN		
CF	QUICK		

Vowels

Palantype	Sound	Palantype	Sound
A.	FARM	E.	LAID
I.	MEAN	O.	LAW
OU	LOOK	OE	LOAN
AI	FIND	AU	NOW
EU	VIEW	U.	WORK

Punctuation

Palantype	Equivalent	Palantype	Equivalent
.	.	+	,
LNT	?	ULFTS	"Error" sign

Fig. 2C Major Coding Conventions of Palantype

Initial Letters Stenotype	Equivalent Sound	Final Letters Stenotype	Equivalent Sound
K	CODE	PB	CAN
S	CITY	BG	KNOCK
S	ZOO	PL	SLIM
TPH	NOW	F	LOVE
HR	LOW	FP	BIRCH
SKWR	JAZZ	PBLG	BADGE
TK	DONE	RB	FISH
KWR	YES	GS	MOTION
PH	MANY	GS	FORMATION
TP	FAST	GS	SPECIAL
PW	BEST		
KH	CHANCE		
SR	VOW		
TKPW	GATE		

Vowels

Stenotype	Sound	Stenotype	Sound
EU	BIT	EU	ANY
AOEU	NIGHT	AEU	MAID
AOU	NEW	OE	MOAN
AOE	MEAN	AO	NOON

Punctuation

Stenotype	Equivalent	Stenotype	Equivalent
FPLT	.	STPH	?
RBGS	,	.	"Error" sign

Fig. 2D Major Coding Conventions of Stenotype

2.3 THE AUTOMATIC TRANSCRIPTION OF STENOTYPE

Salton (1959), realised at the time of his project that using speech directly as an input for data processing purposes was not going to be possible for some years to come. He suggested therefore that since shorthand machine operators were trained to record spoken information at high speed, it might be possible to solve a substantial part of the problem of transforming spoken information into written form by using the output from shorthand machines as the input to a computer performing the transcription process. A study was made of methods for transcribing the output of the Stenotype machine.

Salton stated that the problems raised by the transcription of machine shorthand were identical to problems encountered in the automatic translation of languages with two important exceptions.

> In machine shorthand, it is not necessary to change the word order of the input text, or to alter a sentence by insertion or deletion of words; idiomatic expressions do not therefore cause any difficulties. On the other hand, in language translation it is not necessary to generate correct word forms by altering the spelling of words, or to recognise word boundaries.

Salton went on to design a transcription system based on a dictionary and table look-up approach. It produced a "pseudo-English" output that was comprehensible but syntactically and semantically ambiguous. No attempt was made to reduce the ambiguity, instead the reader was presented with a list of all alternative correspondents, and was required to use his knowledge of syntax and context, etc., to select the correct alternative. Although the words appearing in the output of Salton's system were not always correctly spelt, he concluded by proposing a number of methods for improving the quality of the output text. These included grammatical designation of dictionary entries to resolve syntactic ambiguities, and a frequency analysis of Stenotype texts to be used in analysing semantic content. He also stated that the problem of determining word boundaries needed to be tackled.

Galli (1962), describes the Stenowriter, a system developed at I.B.M. for the automatic transcription of stenotype code. Its principal purpose was to demonstrate the feasibility of real time lexical processing of oral information into high quality English copy. Galli defined the main problems as those with which Salton concluded his report; "How does one resolve the one-to-many correspondence, i.e. the homographic problem, and how does one resolve the word boundary problem ?". After analysing conventional stenotype method, Galli discovered that by far the major sources of homographs were the actual stenotype rules. The occurrence of homonyms was less than (a) the occurrence of ambiguities resulting from the use of the same keys to represent more than one sound, and (b) the use of abbreviations to represent more than one word. It became apparent that many of the ambiguities were such that they could not be resolved by machine realizable syntactic or semantic analyses. In order to resolve the homographic problem therefore, Galli found it necessary to modify the rules of stenotype to overcome problems (a) and (b) above, and developed syntactical and contextual methods to resolve the homophonic ambiguities.

In conventional Stenotypy there is no provision for an end-of-word indication. Galli describes a dictionary search technique based on the principle of obtaining the longest lexical match between the various dictionary entries and the input string of Stenotype strokes. He states that :-

> this algorithm largely eliminates the necessity for an end-of-word indication, since the machine finds the longest set of Stenotype characters having an English equivalent, and this is always a word, a phrase, or a special dictionary item.

Certain types of word ending problems are not resolved by the algorithm. For example, word pairs or groups that, taken together, form a longer dictionary item would always be transcribed as the longer item. Thus, "in adequate quarters" would be transcribed as "inadequate quarters", and there are many more examples. Galli investigated the possibility of resolving the end-of-word problem completely by providing some means for end-of-word identification on the Stenotype input. To achieve this, a palm operated bar was incorporated into the Stenotype keyboard so that the operator could strike the bar on the final stroke (syllable) of each word. The character produced by the depression of the bar was included in the "matching" process performed by the dictionary look-up algorithm. It was found that the end-of-word bar reduced the recording speed by approximately 15%. Galli stated that the end-of-word bar could possibly be eliminated in the light of syntactic and contextual methods to aid in determining word boundaries. The details of a syntactical analysis by table look-up procedure can be found in Craft et al (1961). As far as contextual methods are concerned, one which was implemented was the listing of unique high frequency phrases to exploit local context. Since the longest match search mechanism always looks at longer entries first, if a phrase occurs, it would be found and transcribed correctly. For example, in the case of "some time" and "sometime", the phrases "some time ago", "some time was", "for some time", etc. could be included in the dictionary to help reduce the occurrences of ambiguities involving the word "sometime". This method was also used to reduce the probability of homograph occurrence.

One factor which affected the quality of English text produced by Galli's system was the occasional failure of the transcription algorithm to find a match for Stenotype patterns in the system dictionary. This was not due to any fault in the transcription program, but due either to keying errors made by the Stenotype operator, or an incomplete lexicon.

Galli handled such occurences by printing the "not found" strokes in an abbreviated phonetic form, a process termed "transliteration". In the transliteration process a quasi-phonetic translation of the stenoword occurs. For example, TKPW gives "g", pblg gives "j", etc. The result is to represent the word in a manner similar to phonetic English. Using this technique, some words were spelt correctly while others were not. For example, WAt/SOpb is transliterated "WATSON" and AOeuz/epb/HOur is transliterated "I'ZENHOUR". Galli concluded by describing certain editing procedures which were necessary in order to obtain "perfect" English copy, and certainly the sample output in his report is of very high quality. An extract from this sample is given at the end of this section.

Galli's work formed the basis of further research at I.B.M. by Newitt et al, 1967 and 1970. Here, an analysis of transcription errors divides them into three broad categories:

(i) those that can be corrected by programming or expanded processing capability;

(ii) those that require the Stenotypist to modify or alter his Stenowriting;

(iii) those which are not obviously correctable, e.g. word boundary problems.

Different ways of improving the transcription program are described in Newitt et al, 1967. The most obvious one is to update the system dictionary to include words which were not previously there. An interesting concept discussed is the construction of idiosyncratic dictionaries which could be referenced when the Stenotypist's representation of a steno form differs from the one in the system dictionary. There would be a unique idiosyncratic dictionary for each stenotypist, containing only the dictionary entries which need to be modified. Some of the conclusions from the work by Newitt et al, were as follows:

(i) Errors in Stenotype recording and subsequent computer transcription were such that with available editing techniques the production of final copy was not competitive with conventional typing for information entry.

(ii) Significant retraining would be required for most Stenotypists expecting to use machine transcription.

It was recommended in 1967 that the development of Stenotype as a marketable program or information entry service be discontinued at I.B.M. The process had too many errors, and correcting them offset the cost and speed advantage of Stenotype. Only significant improvements in automatic syntactic and semantic language analysis or major cost reductions in text editing would permit I.B.M. reconsidering automatic Stenotype transcription as a marketable product. An example of the output produced by the system of Newitt et al, is given at the end of this section.

In a different application in 1975, I.B.M. did do some more work involving Stenotype, and this was a speech translation machine for the deaf by Fox et al, 1975. The Stenotype machine was adapted to permit the elimination of spaces between phonetic characters forming a word. The output text which is very poor and does not even compare with the "transliteration" of Stenoforms mentioned earlier, was then displayed "in serial fashion on a line in justified position". The Palantype counterpart of this system has gone a great deal further in producing readable and comprehensible text (see for example Newell et al (1977), Downton et al (1979) and the brief description later in this report).

2.3.1 A summary of the most recent work in stenotype transcription

Downton (1981) found that in the U.S.A. machine shorthand computer aided transcription (CAT) systems based upon the American Stenograph machine were becoming more widely used. A significant amount of experience had been gained on the problems of introducing computer technology to court reporting.

In the field of television subtitling for the deaf via teletext, there is considerable activity in the U.S.A with 20 hours per week of programmes captioned for the deaf. The experience gained from this activity has thrown some light onto the psycho-linguistic problems of captioning.

2.3.2 Some examples of output from Stenotype Transcription Systems

Output from Galli's system

The sample output text below is taken from Galli (1962), and is indicative of the unedited, direct output from Galli's Stenotype to English transcription system. It was produced before any proper names had been included in the dictionary.

It would appear from the quality of this text that Galli was very fortunate to find a Stenotypist who made very few keying mistakes.

"There is a general assumption in Paris that the originl summit meeting, bringing East and West together for general discussions of world problems, will turn into a mountain range - summit after summit. President I'zenhour has been open in his approval of the idea, believing, according to Mr. Hagerti that 'it would be very difficult to solve all problems at one summit conference'. The British, for their part, have long been of the opinion that the period of international affairs now opening would be one of nego'shya'gs - promising no dramatic results from any one series of talks but holding the possibility of nibbling away at the major obstacles to a settlement of East-West differences. And, in any case, the British have considered that it would be better to talk things over with Mr. Krushef face to face than for each (side, sighed) to rail at the other across a gulf of physical distance and misunderstanding."

The text has been underlined where the transcription program has failed to determine the word intended by the operator.

'Output from Newitt et al's system'

The sample output text below is taken from Newitt et al. (1967), where according to Newitt, the errors underlined could be eliminated by retraining the Stenotypists.

"when you talk about bringing a witness 3 thousand miles that is a great burden any would lick to obviate it if we can, but may be we can do it by mate inning colonel after days/daze session so we can come bak in and accomplish sowing lick that."

The intended output now follows:

"When you talk about bringing a witness 3 thousand miles that is a great burden and I would like to obviate it if we can do it by meeting counsel after today's session so we can come back in and accomplish something like this."

2.4 THE AUTOMATIC TRANSCRIPTION OF PALANTYPE

Work on the transcription of Palantype machine shorthand (the British equivalent of Stenotype) began at the National Physical Laboratory in 1966. The original Palantype keying convention was modified to greatly reduce the degree of ambiguity present in Palantype records. The modifications were kept to a minimum in view of the problem of retraining existing Palantype operators. The system designed at N.P.L. incorporated a dictionary of over 75,000 English words and their corresponding Palantype equivalents. The dictionary look-up algorithm attempted to identify in each instance the longest set of Palantype chords which, taken together in sequence, formed an acceptable English word. Any not found chords were transliterated before being output. As with Stenotype, Palantype also suffers shortcomings with the "longest match" approach. Sometimes chords can be incorrectly joined together. This situation arises when two chords which when considered separately produce two English words, yet when considered together (as with the longest match) produce only one. Consider the following examples:

Chords	Considered Separately	Considered Together
/EU N/IT	you nit	unit
/A N/EU	a new	anew
S/UM T/AIM	some time	sometime
/A P+/AC	a back	aback

This problem can occur in another form; the text words "a taxi driver" are rendered as "attack SI driver", "a little more" as "alley TL more". Price, 1971, suggested a way of overcoming the incorrect joining together of chords in this latter form, but he did not implement it. However, it was implemented in a later system developed at Leicester Polytechnic (Booth and Barnden, 1979) and based on Price's work.

The possessive form of irregular plurals and the plurals of proper nouns were not included in the system dictionary at N.P.L. which meant that when the Palantype operator pressed the chords for such words, the transcription algorithm was unable to find a match for them. This problem could have been overcome by updating the system dictionary to include irregular and proper noun plurals, but as there are such a lot of them the size of the dictionary would have been increased significantly. As an alternative, Price designed a special program feature which dealt very well with the problem. There were relatively few cases where the feature could prove detrimental to the transcription. Attempts at resolving other shortcomings of the N.P.L. system are described in various papers by Szanser, see bibliography.

As the success of a transcription system is dependent to a very large degree on the quality of the input data, Price decided to analyse the standard of Palantype operator performance. The first important discovery was the keyboard itself placed the operator at a disadvantage; its ergonomic design left a lot to be desired. For example, as mentioned earlier the weakest fingers (commonly the third and fourth fingers of the left hand) have the most work to do. A course was commissioned at N.P.L. and devised by the Palantype Organisation to convert six experienced Palantype operators from the old convention to the convention necessary for input to a computer transcription system. Before and after the course the operators' error rate was measured and it was found that it had increased significantly. A proposal was made to redesign the Palantype keyboard, and an experimental keyboard was designed and constructed in which many keyboard parameters could be adjusted. It was also proposed to experiment with these parameters and optimise them in practical use with several Palantype operators, but this did not prove practicable.

At the inception of the N.P.L. project the potential applications of the system were as follows:-

(i) transcription of House of Commons proceedings

(ii) transcription of law court proceedings

(iii) transcription of the proceedings of committees, etc.

(iv) very rapid typewriting extended to embrace computer typesetting

A description of the existing system and the applicability of a computer system for each of the above can be found in Price (1970). The most promising application seemed to be computer typesetting. Some conclusions of the N.P.L. project were that operator performance prevented high quality output, the keyboard needed to be redesigned and even then a degree of automatic error correction and editing would be necessary.

A problem encountered by the N.P.L. project was the occurrence of phonetic ambiguities. As Palantype is a phonetic shorthand, words which sound the same are Palantyped in the same way. Of the 75000+ English words in the N.P.L. system dictionary, there are more than 1000 groups of words which fall into this category. An analysis of these groups, Szanser(1970), showed that they could be resolved in different ways. Some of them could be prevented by adjusting the recording convention, others, whenever the interpretations differed significantly in relative frequencies in ordinary language, could be eliminated simply by removing the rare word from the dictionary. A substantial proportion appeared to be solvable by syntactic and contextual analysis. Of the 1000 groups of words, 100 involved a proper noun, e.g. CARR/KERR. In order to perform syntax analysis on the output text, Szanser (1970) corrected any keying mistakes so that the text consisted of whole words and was free from transliterations. Szanser was working within severe limitations; there was no possibility of marking all the entries with grammatical codes, and the analysis procedures had to be brief and fast so as not to reduce the transcription speed. Szanser describes a very interesting approach to "local" syntax analysis using the grammatical classifications of Thorne et al (1968). In the light of the limitations within which he was working, his procedures proved very successful; 65% correct resolutions was achieved. Another method adopted for resolving ambiguities, and one which was described earlier, was the inclusion in the dictionary of high frequency phrases.

The major factor affecting the quality of text produced by Palantype transcription systems is the large number of keying mistakes made by

Palantype operators. An automatic error correction technique designed by Szanser (1970) was named "elastic matching". Although this technique was specifically designed for automatic error correction in Natural Language, it was easily adapted to Palantype transcription. Elastic matching is performed between a word which contains an error, and a list of words in which the intended word is present. All words belonging to the list are "linearized", that is, converted into segments called lines, in which the letters are arranged in an agreed order. This is why Palantype transcription was easily adapted to the elastic matching technique, the Palantype chords are already linearized and each letter is associated with a fixed position. Words in the list which differ from the erroneous word by more than one letter - a letter missing, a wrong letter or two letters interchanged - are rejected. If there is more than one word in the list which differs from the erroneous word by only one letter then Szanser attempts to resolve them in two ways. The first is the "general content check". As the text is processed, each different word satisfying certain conditions is stored, so that when an erroneous word is encountered it can be matched against this list first. The idea behind this was that words tend to be repeated by a speaker. The second approach was to apply the "local" syntax analysis procedures mentioned earlier. These two methods were found to be largely complementary. Examples of the output produced by the N.P.L. system are given at the end of this section.

Another application for the automatic transcription of Palantype is as an aid for the deaf. Newell and King (1977) of Southampton University on considering a transcription system for the hearing-impaired decided that the goal of producing perfect orthographic output, was not as important in this application as other applications since deaf people would still be able to elicit the necessary information even if the text was not perfect. Consequently, a different approach to the one of dictionary and table look-up was adopted. The basis of their system was a series of transliteration algorithms which accepted Palantype chords as input and produced a kind of phonetic English output where some words were spelt correctly and others were not. As no large dictionaries were involved, the system could be kept very small and was in fact installed in an attache case for the use of a deaf M.P., Jack Ashley, so that he could follow debates in the chamber of the House of Commons. A version of this portable system has also proved very successful in a business environment, Hayward (1979). The output produced by Southampton University's prototype system was not very easy to read, and since then a great deal of work has gone into improving the translation algorithms and the visual presentation of the output text, Newell and Downton (1977). In order to improve the output text still further, Downton and Newell, (1979) have combined the translation algorithm and dictionary look-up approaches by simulating several possible systems using very small and very large dictionaries.

2.4.1 A summary of the most recent work in Palantype transcription

Newell and Downton at Southampton University have been involved in subtitling television programmes to benefit hearing impaired viewers. A method of subtitling television using the teletext systems is described in Newell & Hutt (1979). The Leicester Polytechnic system has also been used for this application, Thomas & Hawkins (1980), and on one occasion was used to produce output on a standard TV programme.

In the field of error detection and correction, some interesting methods have been proposed by Passingham (1980) with an emphasis on error prevention. There is currently a great deal of activity in the field of Palantype transcription.

2.4.2 Some examples of output from Palantype Transcription Systems

Output from N.P.L's system

The sample output text below is taken from Price(1971) and is a direct transcription of Palantype code into English text. At this stage, no error-correction procedures were incorporated.

"Which I consider to be one of the most damnable, mean, con TEM PTI BL, indefensible clauses that has EFR been placed in a bill in the House of Commons. When I say this I say it deliberately, I say it is a BA STRD child of panic buy'by cower Diss AND the GodMao ? Godmother is I GNO RNS. I'm sorry to say these harsh words, but I think that the house is not fully seized of the importance of clause'claws three, both ass it affects the school children AND ass it affects the industry which is going to be "E ? severely hit ass a result of this panic measure."

The occurrence of the ? character indicates that the Palantype operator has depressed the chord ULFTS, which means that she is aware that she has just made a mistake.

The following sample output text is taken from Szanser (1971) and is the result of an experiment by Szanser to determine the effect of syntax analysis on resolving phonetic ambiguities present in the output text of the N.P.L. system. Szanser corrected any keying mistakes in the text manually before the experiment. This ensured that the text being subjected to syntax analysis would consist wholly of English words.

"Sew /so /sow there are two things that I think are important, the data base of the patient, and the idea that we must get some /sum order and system into the various things that we try to eliminate some /sum of the errors, and these are the sort of main objectives that ones has approached the system with. The first one / won I talked about was improved collection of data, and I think more important is its /it's retrieval, and the other thing is the power of analysis, with all these odd kinds of paper around, tucked into them is the most unique experience in the world so far as the medicine is concerned, its/ it's totally unanalysable, you cant/ can't do anything except produce the notes for the individual."

A word underlined is one that has been selected by the analysis procedures as the correct alternative. If more than one is underlined, it means that the ambiguity was not resolved.

'Output from Southampton University's system'

The sample output text below is taken from Newell and Downton (1977) and was produced by one of Southampton University's earlier transcription units.

"THIS DEM STRAE SHN SHOES TH PROE TOE TIPE ADE FOR TH DEF WHICH HAS BEEN DE SINGDE AN BILT AT SOU THMAMP TUN UNE VER STI. A NUE SIS TEM FOR..."

From this phonetic English it can be discerned that the orthographic equivalent is as follows:

"This demonstration shows the prototype for the deaf which has been designed and built at Southampton University. A new system for..."

The following sample output text was produced by a more up to date transcription unit, and is an extract from the unedited transcript of "Subtitling Television for the Deaf using the Teletext System", presented at the National Deaf Children's Society's exhibition and conference at Mullard House in May 1978.

"WE HAVE SEEN THAT TELEVISION CAN BE SUP TITLED FOR THE DEF , BUT HAVING THE TEC NOLOGY IS ONLY THE BE INING OF THE PROBLEM , ITS NOT THE END OF IT WHAT I WANT TO DO REALLY IS SAY VERY MUCH THE SEM SORT OF THINGS THAT BILL WAS SAING BUT A PLIDE SPE SFITALLY TO SUP TITLING OVER THE TE LE TECST LINC. . FUR STLI , SUP TITLING VYA TE LE TECST IS VERY DIFFERENT FROM PRO TUSING A PROGRAMME FOR OR AIMD AT THE DEF , BECAUSE ????? FOR TOO REA SONTS ."

The final sample from Southampton's system is taken from Downton and Newall, 1979, and is the result of a simulation of the Southampton system incorporating a 1200 word dictionary.

"I SHA NOT CO MENT ON THE SPEECH OF THE HONOURABLE MEMBER FOR RUT LAND , ALTHOUGH A NUMBER OF THINGS HE SAID CALL FOR CO MENT . I WEESH TO SPEAK ABOUT THE NORTH EAST AND TO CON FINE MYSELF TO THAT BECAUSE TIME IS GETTING ON . I WISH TO DO TWO THINGS . FIRST TO EXPLAIN FIGHT CLEAR LI AT THE RIS K OF BO RING THE HAS WHAT THE PO SI SHN IN THE NORTH EAST OF ENGLAND IS NOW....".

Chapter 3

RESOLVING WORD BOUNDARIES

3.1 THE WORD BOUNDARY PROBLEM

Transcribing shorthand code into English text is made more difficult by the fact that there is no explicit "end of word" marker to guide the transcription process to reconstitute word boundaries, i.e. there is nothing in a string of chords to indicate where one word ends and the next one begins. It was mentioned in the previous chapter that Galli (1962) investigated the possibility of resolving the word boundary problem completely by incorporating a palm operated bar into a Stenotype keyboard, so that the operator could strike it on the final syllable of each word. This resulted in the recording speed being reduced by 15% which may be one reason why there is no record of a similar experiment having been tried on a Palantype keyboard.

There have been several different algorithms developed in an attempt to reconstitute word boundaries from strings of Palantype chords containing no 'end of word' indicators. These are described in detail later, but in general a one-to-one approach, which attempts to match one string of chords against one word in an English dictionary where the Palantype correspondents are listed, is out of the question. This is because some chords can realise the final syllable of certain words and the initial or medial syllables of other words. For example, the chord " T+/I " in figure 3A.

English word	Palantype equivalent
Candy	C/AN T+/I
Distract	T+/I STR/ACT
Expenditure	E/CS P/EN T+/I TE/UR

Fig. 3A Examples of the use of the chord T+/I

A transcription algorithm employing a one-to-one approach, on encountering " T+/I " would not know whether it was at the start, in the middle or at the end of a word. Therefore,a system which references a dictionary of English words with their corresponding Palantype chords must somehow structure this data so that when a chord is received, a transcription program 'knows' whether that chord with respect to the series of chords already received is a constituent of the chord requirement to make up an English word. A transcription program needs to know whether a chord can start a word, constitute a word in its own right or whether it is in the middle or at the end of a string of chords which constitute an English word. Once the data is structured in a way which permits a transcription program to elicit this information, the transcription program itself must adopt a strategy which uses the information to reconstitute word boundaries. Before examining these strategies, the data structures (adopted at Leicester) which provide the necessary information shall be described.

3.2 DATA STRUCTURES

Consider a very small subset of words from the English dictionary, and their associated Palantype chords (Table 3A):

Subset of English Dictionary		Corresponding Palantype chords
1	a	/A
2	aback	/A P+AC
3	abbacy	/A P+/A S/I
4	abbe	/A P+/E
5	agree	/A C+R/I.
6	are	/AR
7	baby	P+/E. P+/I
8	bidding	P+/I T+/IN
9	can	C/AN
10	candy	C/AN T+/I
11	din	T+/IN
12	distract	T+/I STR/ACT
13	father	F/A TH/R
14	fingers	F/IN C+/RS
15	girl	C+/U.L
16	greedy	C+R/I. T+/I
17	her	H/UR
18	nibbles	N/I P+L/S
19	she	SH/I.
20	sheeny	SH/I. N/I
21	when	HFE/N

Table 3A

Subset of English words and corresponding Palantype chords

Note that in Table 3A the symbols to the left of the slash denote those keys operated by the left hand, and the symbols to the right of the slash denote those keys operated by the right hand.

The data structures adopted by the Leicester Polytechnic system were much the same as those used by Price (1971), except that there was no need to create supplementary indexes in order to avoid spending too much time in disc reading. Disc access times are faster than they were when Price's project was in progress, and thus do not cause problems. This means that our data structures are probably simpler because, for any chord at any level of any tree, we need only to store a pointer to the location on disc of the list of chords at the next level, and a record of the number of chords in that list. A brief description follows of how we constructed the trees.

We grouped together all those chords which can start a word, and called this list "initial chords", Table 3B shows the initial chords for our subset of the English dictionary.

Initial Chords
/A
/AR
P+/E.
P+/I
C/AN
T+/IN
T+/I
F/A
F/IN
C+/U.L
C+R/I.
H/UR
N/I
SH/I.
HFE/N

Table 3B

Initial chords for subset of English dictionary

For the 75,000 word dictionary, there are over 10,000 initial chords. Although an initial chord may start more than one word, it only appears once in the list. If an initial chord constitutes a word in its own right, as with /A, /AR, C/AN, T+/IN, C+/U.L, H/UR, SH/I, and HFE/N in Table 3B then it is stored with a pointer to the position of the word in the dictionary. Table 3C shows the initial chord list with pointers to English words.

Absence of a pointer to an English word in Table 3C indicates that the chord does not constitute an English word in its own right.

Initial Chords	Pointer to the position of the word in the English Dictionary
/A	1
/AR	6
P+/E.	
P+/I	
C/AN	9
T+/IN	11
T+/I	
F/I	
F/A	
F/IN	
C+/U.L	15
C+R/I.	
H/UR	17
N/I	
SH/I	19
HFE/N	21

<u>Table 3C</u>

<u>Initial chords and pointers to English words</u>

Taking the first initial chord /A, we now group together all those chords which can immediately succeed it in the chord requirement for an English word. This list is called "second level successors to /A". If the initial chord followed by a second level successor together constitute an English word, then a pointer to the position of the word(s) in the English dictionary is stored with the second level successor. Table 3D shows the second level successors to /A. In Table

3D we see that /A followed by P+/AC points to the second word in the English dictionary, "aback". Absence of a pointer in Table 3D indicates that the initial chord followed by the second level successor do not fulfill the chord requirement for any English word.

Second level successors to /A	Position on word in English Dictionary
P+/AC	2
P+/A	
P+/E.	4
C+R/I.	5

Table 3D

Second level successors to /A

Taking each second level successor in turn we create third level successor lists and so on, remembering to store appropriate pointers to the English dictionary, until we have completed a hierarchical structure of lists (tree structure) to "N" levels for the first initial chord. This process is then repeated for the remainder of the chords in the initial chord list, resulting in the whole of the Palantype dictionary being represented in tree structures.

3.3 APPROACHES TO RESOLVING THE WORD BOUNDARY PROBLEM

There have been several different approaches to resolving the word boundary problem. Although it could be argued that one is only an extension of another, they are reported separately here because as they were developed in different institutions and/or at different times, some of the differences between them are significant.

The first approach, known as 'the longest match' was the technique used by Galli(1962) and Price(1971), and its explanation should help the reader gain a deeper insight into the difficulties associated with shorthand code to English text transcription. The second approach, 'two chord look-ahead', was developed in an adhoc fashion by Booth & Barnden (1979), in an attempt to improve the output text by recognizing the shortcomings of the longest match and trying to eliminate them. The third approach, 'n word look-ahead', was developed as part of the author's research and was inspired by the 'look-ahead' concept of 'two chord look-ahead'. This concept appeared to be the key to producing output consisting wholly of English words from error-free input.

This last point is a very important one since the word boundary problem can be viewed as two separate problems, and indeed was viewed as such for the major part of the present research. The first problem is how to produce output consisting wholly of English words from error-free input. The second problem arises because even when the output consists wholly of English words, the words have not necessarily been partitioned at the correct word boundaries. For example, the chords " /A N/EU S/I T/IN ", can be transcribed equally well in three different ways and still produce English words.

a new sitting

a newsy tin

anew sitting

The fourth algorithm, WORDFINDER, was designed by viewing the above two problems as one, and provides a partial solution to the word boundary problem by not only producing output consisting wholly of English words from error-free input, but also producing all alternative transcriptions for any such input.

3.3.1 The longest match

A transcription program incorporating the longest match approach looks at the first chord in a string of chords which are to be transcribed. If the string is error-free (we shall subsequently discuss non error-free input), i.e. the operator has not made any mistakes, then the first chord will be an initial chord (a chord which can start a word). The program then checks to see whether the next chord in the input string is in the list of second level successors to the initial chord, and if so, whether the chord after that in the input string is a third level successor and so on, until a chord is scanned which is not in the list of (i+1)th level successors to the (i)th level chord,(or until all the chords in the input string have been scanned). If the chord in the (i)th level list has a pointer to an English word, then the English word is output and the chord which failed the test at the (i+1)th level list, hereafter referred to as FAIL chord, is tested to see whether it is an initial chord. If it is, then:

(a) the process of matching the next chords in the input string against the successor lists of this new initial chord is undertaken. If FAIL chord is not an initial chord then the program assumes (incorrectly) that the operator has made a mistake, and so:

(b) outputs the chord in its Palantype format reflecting the keys pressed, and

(c) scans the next chords in the input string until it finds an initial chord, repeating (b) for all the chords it encounters before it finds an initial chord.

Consider the following examples:-

(i) The chords /A P+/E. P+/I C+/U.L should ideally transcribe into "a baby girl". P+/E. is the (i)th level chord and P+/I is the FAIL chord. P+/I is also an initial chord, but C+/U.L is not a second level successor to it so the longest match produces "abbe P+/I girl".

(ii) The chords C/AN T+/I STR/ACT H/UR should ideally transcribe into "can distract her". T+/I is the (i)th level chord and STR/ACT is the FAIL chord. STR/ACT is not an initial chord and so the longest match produces "candy STR/ACT her".

If the chord in the (i)th level list has no pointer to an English word, then the program looks for the chord which is at the deepest level above i in the hierarchical structure and which has a pointer to an English word. The English word pointed at by the chord at this (i-j)th level is output and the chord which immediately succeeds it in the input string is tested to see if it is an initial chord. If it is then (a) above is performed, otherwise (b) and (c) above. If the program is unable to find a chord with a pointer to an English word, then all the chords up to and including FAIL chord are output in Palantype format. The above examples outline the major shortcoming of the longest match approach, which is that when FAIL chord is encountered, the program outputs the English word pointed at by the chord which is at the deepest level (down as far as i) in the hierarchical structured lists.

3.3.2 Two chord look-ahead

'Two chord look-ahead' was developed to improve the output text, by recognizing shortcomings of the longest match approach and eliminating them. An approach along these lines was suggested by Price (1971), but he did not implement it.

The way in which our 'Two chord look-ahead' overcomes the major shortcoming of the longest match approach is as follows. When the longest match approach encountered the FAIL chord, it output the English word pointed at by the chord deepest (as far as i) in the hierarchical structured lists. When a transcription program incorporating 'Two chord look-ahead' encounters the FAIL chord, it outputs the English word pointed at by the chord which is at the deepest level (as far as i) AND which is immediately succeeded in the input string by a "valid successor". The properties of a "valid successor" are as follows:-

1. it must be an initial chord, and

2. it must in turn be succeeded in the input string by a second level successor, or

3. it must have a pointer to an English word and be succeeded in the input string by another initial chord

What this means is that when FAIL chord is encountered, we may have found the end of a word, so we must check that the next chords in the input string start to build the next word. Taking the two examples we used earlier, we have firstly the chords: /A P+/E. P+/I C+/U.L which should ideally transcribe into "a baby girl". 'Two chord look-ahead' on scanning P+/I finds that it is not in the list of third level successors to /A and P+/E. , and is therefore the FAIL chord. It next looks for the chord at the deepest level in the structured lists, which has a pointer to an English word AND which is immediately succeeded by a "valid successor". It finds that P+/E. has a pointer to the word "abbe", and so checks to see if P+/I is a valid successor. Although P+/I is an initial chord and thereby satisfies property (a) of a "valid successor", it fails to satisfy property (b) because C+/U.L is not in the list of second level successors to P+/I . It also fails on property (c) because it has no pointer to an English word. Consequently the program looks higher up the hierarchical structure for another chord which has a pointer to an English word. /A points to the English word "a", and P+/E. satisfies properties (a) and (b) of a valid successor. Therefore 'two chord look-ahead' produces the desired output; "a baby girl".

In the second example, the chords C/AN T+/I STR/ACT H/UR, should ideally transcribe into "can distract her". 'Two chord look-ahead' on scanning STR/ACT finds that it is the FAIL chord. Although T+/I has a pointer to the English word "candy", STR/ACT is not a "valid successor" because it fails to satisfy property (a). C/AN points to the English word "can" and T+/I is the "valid successor" giving the desired output; "can distract her".

Although 'two chord look-ahead' produces higher quality text than the longest match approach, it has in itself a shortcoming. Consider the chords " /A N/EU M/IS M/A T/IST /IS /A ", which should ideally transcribe into "a numismatist is a ". 'Two chord look-ahead' finds that /A is an initial chord to which N/EU is a second level successor with a pointer to the English word "anew". M/IS is not a third level successor to /A and N/EU , and is therefore the FAIL chord. As N/EU has a pointer to the English word "anew", 'two chord look-ahead' now tests to see if M/IS satisfies the requirements of a valid successor. M/IS is an initial chord, thereby satisfying property (a), and it is succeeded by a second level successor, M/A , thereby satisfying property (b). As M/IS has been accepted as a valid successor, 'two chord look-ahead' outputs the English word "anew" and so produces " anew MIS MA TIST is a " instead of the desired output.

3.3.3 N word look-ahead

N word look-ahead was inspired by the 'look-ahead' concept as demonstrated by 'Two chord look-ahead'. The reason for 'N' instead of a definite number can be explained as follows. 'Two chord look ahead' failed to produce output consisting wholly of English words because there existed at least one word in the dictionary with the following characteristics.

1. Its starting syllable could also be the final syllable of a different word

2. It contained the start and second syllable of a different word

3. It contained a chord which was not a word in its own right and was not part of a different word within the original word or of one that followed

As we saw in the previous section with the word "numismatist", N/EU could also be the end of the word "anew", M/IS M/A could start the word "mismanaged" and T/IST was neither a word in its own right nor part of any other word within "numismatist" or of the word that followed.

Remembering the conditions for a "valid successor" in 'Two chord look ahead', the "numismatist" example would have produced English words if condition (b) had been modified to "it must in turn be succeeded by one English word", i.e. 'One word look-ahead'. Now the question can be asked "What would cause 'one word look-ahead' to fail to produce output consisting wholly of English words ?". The answer is a word in the dictionary with the following characteristics.

1. Its starting syllable can also be the final syllable of a different word

2. It contains another word

3. It contains a chord which is not a word in its own right and is not part of a different word either within the original word or of one that follows

Could the dictionary be analysed to see whether such a word exists ? Taking each word in turn, it is possible to write a program to determine whether its starting syllable can also function as the final syllable of another word (characteristic 1 above). It is also possible to determine whether each word contains an 'inner' word (characteristic 2). However, it is not immediately clear, when one imagines the possible combinations of words, how to determine whether the chord which is not a word in its own right and is not part of an 'inner' word, is part of a following word. Even if a program could be written to do this, and it was found that no such word existed, 'One word look-ahead' would only be valid for the dictionary which had been analysed. Should the dictionary be expanded or updated the new words would have to be analysed for the above characteristics.

As it happens, an example has been contrived containing a word which has the above characteristics, to show that looking ahead one word is not enough. Consider the following words,

English words	Palantype chords
concentricities	C/ON S/EN TR/I ST/IS
ruby	R/U P+/I
Rubicon	R/U P+/I C/ON
sentry	S/EN TR/I

The string R/U P+/I C/ON S/EN TR/I ST/IS should produce "ruby concentricities", but 'One word look-ahead' produces "Rubicon sentry ST/IS". It's easy to see that 'Two word look-ahead" (note word, not chord) would prevent the above problem, but what if there exists a word whose second characteristic is that it contains two 'inner' words ?.

Clearly the argument can continue for three, four, five word look-ahead etc.

To be absolutely sure of producing output consisting wholly of English words from the present dictionary, then theoretically the algorithm should be 'Ten word look-ahead'. The reason for this is that the longest word in the dictionary is represented in Palantype by 9 chords. Even if each of these chords constituted a word in its own right, on receiving the tenth word one could be sure that the end of the first word was a boundary that would not prevent the algorithm from producing output consisting wholly of English words.

From a pragmatic standpoint, it would not have been worthwhile to implement 'Ten word look-ahead' because of the extra 'book keeping' involved in keeping track of the large number of chords and pointers that were being used to construct English words. Also, if an eventual goal was 'real time' transcription, 'Ten word look-ahead' has serious and obvious implications. To demonstrate the concept of looking ahead in words instead of chords, 'One word look-ahead' was implemented with the hope that 'ruby concentricities' would not occur too frequently.

3.3.3.1 One word look-ahead

The strategy behind 'One word look-ahead' was to output an English word only when the English word which comes after it has been 'built'. To be consistent with the 'longest match' and 'Two chord look-ahead' descriptions, this can be expressed in terms of FAIL chord. When a transcription program incorporating 'One word look-ahead' encounters a FAIL chord, it tests to see whether it already has an English word which has itself been succeeded by a FAIL chord. If it has not, then it stores the English word pointed at by the chord deepest (as far as i) in the tree structure, and starts to build the next word. If it already has a stored word, then it is only output if the chords processed between the end of the stored word and FAIL chord constitute an English word. Otherwise, it finds the next word and then outputs the stored word before storing the new word. To see this in practice, the "baby girl" example shall be used.

/A is an initial chord to which P+/E. is a second level successor. P+/I is a FAIL chord and, as there is no stored word as yet, the word "abbe" is stored. P+/I is an initial chord but C+/U.L is not a second level successor to P+/I and so is a FAIL chord. P+/I does not constitute a word in its own right, so the program looks higher up the tree structure for a chord with a pointer to an English word which is succeeded by an initial chord. /A has a pointer to the English word "a" which is stored in place of "abbe", and P+/E. is an initial chord to which P+/I is a successor. C+/U.L is a FAIL chord again, but this time the chords between the English word "a" and the FAIL chord do form an English word (baby) and therefore "a" is output and "baby" stored. C+/U.L is an initial chord but C/AN is not a second level successor and is therefore the FAIL chord. Again, an English word is already stored, and the chord between it and FAIL chord does constitute an English word. "baby" is output, "girl" stored and so it proceeds.

Let us now consider the "numismatist" example on which 'Two chord look-ahead' failed. /A is an initial chord to which N/EU is a second level successor. M/IS is a FAIL chord and as there is no word stored as yet, "anew" is stored. M/IS is an initial chord to which M/A is a second level successor. T/IST is a FAIL chord, but M/IS and M/A do not together form an English word, so the program looks higher up the tree structure for a chord with a pointer to an English word and which is succeeded by an initial chord. It finds /A and so "a" is stored in place of "anew". N/EU is an initial chord to which M/IS is a second level successor. M/A is a third level successor to N/EU and M/IS, and T/IST is a fourth level successor to N/EU, M/IS and M/A . /IS is a FAIL chord but since the chord between the English word "a" and the FAIL chord do form an English word (numismatist), the word "a" is output and "numismatist" stored, and so on.

3.4 WORDFINDER

The algorithms described so far have all been based on producing the longest lexical match between the various dictionary entries and the input string of chords. They were also designed with the goals of

1. producing output consisting wholly of English words, and

2. partitioning those words at the correct boundaries

being viewed as two separate problems.

WORDFINDER was designed by considering the two mentioned problems as one. In order to resolve the incorrect joining of Palantype chords it is necessary to

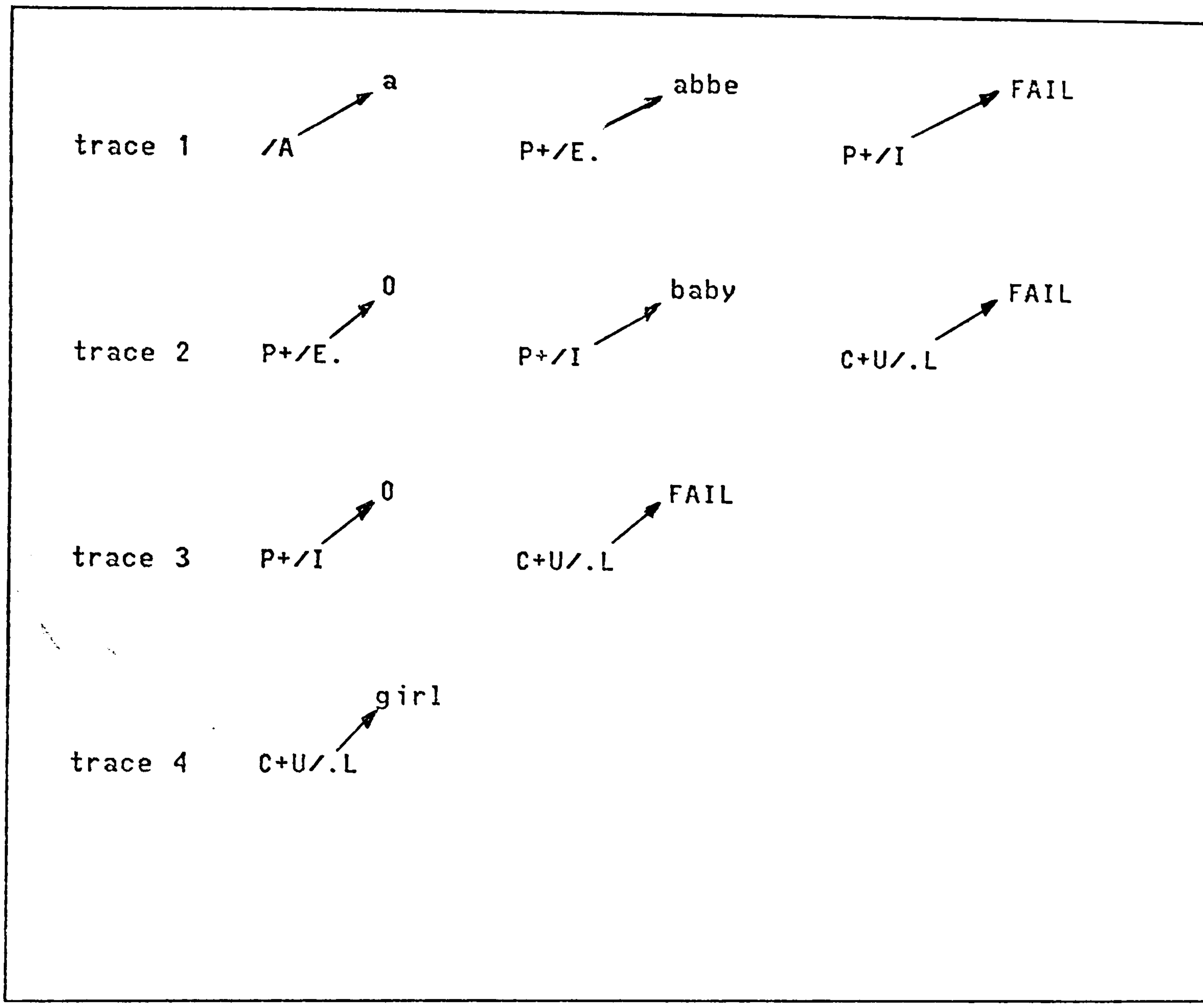
1. recognise during the transcription stage the places at which a sequence of chords can be transcribed in several different ways, and

2. to produce a list of alternative transcriptions so that they may be submitted for subsequent processing (e.g. syntax analysis) in order that the correct one be selected.

Algorithms based on the 'longest lexical match' approach are not able to satisfy point 1 above since they do not search the trees of every initial chord they encounter, but only the trees of the initial chords following a longest match. For example with the chords /A N/EU S/I T/IN , longest match type algorithms can generate "anew sitting" and "a new sitting" but not "a newsy tin". The reason for this is that they do not search the tree of the chord N/EU .

WORDFINDER is based on searching the tree of every initial chord encountered in the input string. To demonstrate how WORDFINDER functions, the examples used by the previous algorithms will be considered.

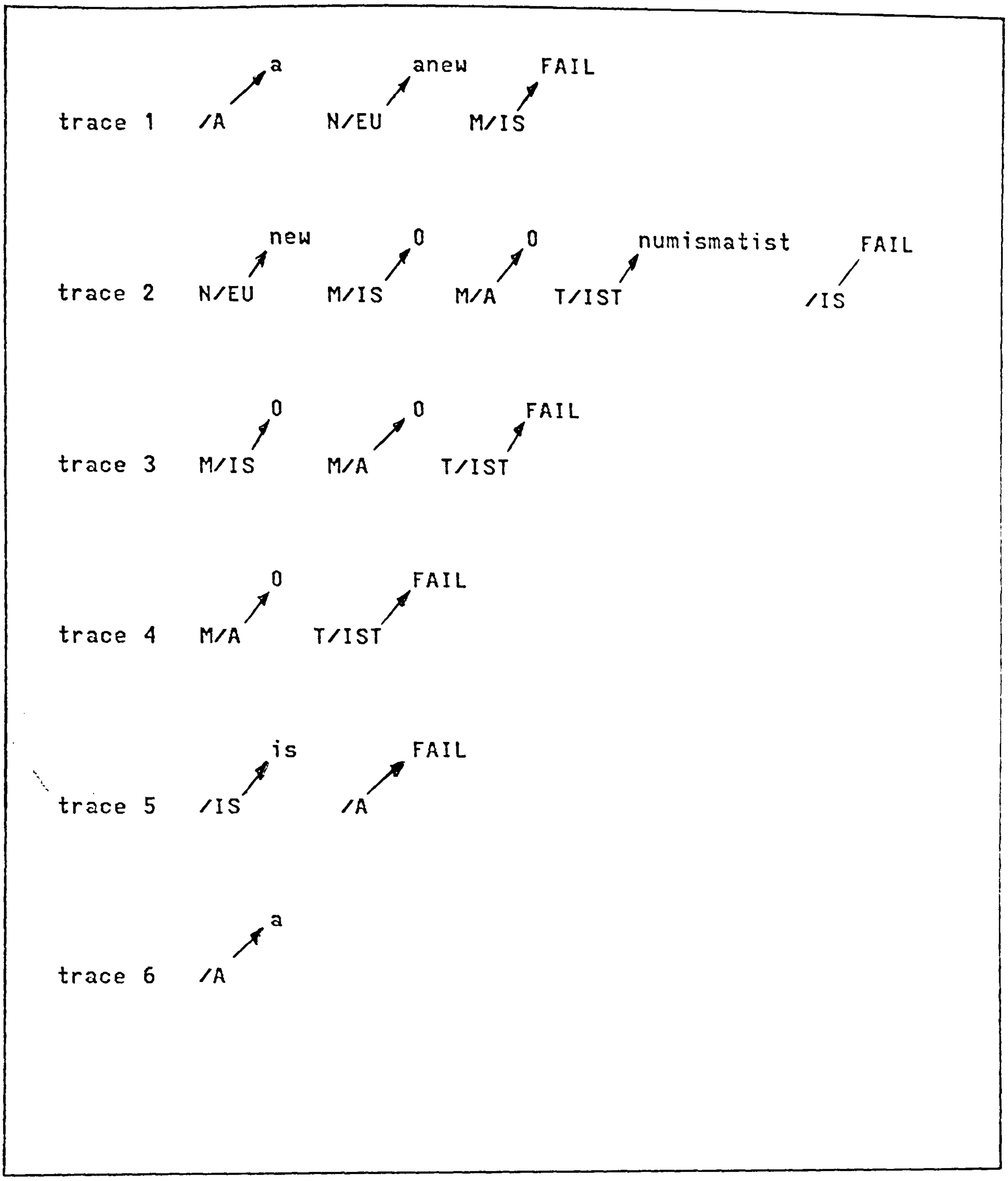
The chords /A P+/E. P+/I C+/U.L should ideally transcribe into " a baby girl". WORDFINDER encounters the initial chord /A and so starts a 'trace' for it. The next chord, P+/E. , is a second level successor to /A but is also an initial chord itself and so WORDFINDER starts a separate trace for P+/E. . The third chord P+/I is the FAIL chord in the tree for /A , but is a second level successor to P+/E. . It is also an initial chord and so a trace is started for it. The next chord, C+/U.L , is the FAIL chord in the tree of P+/E. and also P+/I . However, it is an initial chord itself and so a trace is started for it. C+/U.L is the final chord in the input string and WORDFINDER has produced the following traces:-



Being at the end of the input string represents the same condition to WORDFINDER as encountering a period or comma in that at this stage it scans the traces to determine the output string. If the input is error free then WORDFINDER can produce output consisting wholly of English words since it has constructed all the English words possible from the given input string.

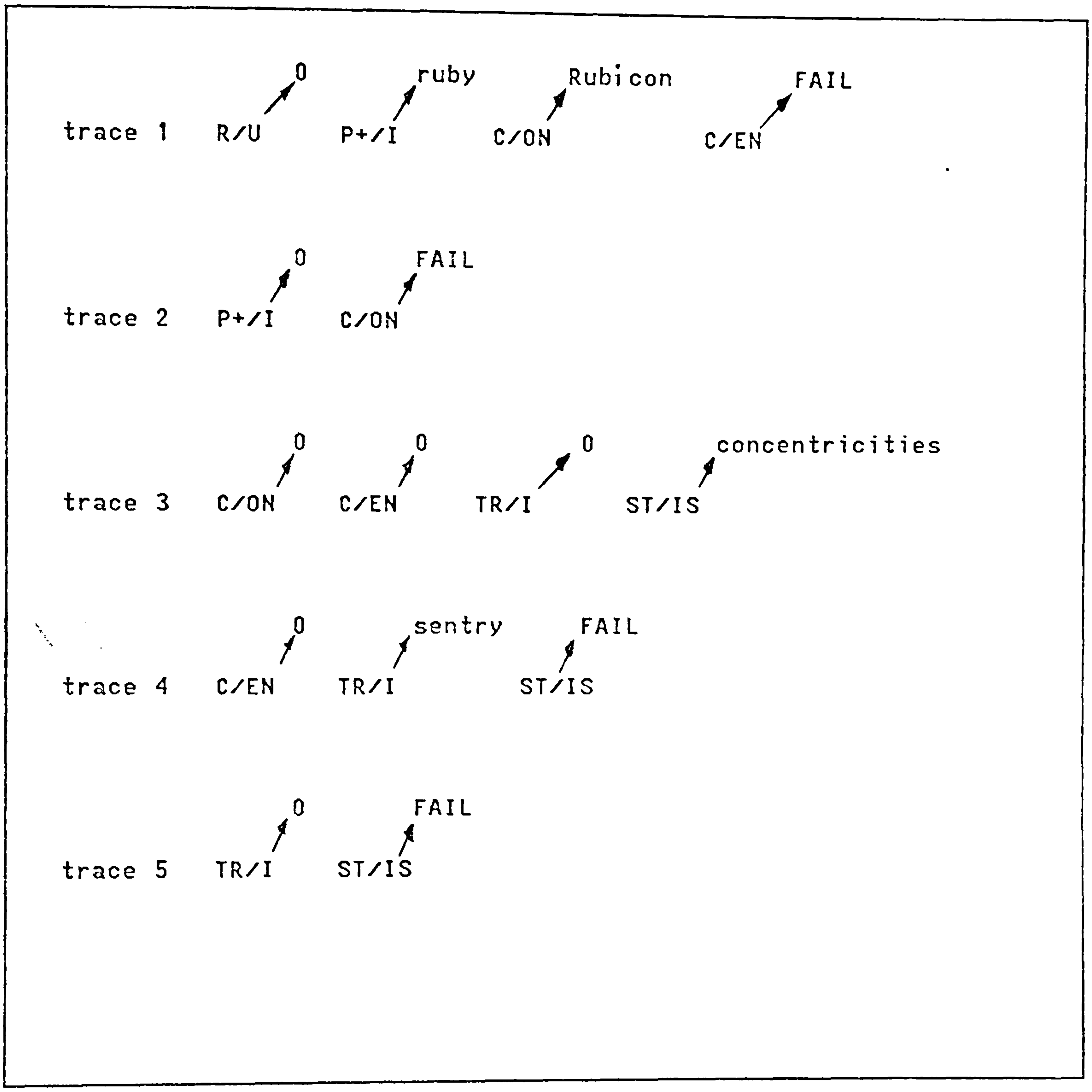
WORDFINDER takes the longest word in trace 1 and skips all other traces which start with a chord that is included in the first word. In this case it takes "abbe" and skips trace 2. As there is no word in trace 3, WORDFINDER realises that it has made the wrong choice and so takes the next longest word in trace 1 and produces "a baby girl".

Similarly for the chords /A N/EU M/IS M/A T/IST /IS /A , which should ideally transcribe into " a numismatist is a ", WORDFINDER produces the following traces:-



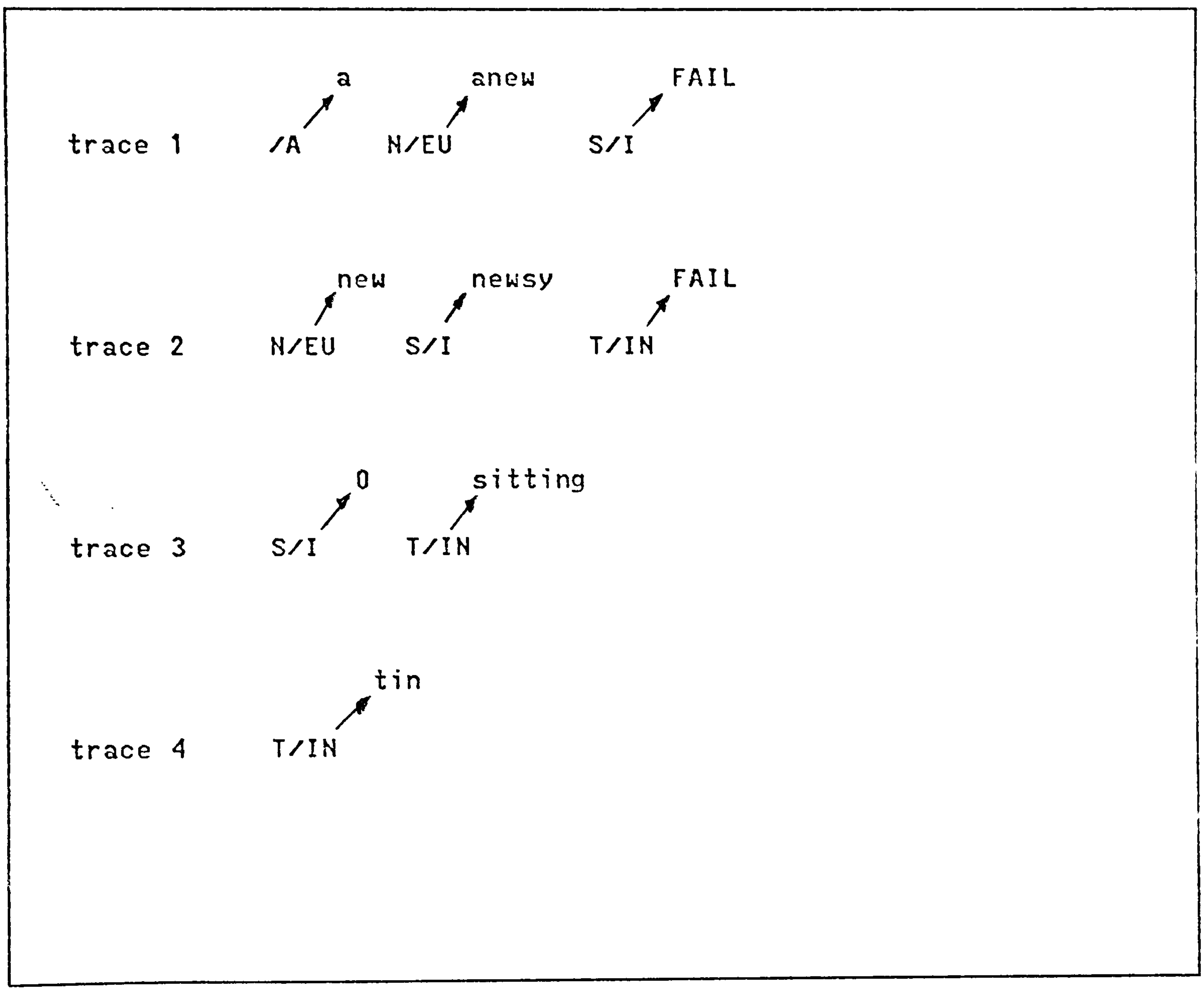
On taking "anew" and skipping trace 2, WORDFINDER finds that there is no English word in trace 3 and therefore realises it has made a wrong choice. The next choice of "a" produces the desired output.

The chords R/U P+/I C/ON C/EN TR/I ST/IS should ideally transcribe into "ruby concentricities" and WORDFINDER produces the following traces:-



Taking "Rubicon" as a first try, WORDFINDER skips traces 2 and 3 and finds the word "sentry" in trace 4. However, there is no English word in trace 5 and WORDFINDER realises that it has made the wrong choice. On taking "Ruby" as a second try, WORDFINDER produces the desired output.

Finally, it was mentioned that in order to resolve the incorrect joining of chords a transcription program must be able to produce a list of all alternative transcriptions so that they may be submitted for subsequent processing. One example which demonstrates that WORDFINDER can produce such lists is as follows:- For the chords A/ N/EU S/I T/IN WORDFINDER produces the following traces:-



On taking the longest word in trace 1 and skipping trace 2 WORDFINDER produces "anew sitting" . Taking the next longest word in trace 1, the longest word in trace 2 and skipping trace 3, WORDFINDER produces "a newsy tin". And finally taking the next longest word in trace 2, WORDFINDER is able to produce "a new sitting".

3.5 SOME CONCLUSIONS

The word boundary problem was viewed as two separate problems for the major part of the research; (i) producing output consisting wholly of English words from error free input, and (ii) predicting the correct word boundaries in the situations where two chords considered together produce only one word, yet considered separately produce two words. This approach resulted in algorithms which were unable to determine all the different ways in which a given sequence of chords could be transcribed.

When the two problems were considered together, an algorithm named WORDFINDER was designed which is not only capable of producing output consisting wholly of English words from error free input, but also of producing a list of all alternative transcriptions for a given sequence of chords. The idea was that such a list could be submitted for subsequent processing, e.g. syntax analysis, so that the correct word boundaries could be determined. However, there was not enough time in the present research to perform the processing of these lists.

The process of determining word boundaries is analogous to 'parsing' in computer compilers (Gries,1971) since many parsers scan the input string from left to right and compare 'tokens' (instead of chords) with terminals in a tree structure. It is therefore worth investigating whether the word boundary reconstitution algorithms cound be defined as a set of paradigms to "drive" a table driven parser.

Chapter 4

DEVELOPING TOOLS FOR THE CORRECTION OF ERRORS

In the previous chapter it was argued that a computer can translate error-free Palantype code into English text. The major factor affecting the quality of English text produced by Palantype transcription systems is the number of keying errors made by the machine operator. It is therefore important to consider operator errors in some detail and how they might be corrected.

4.1 ANALYSIS OF OPERATOR ERRORS

4.1.1 Method

In order to be able to analyse operator errors the transcription program was run with many files of Palantype data. This data consisted of thousands of palantype chords in computer readable form, and came from 5 different operators, - although 70% of the data came from only one of the operators (Miss Isla Beard). This is an important factor which has implications for the correction procedures described later. As there was relatively so little data from each of the other operators, no attempt was made to diagnose each operator's idiosyncracies and how they compared with each other.

The transcription program produced output for each Palantype data file that was input to it. When the transcription program was able to construct English words from a section of the input string of chords, they were output in lowercase characters. However, when the transcription program was unable to construct English words from a section of the input string, then those chords in that section were output in uppercase characters reflecting the keys pressed.

> This has become very important recent LI , ANT+ we are
> experiencing something which is new to us , relative
> IM POF+ RISH MNT . We THINCTH difficulties that we face
> as a nation HAFTH X TROR din RI ....

Fig. 4A Some examples of Palantype text in upper and lower case

Once the computer had produced output for each of the files that were input to it, the remainder of the error analysis was completely manual. It consisted of inspecting each piece of output and recording as much information as was thought relevant. This process was very time consuming.

The first thing to determine was whether the transcription program failed to produce English words because the operator had made a mistake or for some other reason. There were four situations in which the transcription program was unable to produce English words:-

1. A word which the operator had Palantyped correctly was not in the system dictionary. An example of this in figure 4A are the chords " IM POF+ RISH MNT ", which, by applying the rules of Palantype, should have produced the word "impoverishment". As "impoverishment" is not in the system dictionary, the transcription program resorts to outputting the chords in uppercase characters.

Although the dictionary was compiled to include many proper names, etc.,(Price,1971), operators record such a wide range of subject matter that they will inevitably encounter words which are not present in the dictionary.

2. An operators representation (although legitimate) of a word was different from the one(s) in the system dictionary. An example of this in figure 4A is the structure "recent LI", which should of course have been the word "recently". In this case the operator has keyed the first consonant of the final syllable as the final consonant of the penultimate syllable, i.e. the operator keyed "RI. SENT LI" instead of "RI. SEN TLI" .

This situation occurred very frequently with words ending in "Y", and for several operators (including the one that compiled the dictionary). Some more examples are as follows:-

English word	Normal convention	Alternative
correctly	CO REC TLI	CO RECT LI
simply	SIM PLI	SIMP LI
only	OE NLI	OEN LI
interested	IN TRE STT+	IN TRES TT+
question	CFE STJUN	CFES TJUN
responses	RE SPON SS	RES PON SS

some examples of ambiguous syllable boundaries

An operator's representation of a word was different from those in the system dictionary mostly because the operator had chosen different syllable boundaries within a word. However, this situation sometimes occurred because an operator's "short form" of a word was arguably legitimate, but different from the entry in the dictionary. When the dictionary was compiled, the operator was encouraged to include, where applicable, alternative representations for each word (Price, 1971). Nevertheless, with so many words and different representations, some inevitably missed being included in the dictionary.

3. The third situation in which the transcription program was unable to produce English words was when, due to some variable in the recording process, the data which was used as input to the Palantype transcription program had been corrupted in some way. An example of this occurred with an electronic interface between the Palantype machine and the computer. Consider the situation where an operator presses two chords, e.g. THINC TH (representing the words "think the"), but only releases the keys (signifying end of chord) for a very small fraction of a second between the

two chords. Then if, for some reason, an electronic interface missed the release of keys signal, the computer would be presented with the single chord THINCTH . Part of the research at Southampton University (Passingham,1980), has been to try and ensure that the keyboard is scanned at the correct time, and hence prevent this problem occurring.

Another problem concerned with the integrity of the data being input to the transcription program arose from the different sources of input data. The data used during the period of the research came via three different mediums. The first, already mentioned, was the electronic interface. Here, the data direct from the Palantype machine is transmitted by the electronic interface to the computer, where it is stored. The second medium was much more prone to error since data from several operators had been sent to us on the original Palantype printed rolls, leaving us the task of transforming this data into computer readable form ,without the use of a Palantype operator. Having a Palantype machine with an electronic interface to the computer, left only the laborious task of manually keying in the data chord by chord. This was performed at a very slow rate with one person reading the Palantype roll, while a second person operated the keyboard. The probability of keying mistakes in this process was high and also the data on the printed rolls was not always very clear, with the problems of "ghosted" characters etc.,(Price,1971). For these reasons this method was avoided as much as possible.

The third medium was also quite prone to error since data which arrived on computer printout, in the forms of rows and rows of Palantype chords, had to be transferred onto punched cards (with the risk of the errors inherent in that process) and read into the computer.

All data files were marked as to the medium via which they had been created.

4. The fourth situation in which the transcription program was unable to produce English words was when the operator actually made a keying mistake. An operator error is none of the 3 situations mentioned above, but is that situation where extra keys are pressed, keys are neglected to be pressed, or keys are pressed in place of other keys which result in a chord that is different from the intended chord.

Having defined an erroneous chord, it was possible to classify all erroneous chords in terms of insertions(extra keys), deletions(missing keys) and substitutions(keys pressed instead of other keys). A 'data collection' form was designed so that all analysed errors could be recorded with any relevant information. The form was partitioned into several columns; the headings were as follows:-

Error Chord	Intended Chord	Subs Del Ins	Keys	Error Distance	Position init med final	Remarks

Error chord - This column shows the chords which contain errors

Intended chord - This column shows the chords which the operator really intended

Subs Del Ins - This column shows the classification of errors in terms of substitutions, deletions and insertions

Keys - This column shows the difference between the error chord and the intended chord in terms of the actual keys pressed

Error difference - This column shows how close, in terms of key depressions, an error chord was to the intended chord. It is explained more fully in section 4.2.2.

Position - This column shows which syllable of a word is in error (since a chord usually corresponds fairly closely to the spoken syllable.

Remarks - This column contains various kinds of information, e.g. how closely together errors occurred

From the data collection forms the following information was elicited.

1. Error frequency count

A tally was kept of the number of times that each key on the keyboard had been the erroneous extra key, deleted key or substituted key, i.e. the number of times a key was present when it should not have been or vice-versa (this was achieved by inspecting the 'keys' column of the data collection forms). This information was useful in the development of a 'weighting' routine, which when presented with an erroneous chord could help determine the most probable chord intended in its place. The 'weighting' routine is described in detail later.

2. Measurement of error difference.

Each erroneous chord was 'measured' to see how close, in terms of key depressions, it was to the intended chord (column 5 on the data collection form). This information, like the error frequency count, was also useful in helping to determine the most probable chord intended in place of the erroneous chord.

3. Error locality

Information about the locality of errors included how closely together errors occurred and whether an error was at a word boundary or in the middle of a word. This information was stored in column 6 and in the remarks column of the data collection form. When presented with a string of chords where it is known that at least one of them is in error, the above information is valuable in the error identification process.

4. Phonetic properties and grammatical classification of words

It was thought that this information might be useful as an extra pointer to the source of error by analysing whether, for example, there was any correlation between errors and the phonetic properties or grammatical classification of the word in which they occurred. The analysis of errors was very time consuming and in the event there proved to be too little time to record this information.

4.1.2 Results of the error analysis

The results presented here are from an analysis of actual operator keying mistakes. As mentioned earlier, this does not include chords which "appeared" to be in error due to either an incomplete lexicon, alternative representations or corruption of the input data.

The first results presented are those of the error frequency count for each key on the keyboard. Fig. 4D is a histogram showing the involvment of each key in erroneous chords, as a percentage of the total number of analysed errors (approximately 2500). It can be seen that the "+" keys were the cause of most errors. Fig. 4E is a breakdown of the total number of errors in terms of substitution, deletion and insertion errors. Substitution errors constituted only 19.6% of the total number of errors, whereas deletion and insertion errors constituted 44.5% and 35.9% respectively. Fig. 4F is a breakdown for each key showing whether its involvement was a substitution, deletion or insertion error. Fig 4G shows that there were more errors with the left hand consonants than with vowels or right hand consonants.

The next set of results presented are those of error distance measurement. It was found that 90% of all analysed errors differed from the intended chord in only one key. Fig 4H shows this. An equally important point is that no errors were found to have a key difference value of greater than 3.

The final set of results are those from an analysis of the location of errors. Erroneous chords were examined to see whether they constituted the starting syllable of a word (i.e. initial chords), medial syllables or final syllables (i.e. having a pointer to an English word). If an erroneous chord was a final syllable as well as an initial syllable,i.e. a word in its own right, then it was counted simply as an initial syllable since it would be in the list of initial chords that the correction program would look for the intended chord. It was found that 53% of all analysed errors occurred in an initial chord, 39% occurred in a final chord and 8% in a medial chord (however, these figures are not so meaningful since there are more initial and final syllables than medial syllables).

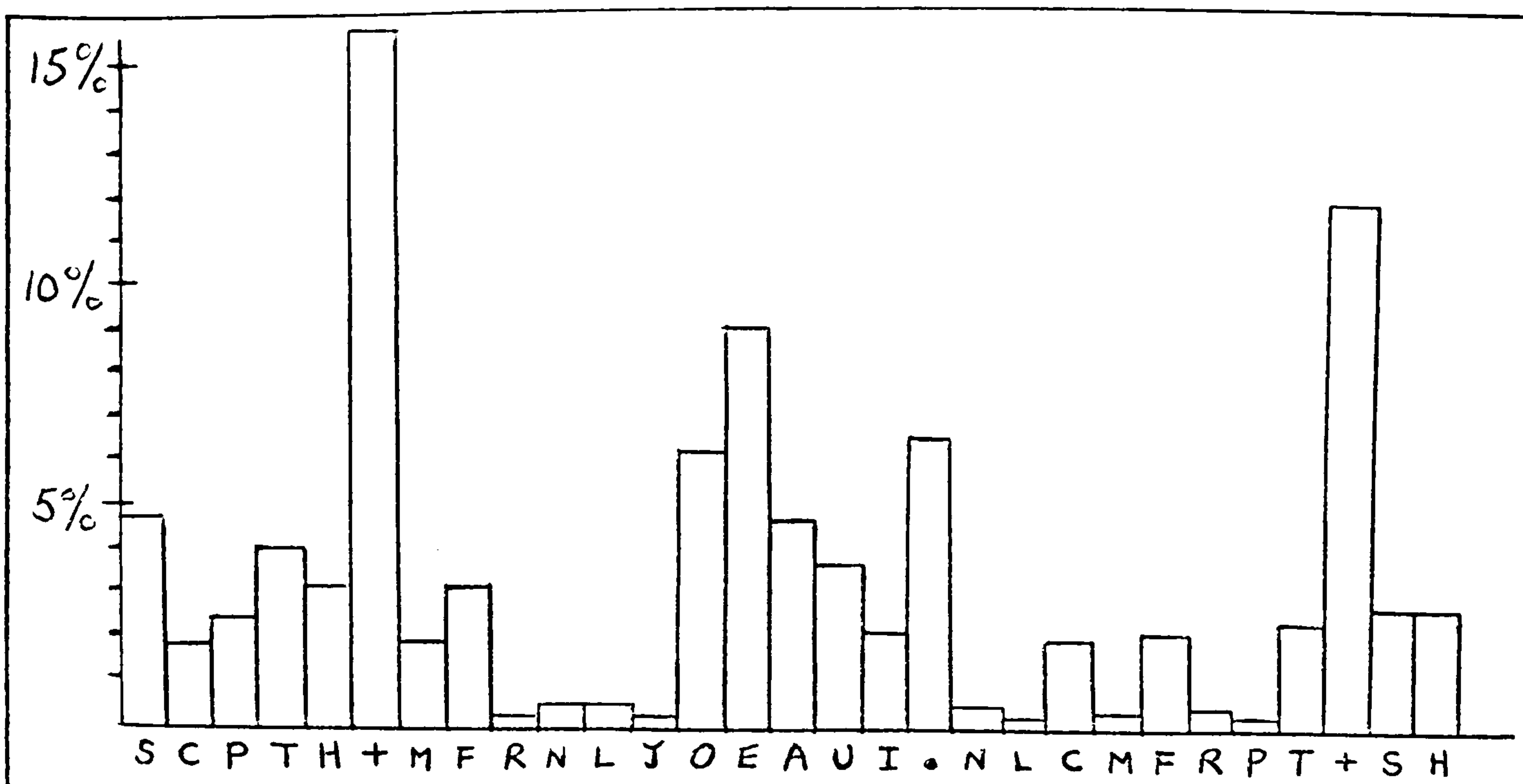


Fig.4D Percentage involvement of each key in erroneous chords

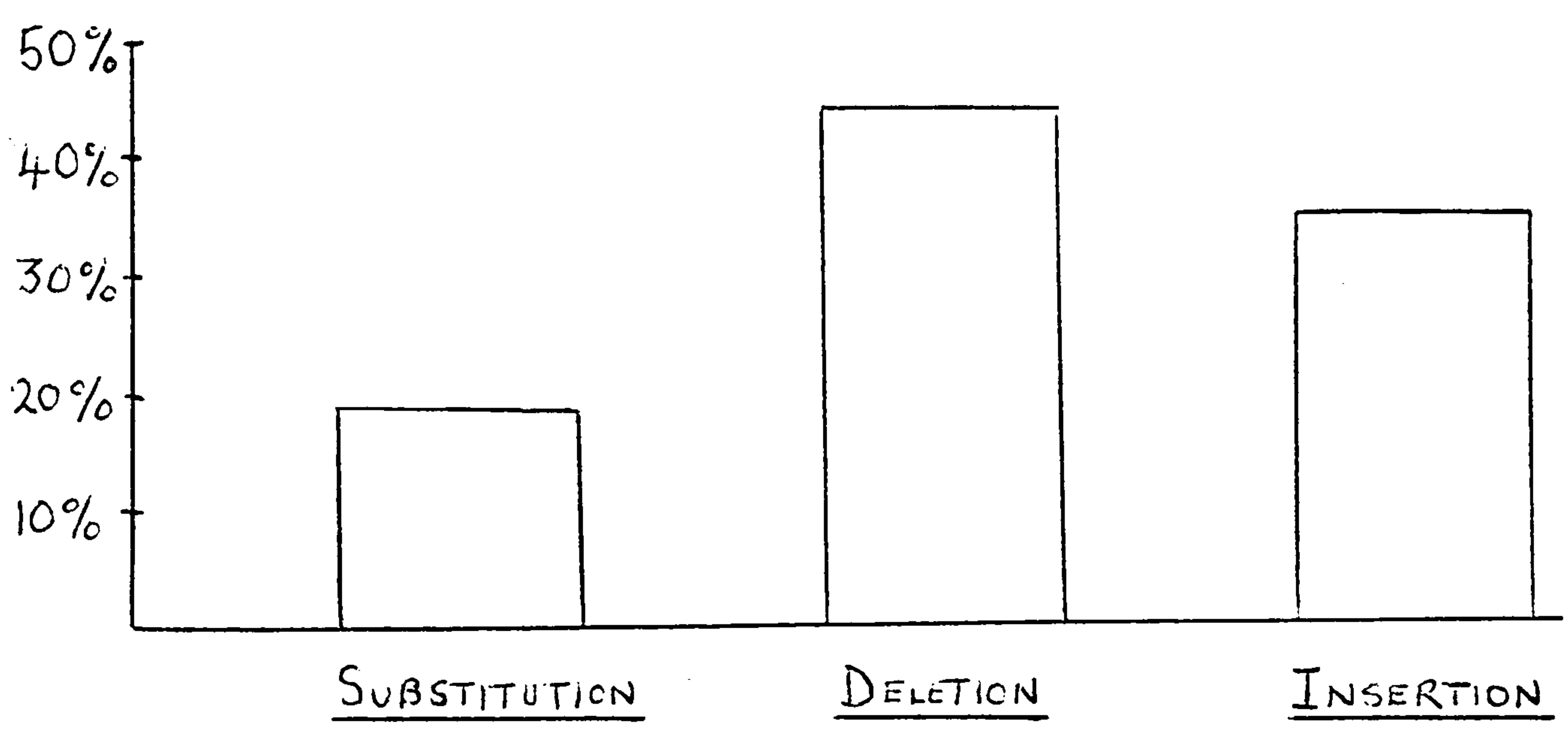


Fig.4E Breakdown of errors (insertions, deletions and substitutions)

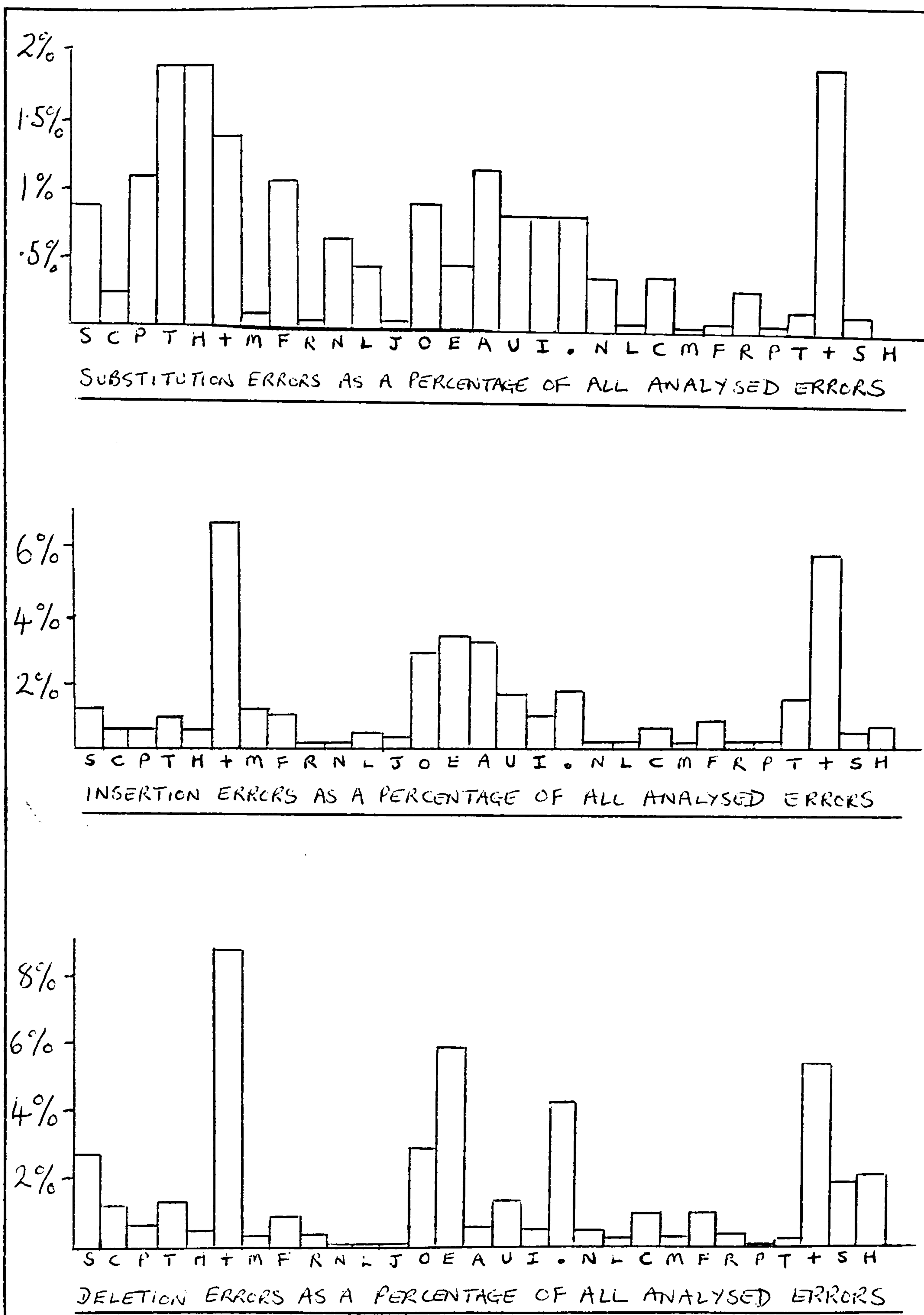


Fig.4F Percentage deletion, insertion and substitution errors for individual keys

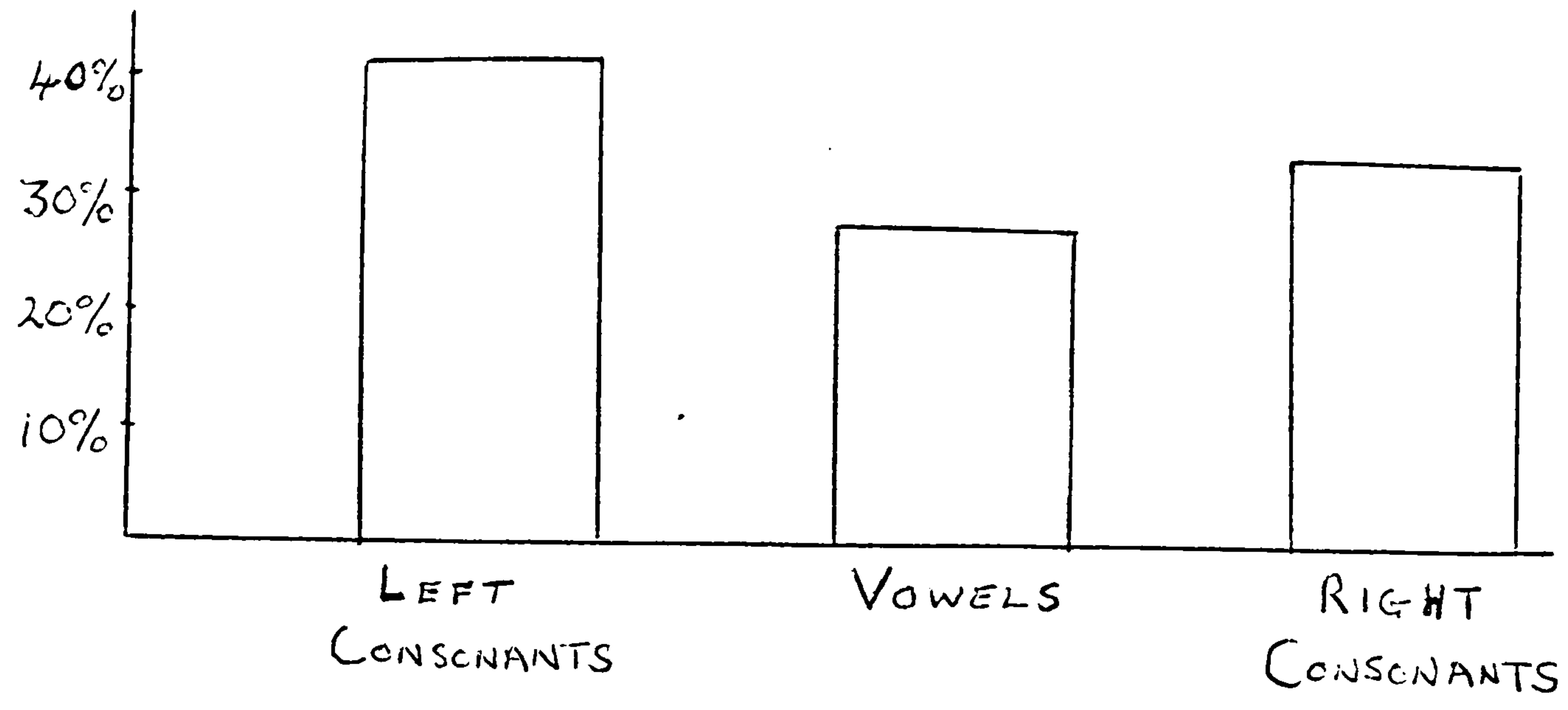


Fig.4G Errors relative to keyboard 'zones'

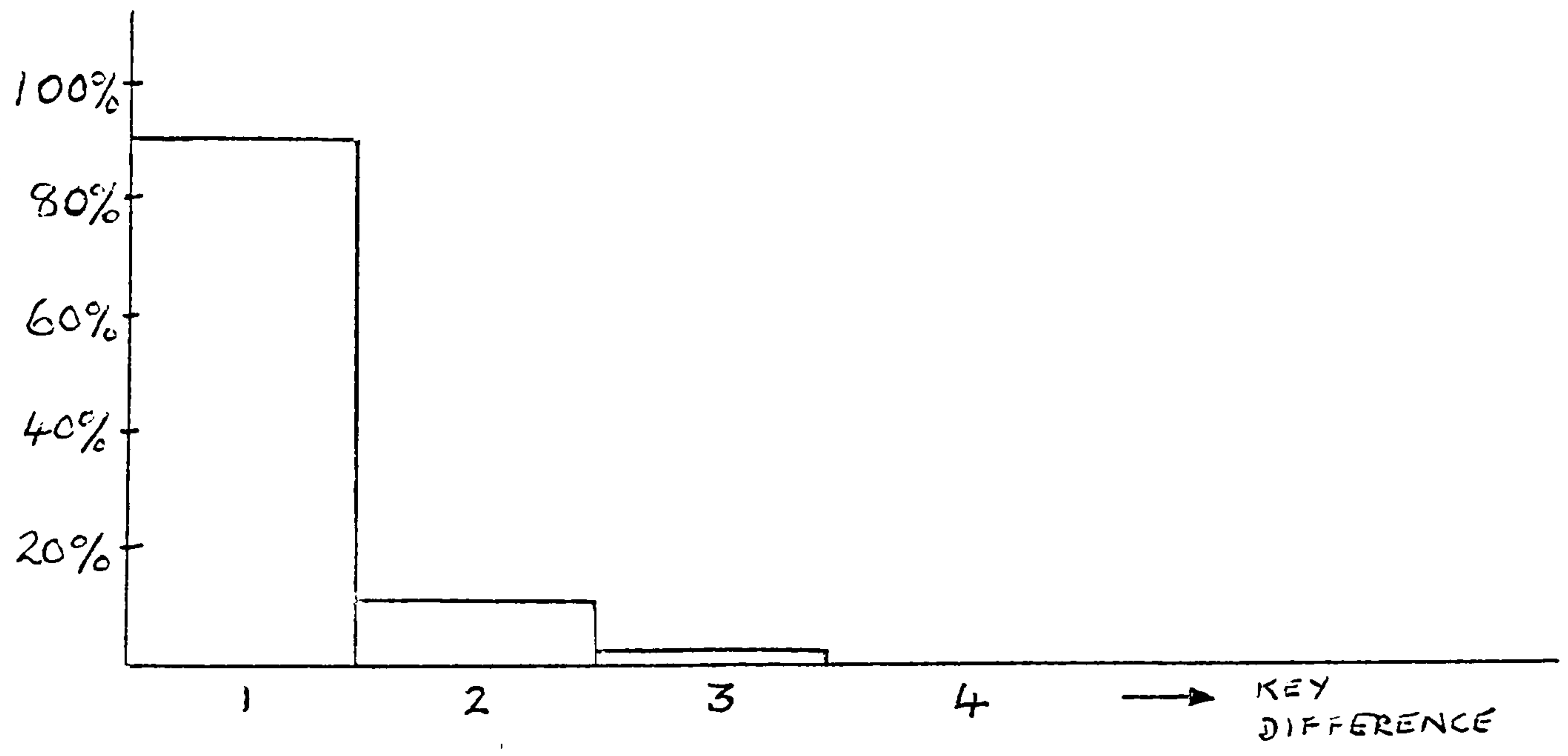


Fig.4H Difference between intended chord and erroneous chord in number of keys

4.2 THE DETECTION AND CORRECTION TOOLS

4.2.1 A device for detecting errors

The first tool required was a device for detecting when an error was present in the input string. The reason for this is clear; it is necessary to know when an error is present before it can be corrected. This device was already provided by the transcription algorithm, since as it always produced output consisting wholly of English words from error-free input (see Chapter 3), then as soon as it attempted to output anything other than English words, it was known that at least one of the following was true:-

1. the operator had typed chords for a word which was not in the system dictionary

2. the operator had typed chords in a different manner from those listed next to the dictionary entry, i.e. an alternative representation

3. the chords pressed on the Palantype machine were different from those which reached the computer to be input to the transcription program

4. the operator had made a keying mistake

If a special program device could be written to cope with situations 1, 2 and 3 above, then the transcription program would provide a tool for detecting errors. Such a device is the subject of the next chapter.

4.2.2 An error metric

We saw in the previous chapter that the transcription process was based on comparing a chord in the input string with chords in the tree-structured lists. If a 'match' was found, then the next chord in the input string was compared with the list of chords at the next level in the tree structure, and so on. When a chord or chords appeared in the output in 'not found' format, (i.e. in uppercase characters reflecting the keys pressed), it is true to say that this occurred because the chord or chords failed to find a 'match' in the tree-structured lists. For example, if the 'not found' chord is the first one in the recording session, it is because it failed to find a match in the list of chord which can start a word, i.e. the initial chord list.

The second correction tool developed was an error metric to determine how close, in terms of key depressions (this phrase is explained subsequently), a chord which failed to find a match was to chords in the lists against which it was being compared. This tool proved very successful in the process of trying to find an erroneous chord and automatically replace it with the one the operator intended. It functioned in the following way.

Each chord in the list differs from the 'not found' chord, in that different keys have been pressed, a different number of keys have been pressed, or both. We saw earlier that these difference were categorised as insertion, deletion and substitution errors. For each chord in the list there is an associated 'key difference' value to the 'not found' chord. The value is the number of different keys between the two. For example, if the chord which failed to find a match (not found) was R/I, then fig.4I shows a list of chords against which it might have been being compared, with their associated 'key difference' values and error classification.

chord list	associated key difference from R/I	type of error
/I	1	insertion (added R)
R/A	1	substitution (I for A)
FR/I	1	deletion (lost F)
R/I.	1	deletion (lost .)
R/IS	1	deletion (lost S)
R/IT+	2	deletion (lost T+)
C+R/I.T+	5	deletion (lost C+.T+)

Figure 4I: Key difference values determined by error metric

The way in which the error metric determines the key difference value for a chord is as follows. Each chord is represented in the computer as a 29-bit pattern, one bit per key on the Palantype keyboard. The first bit of a chord in the list is compared with the first bit of the "not found" chord and if they are different, a count called KEYCOUNT is incremented. The second bit of the chord in the list is then compared with the second bit of the "not found" chord and so on, until all 29 pairs of bits have been compared. Each time a pair of bits differed, KEYCOUNT would be incremented by 1. Comparing the bit pattern of the "not found" chord R/I with FR/I, we can see in Figure 4J that they only differ in the 21st position (F), and therefore KEYCOUNT is 1.

BIT	KEY	R/I	FR/I
28	S	0	0
27	C	0	0
26	P	0	0
25	T	0	0
24	R	0	0
23	+	0	0
22	M	0	0
21	F	0	1
20	R	1	1
19	N	0	0
18	L	0	0
17	J	0	0
16	O	0	0
15	E	0	0
14	A	0	0
13	U	0	0
12	I	1	1
11	.	0	0
10	N	0	0
9	L	0	0
8	C	0	0
7	M	0	0
6	F	0	0
5	R	0	0
4	P	0	0
3	T	0	0
2	+	0	0
1	S	0	0
0	H	0	0

Figure 4J: Comparison of bit patterns

For insertion and deletion errors, KEYCOUNT can be used directly as the key difference value associated with that chord. Substitution errors however have to be dealt with a little differently. When for example, we compare the bit patterns of R/I and R/A, we will produce a KEYCOUNT value of 2. As the operator has mistakenly pressed only one key (an I instead of an A, it would seem more logical for R/A to have an associated key difference value of 1). Therefore, with substitution errors we cannot use the KEYCOUNT value directly. To obtain a key difference value for substitution errors it was necessary to keep a count of the number of keys pressed in each chord as well as producing a KEYCOUNT value. The flowchart in Figure 4K shows how an associated key difference value can be determined for any kind of chord error. When comparing the "not found" chord with the list of chords in which it was supposed to find a direct match, the error metric will give us an indication for each chord in the list, of the likelihood of it being the one the operated intended in place of the "not found" chord. Thus, if the key difference-value between the "not found" chord and a chord in the list against which it is being compared, is low, then the likelihood that the chord in the list is the one the operator intended is relatively high, and vice-versa. This means that in order to determine whether the relative probability of a particular chord being the one the operator intended, one needs only to inspect the 'key difference' value of that chord.

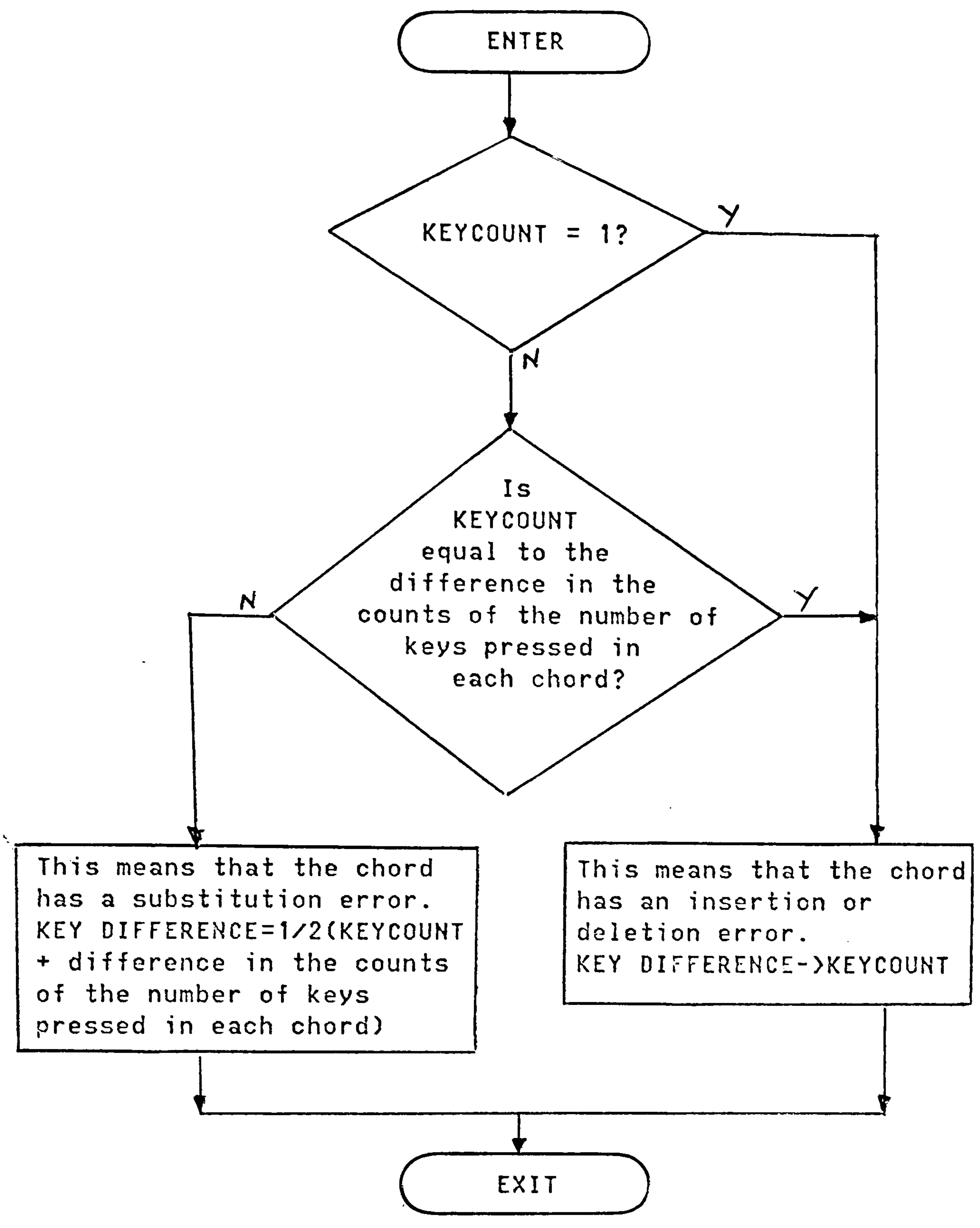


Fig. 4K: Determining the key difference value for a chord

An analysis of the kinds of errors operators make was described earlier in this chapter, but an important fact relating to the error metric is that by far the majority of operator errors are only one "key difference" from the intended chord. There is a small percentage of errors that have a "key difference" of two, and an even smaller percentage of errors that have a "key difference" of greater than two.

4.2.3 A "Weighting" routine

This third correction tool was designed to complement the error metric by taking a list of chords having the same key difference values and ordering them so that of the chords in the list the most likely intended by the operator is placed at the top of this list. The criteria on which the list was ordered are based on information about insertion, deletion and substitution errors elicited from the analysis of operator errors. For example, it was found that the most common mistake made by one particular operator, was that she neglected to press the left hand '+' key when it should have been included in a chord. Clearly then, if there is a chord in the list we are ordering which differs from the "not found" chord in only the left hand '+' key position, then this is the chord most likely intended by the operator and is placed at the top of the list.

4.3 TESTING THE TOOLS IN A SIMPLE CORRECTION PROCEDURE

To get some indication of the effectiveness of the correction tools, they were tested in a simple correction procedure. For this purpose, at the point when the transcription program detected an error in the input string (i.e. when it was about to output a chord or chords in "not found" format), the latest chord in the input string was assumed to be erroneous. Instead of outputting a chord or chords as "not found", the correction procedure described below was invoked to correct the latest chord in the input string. Inevitably, there were many times when the latest chord was not in error, yet was still subjected to the correction procedure. Nevertheless, for the purpose of testing the effectiveness of the correction tools, this factor was not preventive to the aim. After implementation of the correction procedure, a manual inspection of the "corrected text" informed us of the occasions when the latest chord was actually in error and when it was not. Therefore, on the occasions when the latest chord was in error, it was possible to assess how effective the tools had been in correcting errors.

4.3.1 A simple correction procedure

The first task of the correction procedure was to save the contents of output buffers, arrays, etc. so that whether the error was corrected or not, the transcription could resume at the point at which it had been suspended in order to attempt the correction. Having assumed that the latest chord was in error, the next stage was to submit those lists of chords in which it failed to find a match, to the error metric. If the chord immediately preceding the latest chord did not have a pointer to an English word (i.e. if it was not a possible word boundary), then there was only one list submitted to the error metric, i.e. the list of chords at the next level in the tree structured lists to the preceding chord in the input string. But, if the preceding chord did have a pointer to an English word (i.e. there was a possible word boundary), then an additional list was submitted to the error metric, i.e. the list of initial chords (chords which can start a word).

A threshold value was introduced into the error metric so that it only produced two lists of chords; one with a "key difference" value of one, and the other with a "key difference" value of two. The occurrence of errors with a "key difference" value of greater than two was thought to be relatively so few as to justify not producing lists for them. Of course, this imposed a limitation on the correction procedure in that it was unable to correct any error with a "key difference" of greater than two.

Having produced a list of chords with a "key difference" value of one, the next stage was to order this list using the "weighting" routine. Then the chord at the top of the list was substituted for the "not found" chord in its position in the input string. That part of the input string that was currently involved in building words when the error was detected, was then re-transcribed. If the re-transcription was successful, i.e. it produced English words without encountering any more "not found" chords, then the output buffers, arrays, etc. were restored and normal transcription resumed. Otherwise, the next chord in the ordered list was substituted for the "not found" chord in its position in the input string, and another re-transcription attempted. This process was repeated until a successful re-transcription was achieved or until all the chords in the list had been used. In the latter case, the list of chords with a "key difference" of two was "ordered" and the above process of substitution and re-transcription was performed for this list. Because of the way the lists were ordered, the first successful re-transcription is the one most likely intended by the operator. If after trying all the chords in both lists, a successful re-transcription had still not been achieved, then it was assumed that a different chord to the one being corrected was in error, and that a more 'intelligent' correction procedure was necessary in order to correct it.

4.3.2 Example of output

The output in Figure 4L was produced from two separate transcriptions of the same input string. The text on the left was produced by a transcription program incorporating no error correction, and the text on the right was produced by a transcription program incorporating the simple correction procedure described above.

Uncorrected output	Corrected output
I though it P+OUT+ not work, ant possibly you thought it would not +FUC, but now I find its in I use the tell FON as much as I did PFOR I went deaf	I thought it would not work, ant possibly you thought it would not work, but now I find its in I use the tell FON as much as I did performing went deaf

Fig.4L Sample Output

Although this example is small it serves to demonstrate the basic concepts involved in the simple correction procedure. When the error was detected in the first line, the latest chord was P+OUT+, and in fact this was the erroneous chord. The "P" key had been pressed instead of the "F" key. Our simple procedure successfully corrected this substitution error.

In the second line, the word "ant" should have been "and". This error was not detected since the transcription algorithm was still producing output consisting of English words.

When the error was detected in the third line, the latest chord was +FUC, and again this was the erroneous chord, (a deletion error made by the operator because she had neglected to press the "." key). This error, like the one in the first line, had a "key difference" value of one, and was corrected to the chord the operator intended.

When the error was detected in line four, the latest chord was AS, but the error was in fact in the chord FON, and therefore the correction procedure was unable to correct this error.

When the error was detected in line five, the latest chord was I, but again the error was in the preceding chord; PFOR. PFOR should have been P+FOR, which is the chord for the English word "before". In attemtping to correct "I", the correction procedure found the chord MIN which has a "key difference" value of two, and with PFOR makes the word "performing", thus satisfying a successful re-transcription, even though the correct result has not been obtained.

4.3.3 Some Conclusions

The major factor affecting the quality of English text produced by Palantype transcription systems is the number of errors made by the machine operator. An analysis of transcribed texts produced by the computer showed that there were four situations in which a transcription program was unable to produce English words. Only one of these situations was attributable to actual operator error. In turn, an analysis of these errors showed that there were twice as many deletion and insertion errors than substitution errors. Also, almost 90% of all analysed errors differed from the intended chord by only one key.

Detection and correction tools were developed and tested in a simple correction procedure. This procedure always assumed that the latest chord in the input string was in error. Although this meant that at certain times the correction tools were invoked for a chord which was NOT in error, it was still possible to ascertain that the tools were 35% successful in correcting chords which WERE in error.

No attempt was made in the error analysis to examine the ergonomic aspects of the keyboard and determine why a particular recurring error was being made.

Chapter 5

IMPROVING THE QUALITY OF THE OUTPUT TEXT BEFORE ERROR CORRECTION

5.1 INTRODUCTION

It was shown in the previous chapter that when the transcription program failed to produce English words it was because one of the following situations was true:-

1. the operator had typed chords for a word which was not in the system dictionary

2. the operator had typed chords in a different manner from those listed next to the dictionary entry, i.e. an alternative representation

3. the chords pressed on the Palantype machine were different from those which reached the computer to be input to the transcription program

4. the operator had made a keying mistake

In order to have a device for detecting the presence of an error, it was necessary to investigate the possibility of eliminating 1, 2 and 3 above.

5.1.1 Incomplete lexicon

It was stated in the introductory chapter that no matter how large the dictionary becomes, the operator will inevitably encounter words which are not in it. So, there is a FULL STOP; it is not possible to eliminate situation 1. However, just because the dictionary can never contain ALL the obscure, foreign, made-up and swear words that the operator will inevitably encounter, does not mean that it should not contain ALL those words that an operator is reasonably likely to encounter. At the moment, the dictionary does not contain all the words that an operator is likely to encounter. A glaring omission is the possessive form of proper nouns. Words such as "Britain's", "London's", "Churchill's", etc. have not been included in the dictionary although they readily occur in everyday English. To overcome this problem, the system dictionary could be updated to include the possessive form of each proper noun. However, this would be a very significant increase in dictionary size and might have implications for tree-searching times, etc. An alternative approach adopted at both N.P.L. (Price,1971) and in the present research is as follows. When the transcription program assumes a chord to be erroneous that ends in "S" (final), it deletes the "S" from the erroneous chord and performs a retranscription of the relevant part of the input string. If the retranscription is a successful one where the modified chord has matched with a proper noun (since dictionary entries for proper nouns are marked to indicate that they start with a capital letter), then apostrophe "s" ('s) is added to the proper noun before it is output. If the retranscription is unsuccessful, then the "S" is added to the erroneous chord again, and the normal correction procedure is invoked.

An alternative approach to this and one which could reduce the dictionary size significantly would be to implement Winograd's (1972) word ending algorithm. The flowchart in fig.5A is taken from Winograd's book and is designed to handle not only possessives, but a number of other inflectional endings. For example, "-nt" for negatives, "-s" and its various forms for plural nouns and singular third-person verbs, "-ing", "-ed" and "-en" verb forms etc.

If such an algorithm were implemented, there would be no need to have words like "running", "beating", "pleasing", "pleased" etc. since they are inflected forms of the verbs "run", "beat" and "please". Of course one would have to ensure that there was some mechanism to relate chords to endings in various contexts. An interesting feature of the Winograd algorithm is that although it has the ability to capture generalities as well as treating special cases, not all 'exceptions' need be explicitly included in the analysis program. For example, if the word 'was' is included as a dictionary entry, it will never be subjected to the analysis algorithm. Morphographemic analysis (dealing with written language) alone is sufficient to do a great deal of the interpretation of English word endings. Some systems (e.g. Thorne et al, 1969) have used it to avoid having a dictionary of open class words altogether (see Chapter 9, section 2.3 of this thesis). The implementation of the Winograd algorithm would not adversely affect the parsing techniques described in Chapter 7.

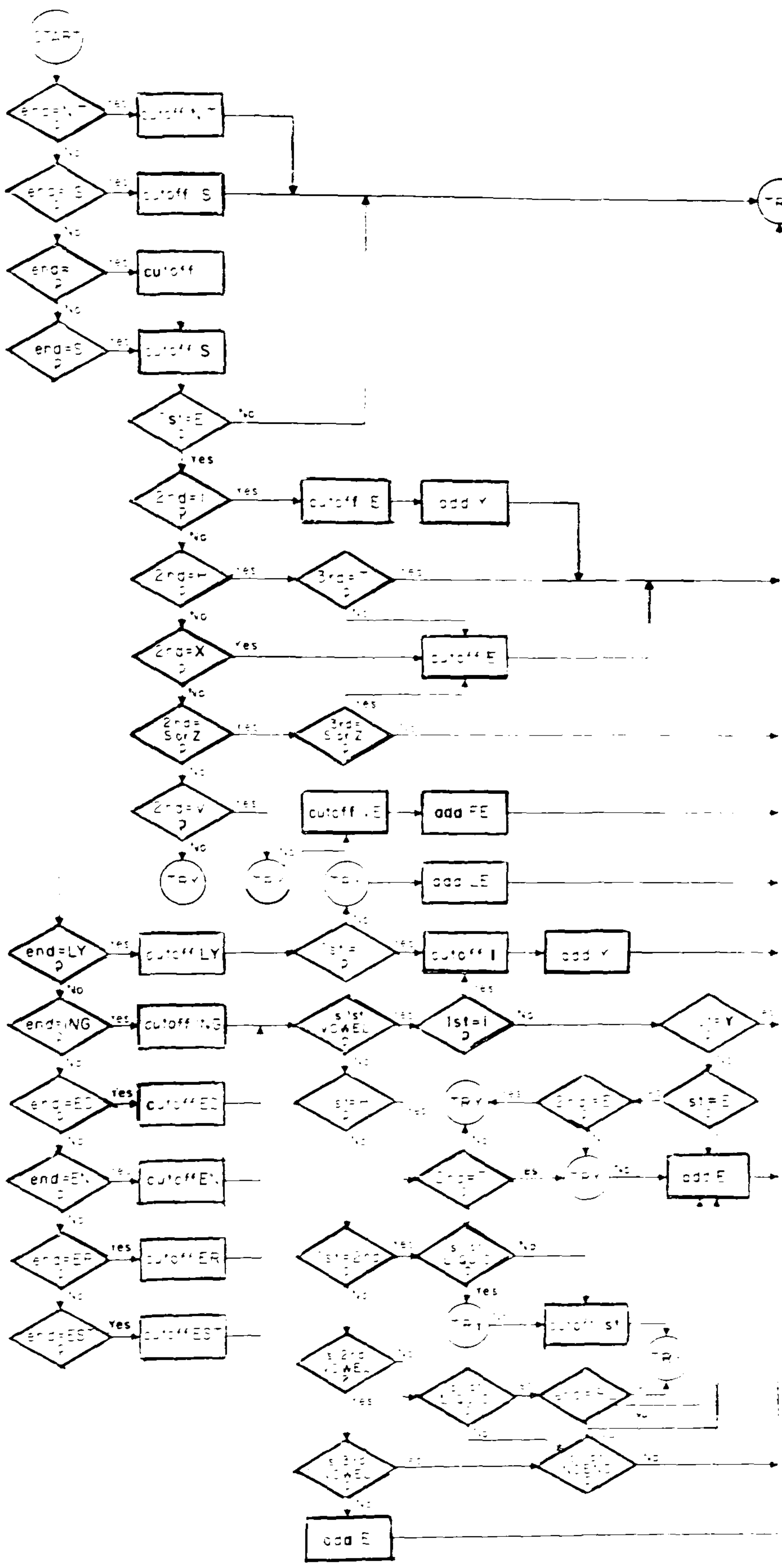


Fig. 5A Analysis of English Endings

BEST COPY

AVAILABLE

Variable print quality

5.1.2 Alternative representations

It was mentioned in the previous chapter that operators frequently had an alternative representation for words ending in "y". Some more examples are given in fig. 5B.

English word	Dictionary rep.	Alternative
correctly	CO REC TLI	CO RECT LI
recently	RI. SEN TLI	RI. SENT LI
history	HI STRI	HIS TRI
only	OE NLI	OEN LI
entirely	EN TAIR LI	EN TAIR LI
surely	SHOU RLI	SHOUR LI
simply	SIM PLI	SIMP LI
cheaply	CHEA PLI	CHEAP LI
quickly	CFI CLI	CFIC LI
etc.		

Fig. 5B Examples of alternative representations of words ending in "y" .

To overcome this problem, the system dictionary could be updated to include two entries of all words ending in "y". Again, this would increase the size of the dictionary significantly and so a different approach was adopted. When the transcription program assumes a chord to be erroneous that ends in "I", it deletes the last consonant of the previous chord, adds it as the first consonant of the erroneous chord and performs a retranscription of the relevant part of the input string. If the retranscription is successful, then it is assumed that the chord has been corrected. If the retranscription is unsuccessful, then the chords are restored to what they were before this "special" correction attempt was made, and the normal correction procedure is invoked.

For alternative representations of words not ending in "y" a different solution was adopted. During the analysis of output texts it became apparent that there were between 20 and 30 words that were consistently represented by an alternative Palantype chord sequence. Some of these words for which operators had a legal/valid alternative representation are listed in fig. 5C.

English word	Dictionary rep	Alternative
responses	RE SPON SS	RES PON SS
interested	IN TRE STT+	IN TRES TT+
signal	SI C+NL	SIC+ NL
system	SI STEM	SIS TEM
question	CFE STJUN	CFES TJUN
and	AN+	ANT+
produced	PROE T+EU ST+	PRO T+EU ST+
viewer	MFEU.R	MFEUR
university	EUN FUR STI	EUN MFUR STI
etc.		

Fig. 5C More alternative representations

It can be seen that this could be quite a serious problem, especially if an alternative representation is for a word that occurs as often as "and". The solution to this problem was to update the system dictionary with the alternative representations. However, there was no easy way of updating the system dictionary, without running a program which produced the tree structures for ALL 75,000 words. As the execution of this program took such a large amount of computing resources (a few hours processor time) an alternative method was searched for. The method adopted was to make the dictionary appear as though it had been expanded by having a pre-processor in front of the transcription program which continually checked for chord sequences which matched with any of the words in the above list. If a match was found, the chord string was modified to the normal convention as in the dictionary. The alternative solution which involved developing a program to update the tree structures was not very attractive because it would have been a much more time consuming task.

A further problem was the frequent omission of the "H" key in the chord "TH". This causes the letter "T" to be output, instead of the word "the". As "the" occurs so frequently, a few mistakes of this kind can soon affect the readability of the output text. To overcome this problem the chord "TH" was substituted for "T" in all situations except where the chord which succeeded "T" was a valid successor to "T". For example, in the string "T" "T+/E", "TH" would not be substituted for "T", as "T+/E" is a valid successor to "T" in the chord requirement for the word "today".

5.1.3 Data integrity

An analysis of output texts showed that two separate chords were frequently transmitted as one chord. Whether this was because the operator did not release the keys sufficiently between chords, or the keyboard was not 'scanned' at the right time, cannot be proven for the data used on this project. However, Passingham 1980, has shown that 'scanning' the keyboard at the wrong time in electronic interfaces can certainly result in two chords being transmitted as one. Some examples are given in fig. 5D.

TO AS	TOAS
HFOT I	HFOIT
TO A	TOA
THAT TH	THATH
SHOUR TH	SHOURTH
HFEN IN	HFEIN
THAT IS	THAITS
THAN IT	THAINT
HI.R TH	HI.RTH
TO TH	TOTH
THIS IS	THIS
THAT AT	THAT
etc.	

Fig. 5D Examples of Two chords transmitted as one.

It can be seen from the above table that a common factor is that each sequence contains small 'function' words that can be said together very quickly and can be typed very quickly. To overcome this problem, chords assumed to be erroneous were tested to see if they contained small function words such as IN, AS, TO, IT, IS, I, TH, etc. If so, they were split into two chords; one chord containing the function word and the other chord containing the remaining keys from the original chord. The relevant part of the input string was then retranscribed. This worked quite well for chords which did not share common keys, e.g. TO AS, TO TH, etc. For chords which share common keys, taking the function word out as one chord leaves something which is different from the intended chord. For example, taking IN out of HFEIN leaves HFE, but the intended chords were HFEN IN (when in). The solution to this problem was rather crude in that the erroneous chords were scanned for particular combinations such as THATH, HFEIN, etc. This type of problem has serious implications for combinations of chords where all of the function word is part of the first chord. For example, everytime THAT or THIS is received, how can the transcription program be sure that THAT AT or THIS IS was not intended? Syntax analysis may seem an obvious answer, but due to the time factor it was not applied to the above problem.

5.2 SOME CONCLUSIONS

In order to have a device for detecting the presence of an error in the input string, it was necessary to investigate the possibility of eliminating the three other situations in which the transcription program was unable to produce English words. The conclusions of this investigation are that although it is not possible to eliminate them completely, their adverse effect can be reduced.

It is worth investigating the possibility of implementing the Winograd 'word ending' algorithm. If implemented, the size of the dictionary could be reduced significantly. This has an important implication in that dictionary search times may be faster, which would leave more time for on line syntax analysis.

Chapter 6

THE AUTOMATIC CORRECTION OF ERRORS

6.1 COMPLEXITY OF ERRORS

As an error can occur at any point in the input string, methods for determining the actual chord or chords which are in error can be quite complex.

Consider the situation where two chords, C1 and C2, of the input string have been scanned. C1 is a valid start of a word (there is a match for it in the list of initial chords), but there is no match for C2 in the list of chords which can succeed C1, or in the list of initial chords. Also, C1 does not constitute a word in its own right. Under these circumstances, although only two chords of the input string have been scanned, the possible sources of error are already quite numerous:-

1. C2 could be in error in that it should have had a match in the list of successors to C1. For example, if the operator was intending to type the chords " P+/E. P+/IS " (babies) but typed " P+/E. SP+/I " by mistake, there would be no match for SP+/I in the list of chords which can succeed " P+/E. " . See fig. 6A and 6B for a subset of the dictionary of English words and corresponding data structures.

No.	English word	Palantype chords
22	abductor	AP T+UC TR
42	babies	P+E. P+IS
43	baby	P+E. P+I
48	bacon	P+E. CN
52	badger	P+A JR
57	bakery	P+E. CRI
69	can	CAN
85	father	FA THR
97	greed	C+RI.T+
98	greedy	C+RI. T+I

Fig.6A Subset of English word dictionary

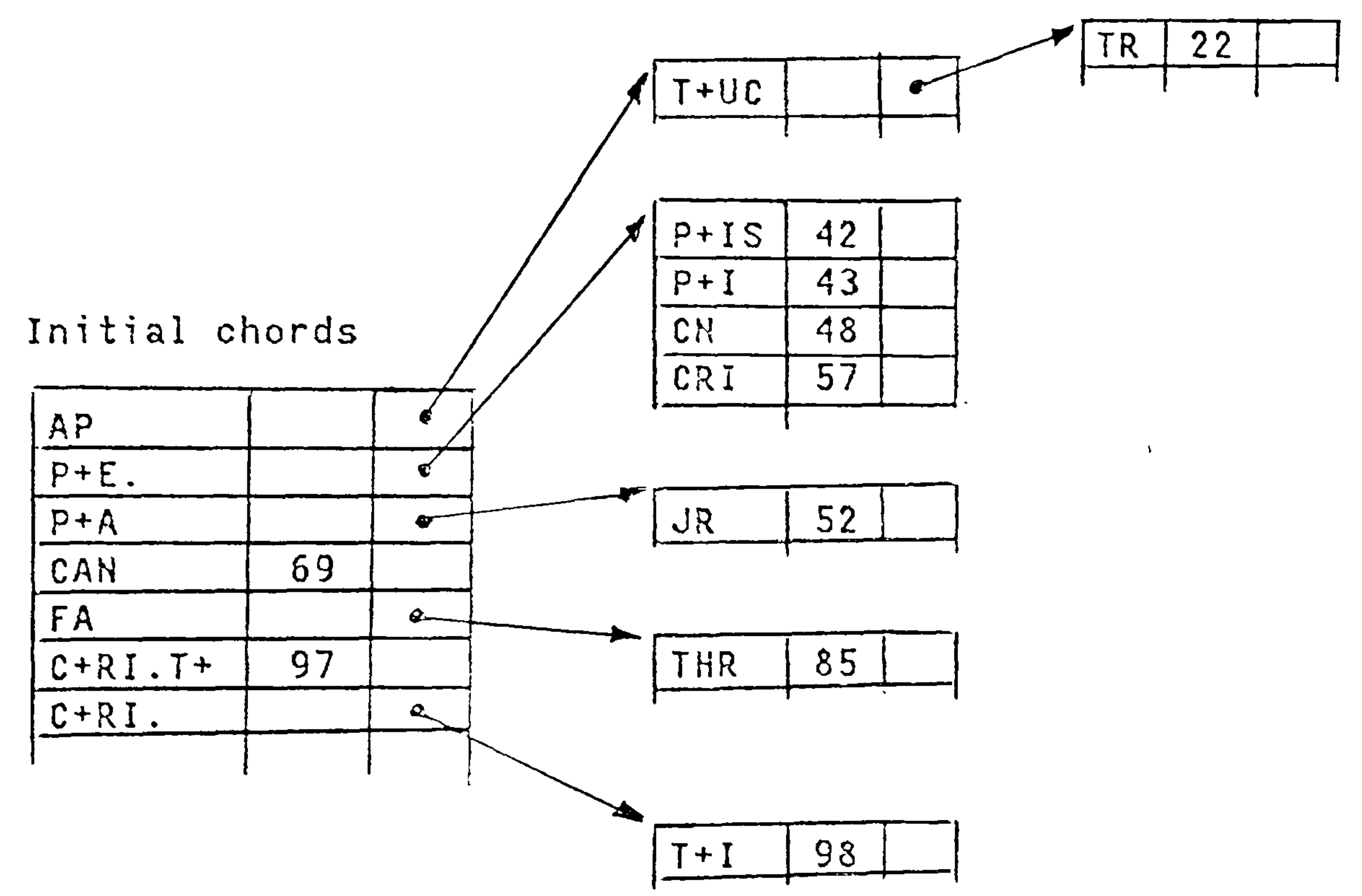


Fig.6B Data structures for 6A

2. C1 could be in error in that it should have been a different initial chord. For example, if the operator was intending to type the chords " P+/A J/R " (badger), but typed " F/A " instead of " P+/A ", then although F/A is an initial chord, there would

be no match for " J/R " in the list of chords which can succeed " F/A " , i.e. there is no word "fadger" or that starts with "fadger" (fadger...).

3. C1 and C2 could both be in error in that C1 should have been a different initial chord which was also a word in its own right, and C2 should have been the start of the next word (i.e. an initial chord). For example, if the operator was intending to type the chords for the two words "greed can", which are " C+RI.T+ C/AN ", but typed C+RI. and C/N, then again there is no match for C/N in the list of chords which can succeed C+RI. , i.e. there is no word "greecon".

4. C1 and C2 could both be in error in that C1 should have been a different initial chord and C2 should have been a chord which had a match in the list of successors to the intended C1. For example, if the operator was intending to type the chords "C+R/I. T+/I " (greedy), but typed " CR/I. T/R ", then again there is no match for T/R either in the list of chords which can succeed CR/I. ,i.e. there is no word "creeter".

5. Either C1, C2 or both could be in error in the situations where extra chords have been pressed or chords have been missed being pressed. For example, if the operator was intending to type " /AP T+/UC T/R " (abductor), but typed /AP T/R (missing the chord T+/UC) then it is known that at least one error is present in the input string as there is no word "aptor".

As more chords of the input string are scanned, the possible combinations and sources of error increase dramatically. When one considers that some words in the dictionary used on this project are represented in Palantype by up to 9 chords, it becomes clear why the methods for identifying erroneous chords can be complex. Fig. 6C shows the possible roles that each chord can play in the act of reconstituting word boundaries.(Extra chords and deleted chords are not considered in fig. 6C).

C1	C2	C3	C4	C5	C6	C7	C8	...
IC	2LS	3LS	4LS	5LS	6LS	7LS	8LS	...
	IC	2LS	3LS	4LS	5LS	6LS	7LS	...
		IC	2LS	3LS	4LS	5LS	6LS	...
			IC	2LS	3LS	4LS	5LS	...
				IC	2LS	3LS	4LS	...
					IC	2LS	3LS	...
						IC	2LS	...
							IC	...

IC = INITIAL CHORD

2LS = SECOND LEVEL SUCCESSOR

3LS = THIRD LEVEL SUCCESSOR

ETC.

Fig. 6C Possible roles for each chord

6.2 IDENTIFYING ERRONEOUS CHORDS

Once the presence of an error has been detected, the actions taken to identify the source of error depend on the point in the transcription program at which the presence of the error became apparent, i.e. the actions taken are governed by the characteristics (e.g. initial, medial, final) and number of chords processed since the last English word was output.Note that the transcription program can be constructing several English words at one time without outputting any of them (see chapter 3). The points in the transcription program at which it becomes apparent that an error is present in the input string, have been classified and labelled. These labels (E1 to E8), the conditions which cause these points to be reached, and the corresponding actions taken to identify the source of error are all listed in appendix B. However, in order to get a flavour for the error identification process, error points E3 and E5 shall now be described.

Error point E3 is arrived at when the following conditions are satisfied.

1. Only two chords of the input string have been processed since the last English word was output.

2. The latter of these chords is not a successor in the tree of the former.

3. The former is an initial chord and is a word in its own right.

4. The latter chord is not an initial chord.

For example, this point would be arrived at if the operator was intending to type " C/AN T+/I " (candy), but typed " C/AN +/I ".

Actions taken to identify the source of error at error point E3 are as follows. As either or both of the two chords could be erroneous, the correction procedure first assumes one of them to be in error and attempts to correct it. If this is not successful, the other chord is assumed to be in error and again attempts are made to correct it. If this is also unsuccessful, then special correction techniques are required since both chords are in error or there is an added or deleted chord situation. At error point E3, since the first chord is not only an initial chord, but is a word in its own right, the latter of the two chords is more likely to be in error . Therefore,, in the correction procedure described in the next section, it is the first to be assumed erroneous.

Error point E5 is arrived at when the following conditions are satisfied.

1. More than two chords have been processed since the last English word was output.

2. The latest chord to be scanned is not a successor to the previous chord in the input string.

3. There has been no chord processed since the last English word was output which has a pointer to an English word.

For example, this point would be arrived at if the operator was intending to type " S/I C+N/IF C/NS " (significance), but typed " S/I C+N/IF C/N ".

Actions taken to identify the source of error at error point E5 are as follows. Any of the chords processed since the last English word was output could be erroneous. However, since the latest chord to be scanned was the first to "break the chain", it is the first to be assumed erroneous. If it cannot be satisfactorily corrected, then the error correction procedure scans the chords until it finds one that is the start of a word, and then assumes the chord immediately preceding it to be erroneous. If this does not produce a satisfactory correction, the correction procedure looks for another initial chord, other than the one that immediately follows the last English word. If there are no more initial chords then special correction techniques are required since there are multiple errors or added or deleted chords.

6.3 A STRATEGY FOR ERROR CORRECTION

Once the presence of an error in the input string has been detected, the identification process described in the previous section is combined with a correction procedure in a routine to automatically correct errors. The first correction routine described uses knowledge about the chords processed since the last English word was output up to the chord at which it became apparent that an error was present in the input string. For want of a better phrase, this was called the 'History' routine. The second correction routine described not only uses historical knowledge, but also looks ahead in the input string to obtain more information to help it correct erroneous chords. This routine was called the 'History and Future' routine.

6.3.1 The history routine

The history correction routine functions in the following manner.

1. It assumes one chord to be erroneous (depending on which error point it has been invoked from, as explained in the previous section) and submits the lists in which the erroneous chord was supposed to find a match to the error metric. The error metric produces two lists of chords, having a 'key difference' value of 1 and 2 respectively. The 'weighting' routine is then invoked to order the lists so that the chord most likely intended by the operator is at the top of the lists.

2. It then substitutes each chord in turn for the erroneous chord in its position in the input string (using first the list of chords with a key difference of one, and if necessary the list of chords with a key difference of two).

3. That part of the input string that was currently involved in building words when the error became apparent is then retranscribed with a substituted chord in place of the erroneous chord.

4. If the transcription is successful, i.e. English words are produced without detecting any more errors, then it is assumed that the substituted chord was the correct one. If the transcription is unsuccessful, then the process is repeated from 2 above, until all the chords in the ordered lists have been tried.

5. If this still does not produce a successful transcription, then the process is repeated from 1 above, until all possible sources of error have been tried.

This correction routine implemented on the computer improved the quality of the output text significantly. See figures 6D and 6E. However, in some cases the erroneous chord was not successfully identified, and in other cases even when successfully identified was not corrected to the chord the operator intended. In many cases the reason for this was because the correction routine could only use the string of chords up to the one at which the error became apparent, to test whether it had identified and corrected an erroneous chord successfully. If the erroneous chord is the latest one to be received, then this string is clearly inadequate as the correction routine is unable to see how the substituted chord would 'fair' with chords which came after it. For example, consider that the transcription program has received the following chords NOR MLI PFRO T+EU , when it becomes apparent that there is an error present in the input string. NOR is an initial chord, MLI is a second level successor to NOR and has a pointer to the English word "normally". PFRO is not a third level successor to NOR & MLI, but is an initial chord (i.e the start of the next word). T+EU is not a second level successor to PFRO, and PFRO is not a word in its own right. There is no other initial chord between NOR and PFRO, and so the correction routine assumes T+EU to be erroneous. Attempts at correcting T+EU prove unsuccessful since it is very far in terms of 'key difference' from any of the second level successors to PFRO . The correction routine now assumes PFRO to be erroneous and finds that there is another initial chord only one key difference away, i.e. FRO. (substitution error; P for .). T+EU is a second level successor to FRO. and so the retranscription is successful. It appears as though the transcription is going to produce "normally fraudulent ....", but although the error has been successfully identified, it has not been successfully corrected. This is confirmed when the next chord in the input string is received, SS, and there is no word "frauduses". Had the history routine been able to use SS to test whether FRO. was a successful correction, it would have found that it wasn't, and indeed as can be seen in fig. 6F would have found that the intended chord was PROE (substitution error; F for E), and the intended text "normally produces ....".

6.3.2 The history and future routine

The history and future correction routine has the ability to look ahead in the input string to help it test whether a chord has been successfully corrected. It does this by keeping several chords behind the operator. The history and future routine functions in exactly the same way as the history routine except that when the correction routine is retranscribing the relevant part of the input string containing the substituted chord, it also transcribes several chords from the 'future'. This was a significant improvement on the history routine. See fig. 6F. More examples of output are given in appendix D.

6.3.3 Results

The effect of the correction routines on the quality of the output text varied significantly depending on the number of errors in the input, how closely together the errors occurred and how 'bad' they were. Overall, the 'History' correction routine corrected 34% of all errors and the 'History and Future' correction routine corrected 52% of all errors. Fig.6D shows the output from a transcription program incorporating no error correction whatsoever. Fig.6E shows the same output but after the preprocessing corrections of Chapter 5 and the 'History' correction routine have been applied. Fig.6F shows the output after the preprocessing corrections and the 'History and Future' correction routine have been applied. Figures 6E and 6F are examples of output in which the Palantypist was performing rather better than average. In figures 6E and 6F successful corrections are underscored with an unbroken line where failures are underscored with a broken line.

T+R NEUL . Thank you Mr Shaw . PFOR I give my talk I want to say something about the SIS TEM we are using to-day . It uses a shorthand machine to take down the words that I am SEIN+ . But the HORT hand machine operator uses a peculiar code ANT+ the first slide is what a shorthand machine operator normally PFRO dew SS . We are using a computer to CHE.N +J that into the words you can see on those screens . Not all the words are spelt correct LI . But use the same technique as you use in lip reading . If you don't understand a word + carry on ANT+ try ANT+ make sense of the whole SEN tense . If you see a symbol it means that there is amiss TEU.C + ANT+ you ignore the words PFOR it . This SIS TEM is an IM proved version of the one used by Mr +JAC Ashley + MP . In the house of CO MONS . He has found the SIS TEM very beneficial ANT+ we are for TEU gnat to have Mr ash LIS Palantypist + miss beard + who is a T+REC tor of the Palantype or C+NAI SE. SHN ANT+ has been of tremendous help to us in developing this SIS TEM . She is here to-day Palantyping HFOIT say for you . Mr Ashley found the SIS TEM of great benefit + we hope that you will also find it help full in following T talk to-day . . . What I have been asked to lecture on is T V subtitling . TF+ Sup titling for the hearing impaired . This has become very important RI. sent LI in in gland because of the in TRO duck SHN of the TF+ SIS Thames which are called see FACS ANT+ oracle . I can now see my notes . . The see FACS oracle SIS TEM is a data transmission SIS TEM ANT+ it allows you to see data which a computer has PRO T+EU ST+ on your own T F+ set .

Fig.6D Uncorrected output

DR NEWL . Thank you Mr Shaw . Before I give my talk I want to say something about the system we are using to-day . It uses a shorthand machine to take down the words that I am saying . But the shorthand machine operator uses a peculiar code and the first slide is what a shorthand machine operator normally FRAW DEW SS . We are using a computer to change that into the words you can see on those screens . Not all the words are spelt correctly . But use the same technique as you use in lip reading . If you don't understand a word , carry on and try and make sense of the whole sentence . If you see a symbol it means that there is amiss steak , and you ignore the words before it . This system is an improved version of the one used by Mr yak Ashley , MP . In the House of Commons . He has found the system very beneficial and we are fortunates to have Mr Ashley's Palantypist , miss beard , who is a drake tor of the Palantype organisation and has been of tremendous help to us in developing this system . She is here to-day Palantyping what I say for you . Mr Ashley found the system of great benefit , we hope that you will also find it help full in following the talk to-day . . . What I have been asked to lecture on is T V subtitling . TV Subtitling for the hearing impaired . This has become very important recently in in gland because of the into duck sane of the TV system which are called see wax and oracle . I can now see my notes . . The see wax oracle system is a data transmission system and it allows you to see data which a computer has produced on your own T V set .

Fig.6E Output produced by a transcription program incorporating preprocessing corrections and the 'History' correction routine.

DR NEWL . Thank you Mr Shaw . Before I give my talk I want to say something about the system we are using to-day . It uses a shorthand machine to take down the words that I am saying . But the shorthand machine operator uses a peculiar code and the first slide is what a shorthand machine operator normally produces . We are using a computer to change that into the words you can see on those screens . Not all the words are spelt correctly . But use the same technique as you use in lip reading . If you don't understand a word , carry on and try and make sense of the whole sentence . If you see a symbol it means that there is amiss steak , and you ignore the words before it . This system is an improved version of the one used by Mr yak Ashley , MP . In the House of Commons . He has found the system very beneficial and we are fortunates to have Mr Ashley's Palantypist , miss beard , who is a drake tor of the Palantype organisation and has been of tremendous help to us in developing this system . She is here to-day Palantyping what I say for you . Mr Ashley found the system of great benefit , we hope that you will also find it help full in following the talk to-day . . . What I have been asked to lecture on is T V subtitling . TV Subtitling for the hearing impaired . This has become very important recently in in gland because of the introduction of the TV system which are called see wax and oracle . I can now see my notes . . The see wax oracle system is a data transmission system and it allows you to see data which a computer has produced on your own T V set .

Fig.6F Output produced by a transcription program incorporating preprocessing corrections and the 'History and Future' routine.

6.3.4 A better solution: discussion

The lessons learnt from attempting to correct single errors, and techniques elicited from the parser (described in the next chapter), have thrown light on what would probably have been a more general and better solution. There was not enough time to implement what follows.

As soon as an error became apparent, the correction routine could have been presented with a lattice of chords from which to select the single string intended by the operator. The word 'lattice' is used here in the same sense as by Miller (1974).

Consider the situation where the chords C1 to C9 have been transcribed and it has become apparent at C9 that there is an error in the input string.

C1 C2 C3 C4	C5 C6	C7 C8	C9

At this point the program has built a word from C1 to C4, C5 and C6, but has been unable to find any combination of chords that result in C9 satisfactorily following C8. Assuming that the correction routine can look-ahead five chords (as with the 'History and future routine), the lattice of chords which could now be presented to it is shown in fig. 6G and can be interpreted as follows:-

1A1 is the first alternative to chord C1

2A1 is the second alternative to chord C1

....

1A2 is the first alternative to chord C2

....

2A3 is the second alternative to chord C3

...... and so on.

I1 C1	I2 C2	I3 C3	I4 C4	I5 C5	I6 C6	I7 C7	I8 C8	I9 C9
1A1	1A2	1A3	1A4	1A5	1A6	1A7	1A8	1A9
2A1		2A3	2A4	2A5	2A6		2A8	
3A1			3A4	3A5				
			4A4					

Fig.6G Lattice of chords

The chords I1 to I9 stand for Inserted chords and are, at the time of being presented to the correction routine, empty (to allow for the situation where the operator has neglected to press a chord, i.e. missing chords).

6.3.4.1 Finding paths through the lattice

Having eliminated most of the situations in which the transcription program is unable to produce English words for reasons other than operator error (see chapter 5), it is possible to set a threshold value for the correction routine as to how many chords in the input string can be assumed to be erroneous. If it was decided that the average operator's error rate was approximately 20% (1 chord in 5), then the threshold value would be 3 chords for the input string C1 to C14. This is an important factor to consider when it comes to finding successful paths through the lattice. A successful path is one which produces English words and terminates with C14 following C13 in some satisfactory way. With a threshold value of 3, any such path should contain 11 of the original 'C' chords (in the situation where no extra chords have been pressed). In the situation where C10, for example, was pressed and should not have been, then any path should contain 12 of the 14 'C' chords (since C10 is one of the 3 errors). Some examples of successful paths are as follows:-

C1 C2 C3 C4 C5 C6 C7 C8 1A9 C10 C11 C12 C13 C14

C1 I2 C2 C3 C5 C6 C7 C8 C9 C10 C11 C12 C13 C14

C1 C2 C3 C4 I5 C6 C7 C8 C9 C10 C11 C12 C13 C14

An example of an unsuccessful path is as follows:-

C1 I2 C3 1A4 2A5 C6 2A7 C8 1A9 1A10 1A11 I12 C13 C14

Since it contains only 6 of the original 'C' chords.

If several successful paths through the lattice are found, then the criteria for selecting the path which the operator intended will in the simple case be the following:-

1. which path contained most 'C's

2. which is the most probable path (based on operator error frequency)

However, a more objective decision could be made by submitting the successful paths to syntax analysis. The parser described in the next chapter has the attributes for both implementing the lattice search and performing syntax analysis.

Chapter 7

A GRAMMAR AND PARSER FOR THE OUTPUT FROM TRANSCRIPTION SYSTEMS

7.1 INTRODUCTION

It was previously mentioned that human transcribers are able to draw on many knowledge sources to enable them to transcribe Palantype code into English text. One of these sources, syntax, is concerned with the goal of producing a consistent, meaningful, grammatical structure for a sentence.

Of the problems associated with Palantype code to English text transcription, there were three in which it was thought that syntax analysis might prove useful; homophonic ambiguities, keying errors and the incorrect joining of chords in word boundary reconstitution. For this reason, a set of syntactic rules (which here shall loosely be called 'The Grammar'), were defined and used as the input to a powerful translator writing system* which performed the constituent analysis (parsing). The effect of syntax analysis is somewhat reduced in the present research since the definition of the rules do not take into account the many varieties of English (Quirk et al.) including non-grammatical uses of English. This chapter describes the Grammar and the Parser. The ways in which they have been used to help resolve the above mentioned problems are described in later chapters.

* SYNICS (Edmonds & Guest, 1978)

7.2 PARSING THE OUTPUT FROM PALANTYPE TRANSCRIPTION SYSTEMS

Parsing the output from Palantype transcription systems is a much more difficult problem than parsing 'book type' English text. Even if there were no errors introduced by the recording process, and the output was an exact orthographic representation of what was said, the task of producing a meaningful analysis would still be very difficult due to the high degree of error and ambiguity in speech itself. Although an operator can to some extent cope with inaccurate or hasty articulation, some ambiguity is inevitable because of the occurrence of homophones (e.g. wait/weight, inn/in, etc.) and the fact that Palantype is a phonetic shorthand means that for example, both "wait" and "weight" are represented in Palantype by the same chord. On top of these problems is the fact that operators make keying mistakes which sometimes have the result that the transcription program is unable to produce English words from the erroneous part of the input string.

The implication of all this for a parser of Palantype output is that, at certain points of the string being parsed, there will be lists of alternatives instead of single words, and some structures which are not English words at all. See fig. 7A. When the parser scans the 'next word' in the input string, it must be able to deal with a list of possible words and also be prepared to cope with the possibility that the right word is not included in that list. Due to the fact that some errors result in words (or chords) missing from the input, or words(or chords) added to the input, the parser must be able to deal with gaps in the input and extra input.

```
Palantype
Input string  SO   TH   +FEIS   IN   HFICH   A   LIM   TT   AM   NT+

Intended
Output        So   the  ways in which a   limited amount .........

String                                    limb teet
submitted     So                          limb tied
to parser     Sow the weighs in which a limb ted AM   NT+ ...
              Sew      ways               limited
```

Fig. 7A   Input for a Parser

Another goal of a parser of Palantype output must be to predict words or syntactic categories to help correct errors and fill gaps in the input string. A very simple example follows; if the transcription program had produced the string "the PRE TI girl", where PRE was an erroneous chord, the parser could predict that the word between "the" and "girl" was an adjective (or noun in compound noun phrases). The correction procedure could then focus its search on chords which transcribed into adjectives or nouns.

7.2.1 Construction tasks

The construction of a parser for Palantype output, like a parser for speech, is a complicated affair. Therefore, it was broken up into the following tasks.

1. The specification of a base syntactic component

2. Modification to the base to allow for structures in the input string that were not English words

3. Modification to the parser to predict syntactic categories

4. Modification to the base to allow for gaps, extra words and wrong words in the input string

5. Modification to the parser to deal with lists of possible words at each point in the input string

As will be seen shortly, there was not enough time to complete all the tasks. Tasks 1 and 2 were developed to a stage where they could be used in the three problem areas mentioned above. Task 3 was designed but not used in the problem areas. Tasks 4 and 5 were started but did not reach a usable state.

In most systems which work with natural language the purpose of the parser is to provide a representation of the syntactic units of the input and their relationship to each other. This representation is frequently a 'deep structure' tree as in fig. 7B, which may then undergo semantic analysis or interpretation (Chomsky,1965).

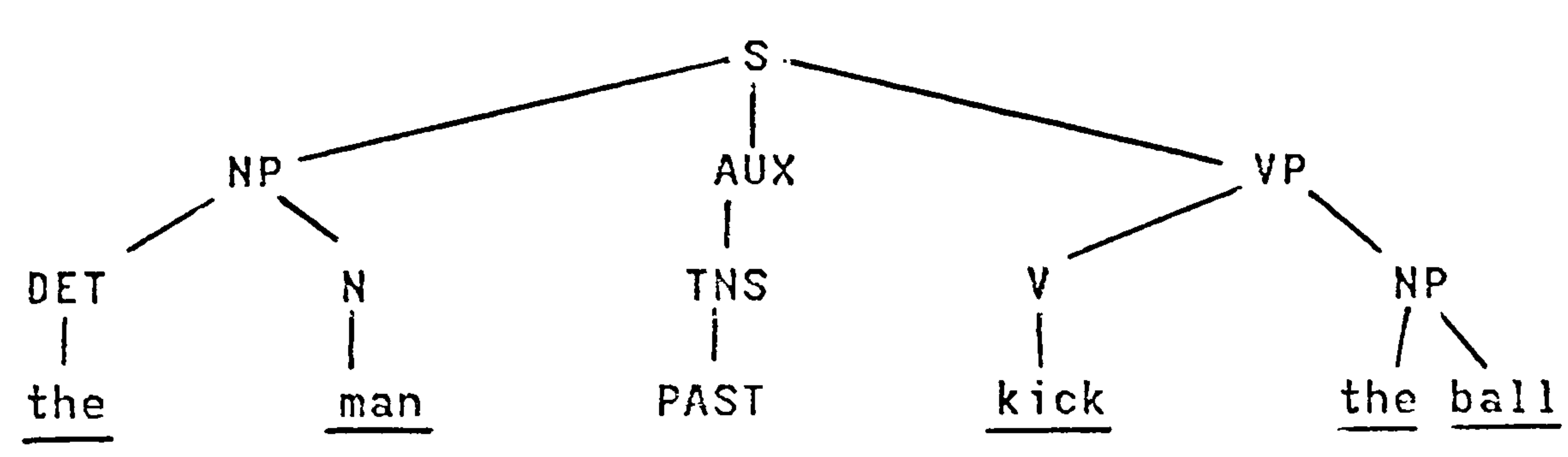


Fig. 7B Deep Structure of a Sentence

The present research has been concerned with the creation of a self contained syntactic structure as an aid to selecting a syntactically well-formed sequence of words from the many sequences of words which are possible in the input string. No attempt has been made at semantic analysis or interpretation.

7.3 SPECIFICATION OF THE BASE SYNTACTIC COMPONENT

The syntactic rules are defined in terms of recursive transition networks (Woods,1970), and based on the grammatical classifications of Quirk et al, 1972.

A basic transition network is a directed graph with labelled states and arcs, a distinguished state called the start state and a distinguished set of states called final states. When the labels on the arcs are allowed to be state names (non terminals) as well as terminal symbols, the resulting graph is the recursive transition network.

The structure of the grammar shall now be described in outline so that the reader may gain a fuller understanding of the parsing strategies described in later chapters.

Sentences are either simple (containing just one clause) or complex (containing more than one clause), Quirk et al,1972. See fig. 7C.

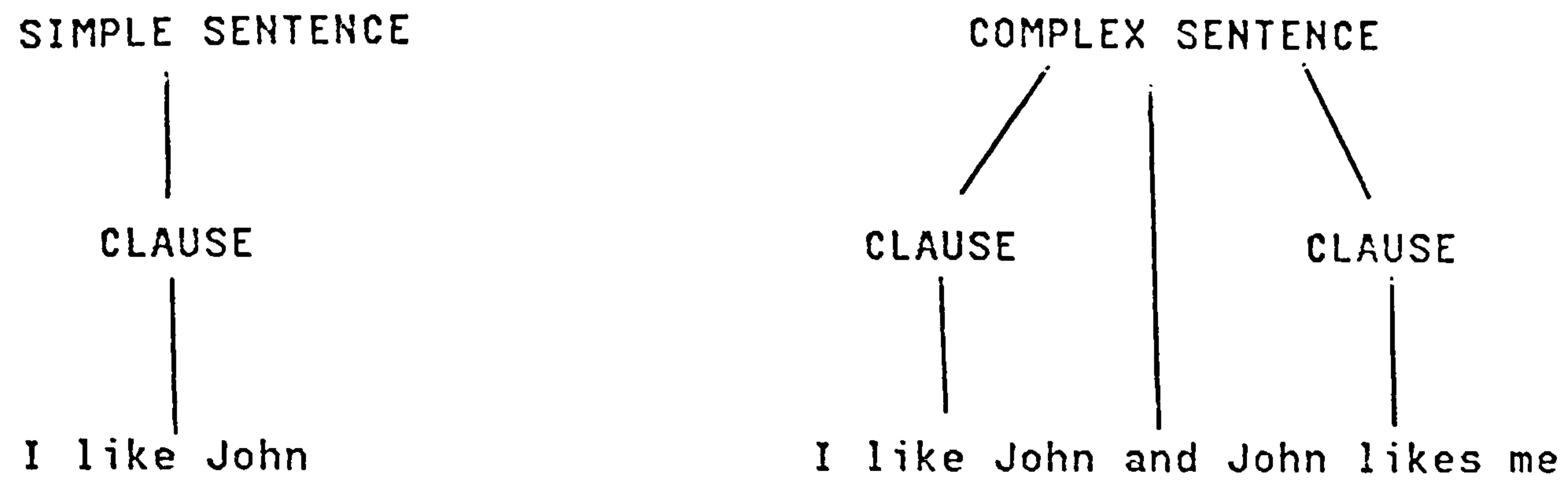


Fig. 7C Simple and Complex Sentences

The sentence is the start state of our grammar. A clause can be analysed into the following elements: subject(S), verbphrase(V), object(O), complement(C) and adverbial(A). The clause types as defined in Quirk et al, in their simple declarative form are given in fig. 7D.

Two types of object and two types of complement have been distinguished. There is a direct object, O(d), as with "the ball" in (4), and an indirect object, O(i), as with "me" in (7) of fig. 7D. There is a subject complement, C(s), as with "a nurse" in (1), and an object complement, C(o), as with "wrong" in (5) of fig. 7D.

(1) type	S	V(intens)	C(s)	
e.g.	Mary	is	a nurse	
(2) type	S	V(intens)	A(place)	
e.g.	Mary	is	in the house	
(3) type	S	V(intrans)		
e.g.	The child	was laughing		
(4) type	S	V(trans)	O(d)	
e.g.	Somebody	caught	the ball	
(5) type	S	V(com-trans)	O(d)	C(o)
e.g.	We	have proved	him	wrong
(6) type	S	V(com-trans)	O(d)	A(place)
e.g.	I	put	the plate	on the table
(7) type	S	V(ditrans)	O(i)	O(d)
e.g.	She	gives	me	expensive presents

Fig. 7D Examples of Clause types

Cutting across this sevenfold division are the main verb classes; INTRANSITIVE (followed by no obligatory element);INTENSIVE (followed by

C(s) or A(place);and TRANSITIVE (followed by O(d)). A transitive verb which permits an O(i) as well as an O(d) is known as DITRANSITIVE, and those which take an object complement as in (5) and (6) are referred to as COMPLEX-TRANSITIVE. The seven types of clause can be expanded by the addition of various optional adverbials; the transition network for the clause is given in fig. 7E.

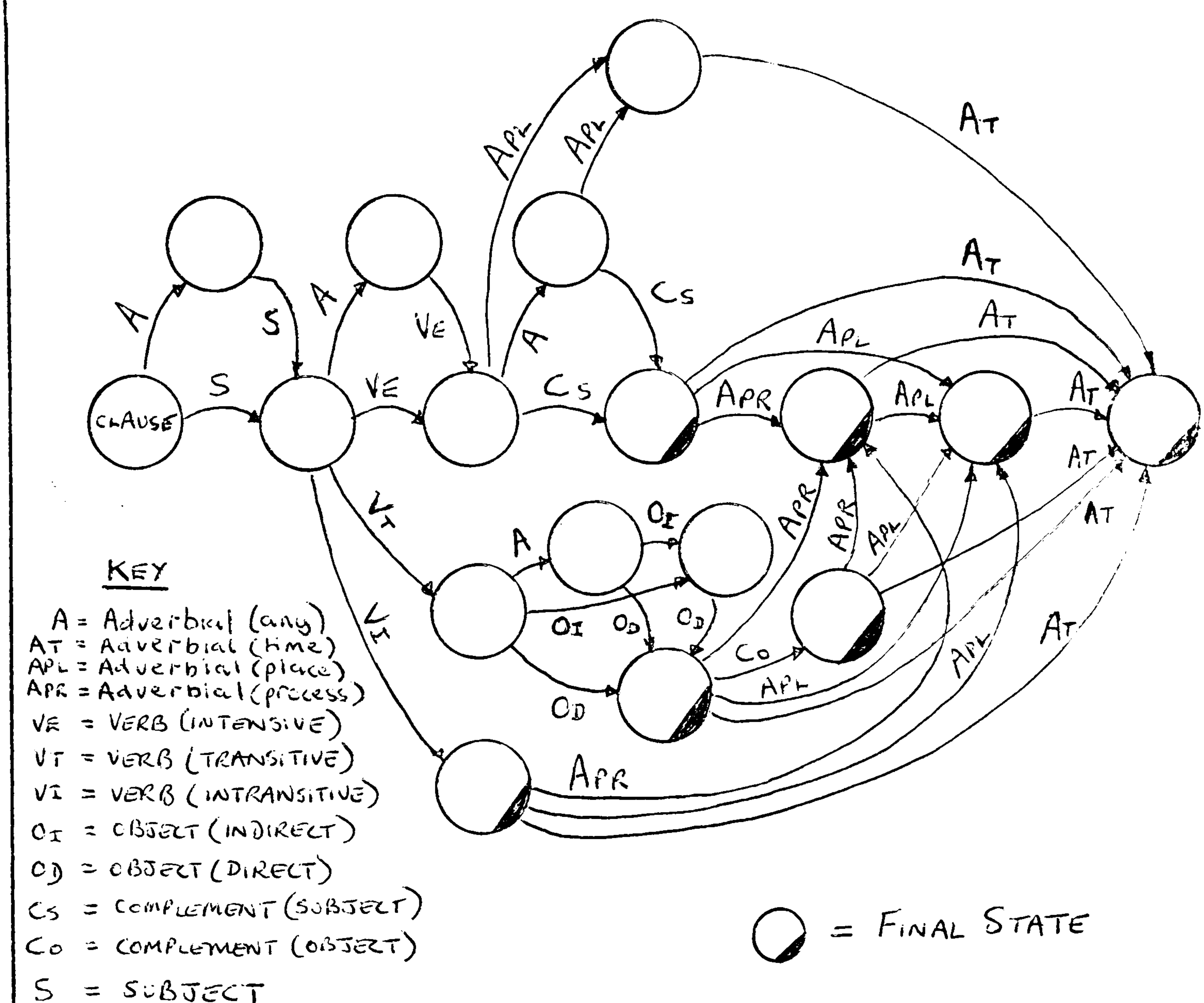


Fig. 7E Transition network for a Clause

7.3.1 Identifying the elements

The identification of the verbal element presents no problem, as this is the only function in which a finite verb phrase can occur, but the way in which the other elements can be realised are as follows:-

SUBJECT

A subject can be a nounphrase, as with "the child" in (3) of fig. 7D, or a clause with a nominal function, as with the "that clause" in fig. 7F. The subject occurs before the verbphrase in declarative clauses (as above), and immediately after the operator in interrogative clauses (explained later). It has number and person concord, where applicable, with the verbphrase.

OBJECT (direct or indirect)

An object, like a subject, can be a nounphrase, as with "the ball" in (4) of fig. 7D, or a clause with a nominal function. It normally follows the verbphrase. A direct object may occur without an indirect object, but not vice-versa. When both are present, the indirect object precedes the direct object.

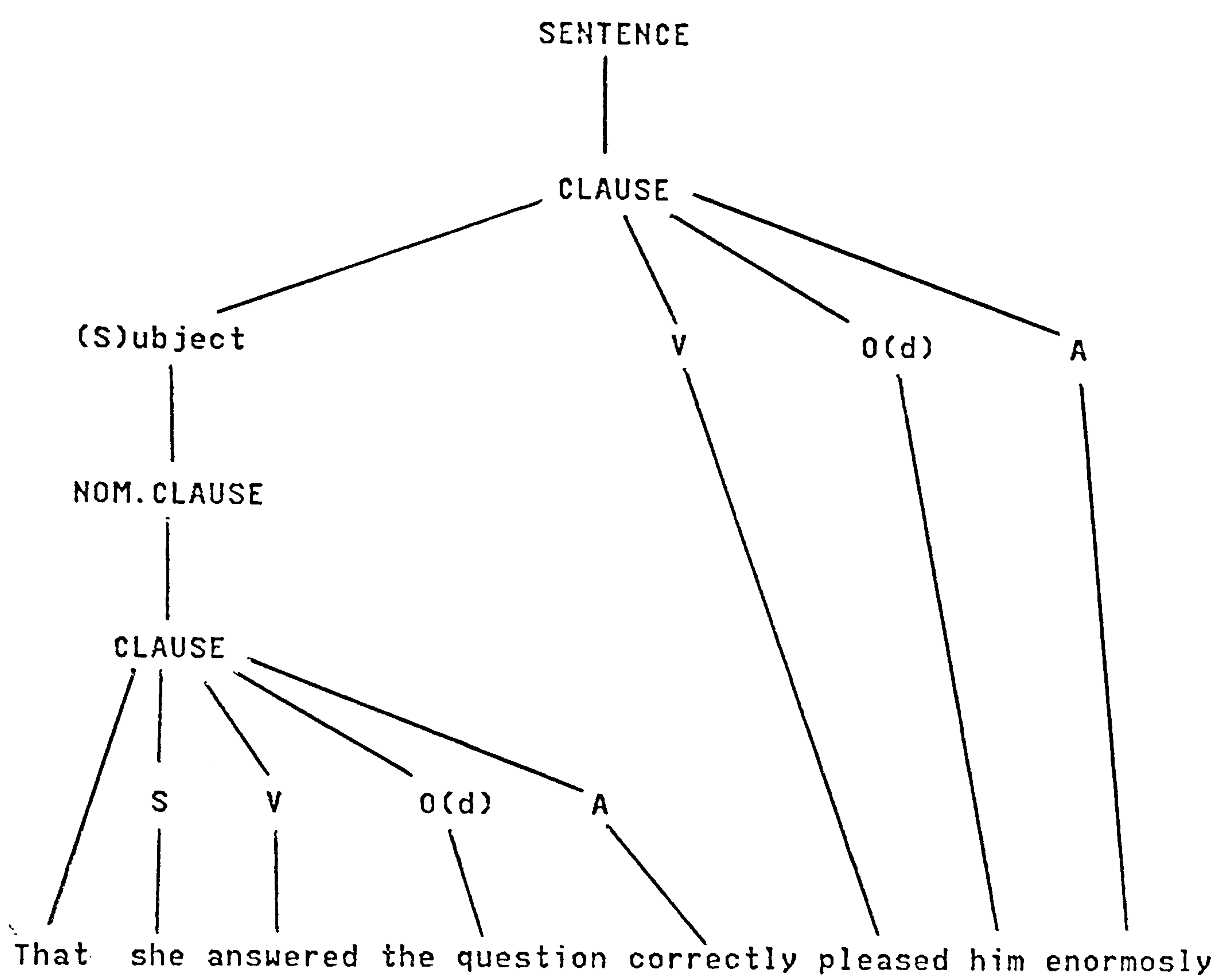


Fig. 7F 'THAT' Clause

<u>COMPLEMENT</u> (<u>subject</u> <u>or</u> <u>object</u>)

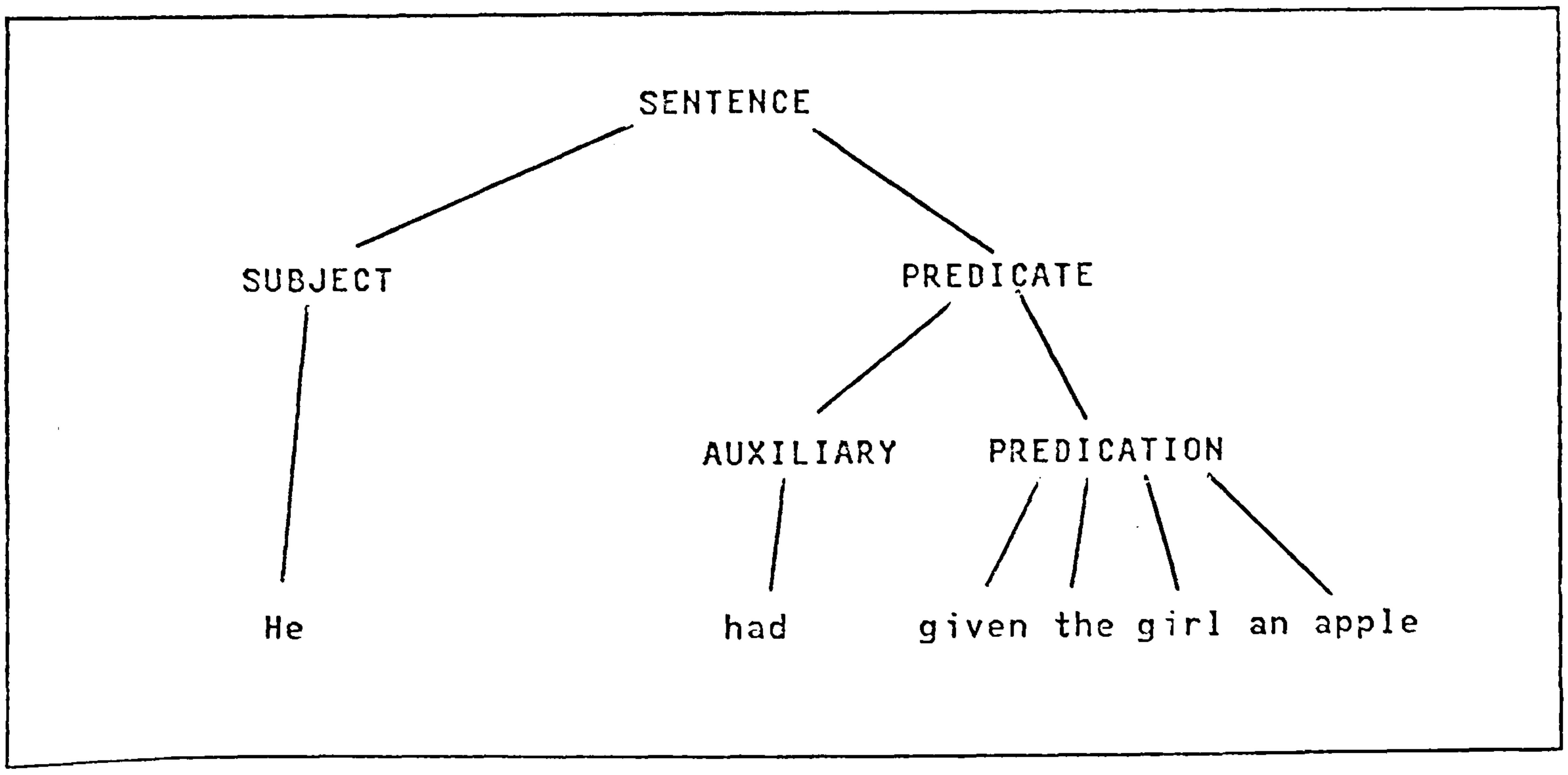
A complement can be a nounphrase, adjective phrase or a clause with a nominal function. It follows the subject, verbphrase, and (if one is present) the object.

ADVERBIAL

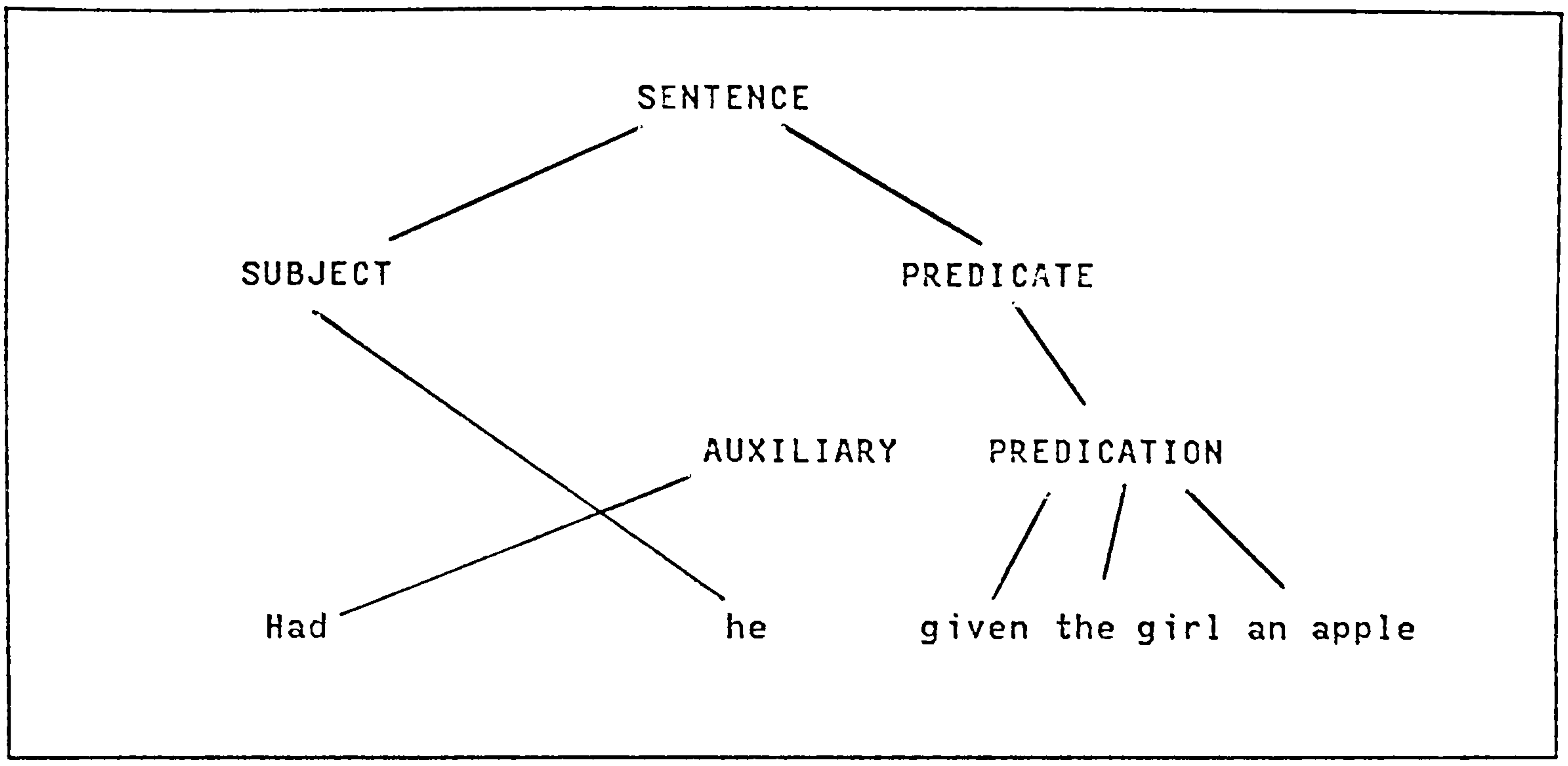
An adverbial can be an adverb as with "now", a prepositional phrase as with "at a large Polytechnic" in the sentence "The girl is now a student at a large Polytechnic", an adverbial clause or a nounphrase. The adverbial is generally mobile, i.e. capable of occurring in more than one position in the clause.

7.3.2 Interrogatives

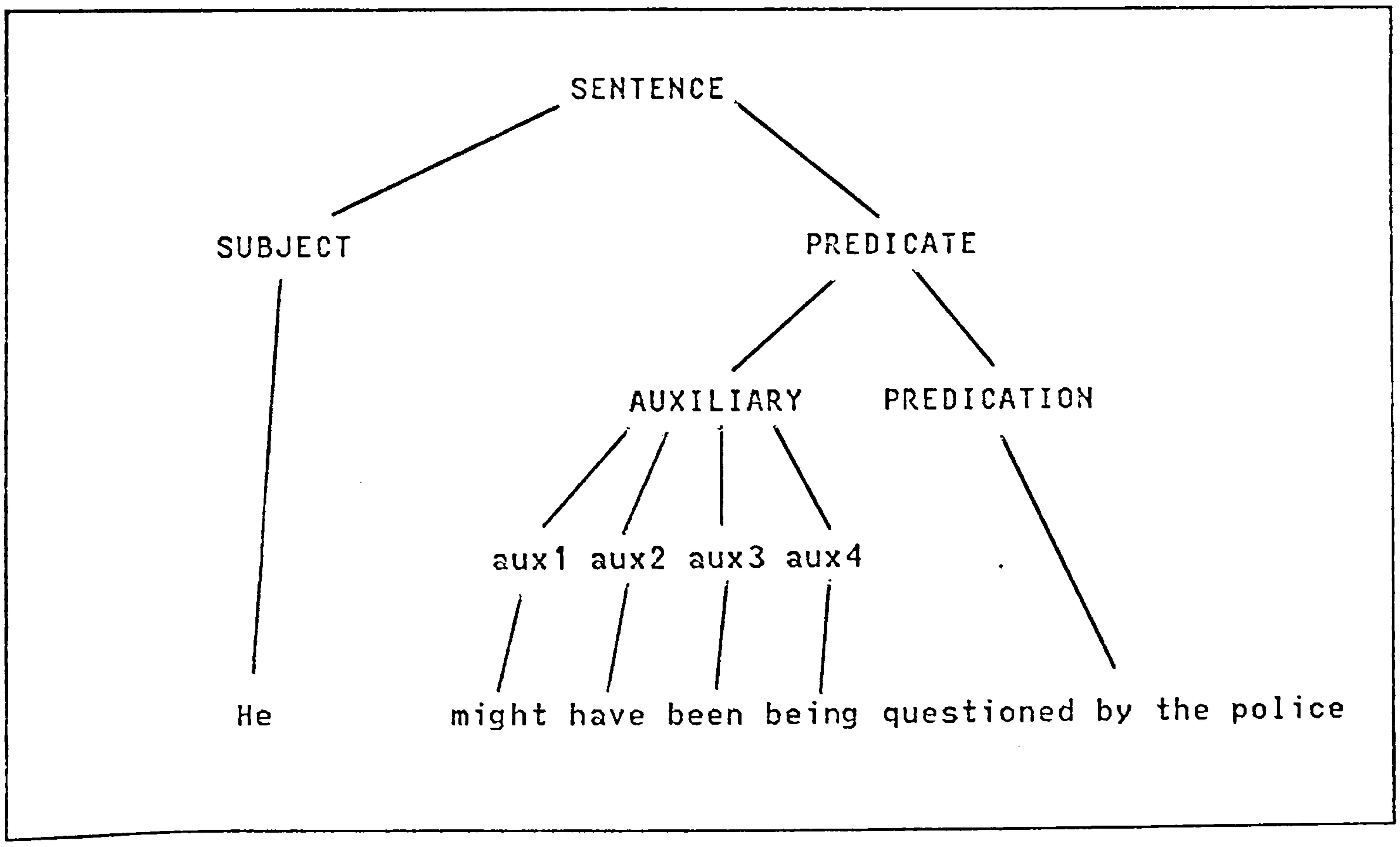
So far, only declarative sentences of one clause have been considered, which it has been said, can be analysed into the elements S, V, O, C and A. To understand how interrogatives (and negative) forms of sentences are formed, it is necessary to look at the division of a sentence in a different way. Consider that a sentence is divided into subject and predicate, and that the predicate is divided into an auxiliary and predication. In the following example, the auxiliary "had" is an operator, i.e. an auxiliary which occurs in an initial position in the finite verbphrase.



To obtain the interrogative (question) form of this sentence, the operator and subject exchange positions.



In sentences where the verbphrase has several auxiliaries, the first one is the operator.



The interrogative again is obtained by exchanging the subject and the operator, i.e "Might he have been being questioned by the police?" In sentences where the verbphrase doesn't have an auxiliary, the dummy operator "DO" can be introduced to obtain the question form. For example, the question form of "John searched the room" is "Did John search the room". The lexical verbs "BE" and "HAVE" can also be used as operators since they take inversion without the need to introduce the dummy operator "DO".For example, "Is she a pretty girl" and "Has she any money".

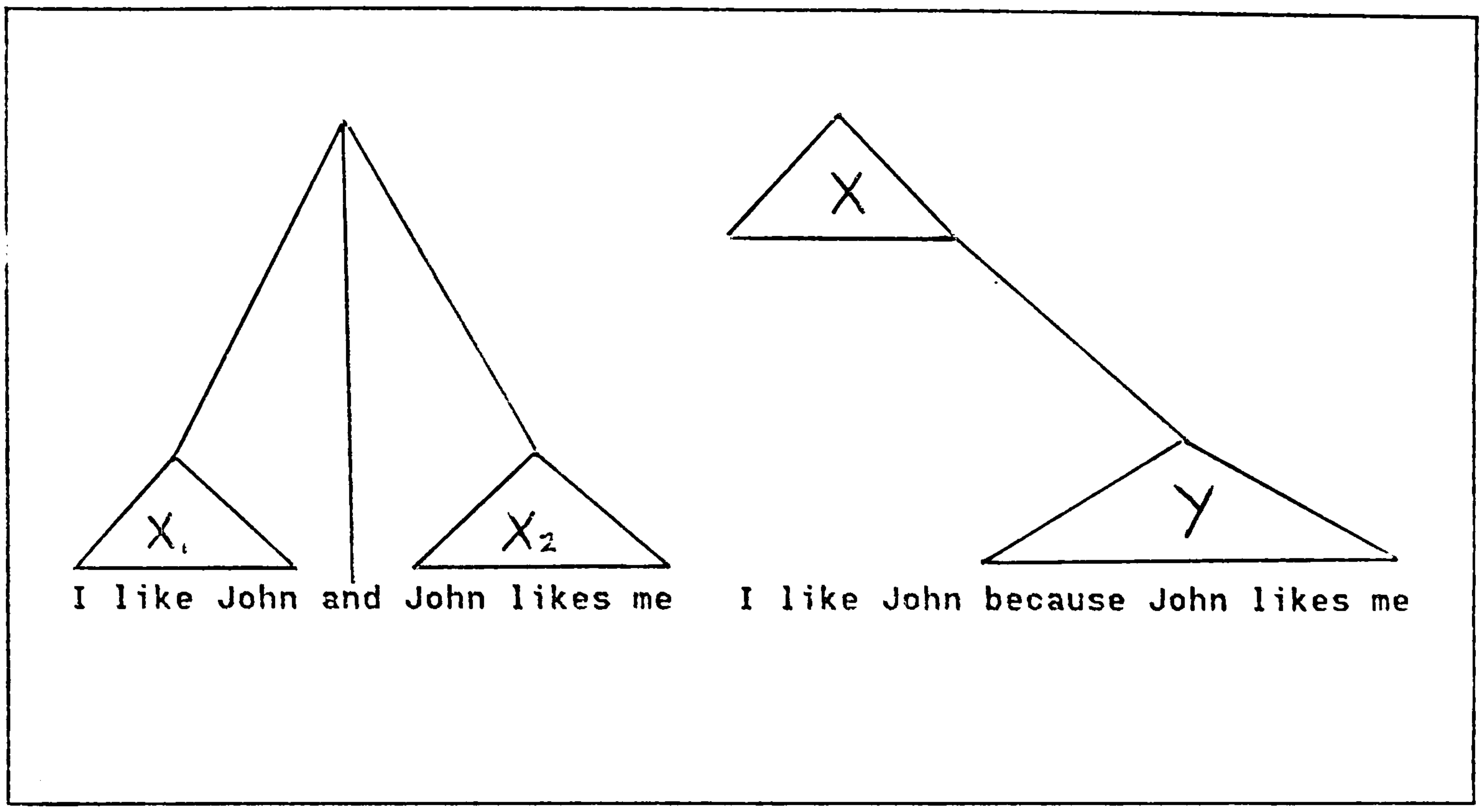
Another way of forming a question is to use what is referred to in Quirk et al, as a Q-word,i.e. which, when, why, where, how, what, whose, whom and who. It often happens that the Q-word is first in a sentence, followed by the operator, which is in turn followed by the subject and predication. For example, "When do they make him the chairman" and "What had he given her".

7.3.3 Clause connection

It has been shown that a simple sentence (whether declarative or interrogative) is a sentence which can be analysed as a single clause, in terms of subject, verb, complement, object and adverbial. The two main devices for linking clauses together to make up a complex sentence shall now be examined.

7.3.3.1 Coordination and subordination

The difference between coordination and subordination was shown diagrammatically in Quirk et al, as follows:-



In coordination there is a linking together of two or more elements of equivalent status and function, whereas subordination is a non-symmetrical relation, holding between two clauses X and Y in such a way that Y is a constituent or part of X. A second differnece is that a coordinate relationship may have more than two members, while only two clauses enter into the relationship of subordination: we may call them the subordinate clause (Y in the diagram) and the superordinate clause (X in the diagram).

7.3.3.2 Subordination

The device of subordination enables us to organise multiple clause structures since each subordinate clause may itself be superordinate to one or more other clauses. A further terminological distinction is that between an independent clause, i.e. one capable of constituting a simple sentence, and a dependent clause, i.e. one which is subordinate to another clause. Consider the following example,

Grammatical:	It is late. (independent)
Ungrammatical:	Because it is late. (dependent)
Grammatical:	I am going because it is late (independent with dependent)

There are various ways in which the subordination of one clause to another is indicated, but on the whole this indicator is signalled in the subordinate rather than superordinate clause. It is most often a subordinating conjunction as with "because","before","since",etc., but can be a "wh-element", the item "that", inversion or the absence of a finite verb form.

7.3.3.3 coordination

There are three conjunctions regarded as coordinators; "and", "or" and "but", although "for" is often classed as a coordinator. AND, OR and BUT allow ellipsis (omission when there is no doubt what word was intended), of the subject of the clause they introduce, but not so with FOR . Consider the following, I may see you tomorrow or I may phone today

could be written as

" I may see you tomorrow or may phone today "

" They may complain, but they haven't said anything yet "

could be written as

" They may complain, but haven't said anything yet "

BUT,

" He did not want it, for he was obstinate "

could not be written

" He did not want it, for was obstinate "

An indefinite number of clauses can be linked by "and" and "or". In cases where several clauses are linked by the same coordinator, it is usually omitted in all but the final instance. For example, " Attend all the lectures,(and) write full notes on them,(and) read the recommended books, and you'll be sure to pass the exams".

7.3.3.4 Testing the grammar

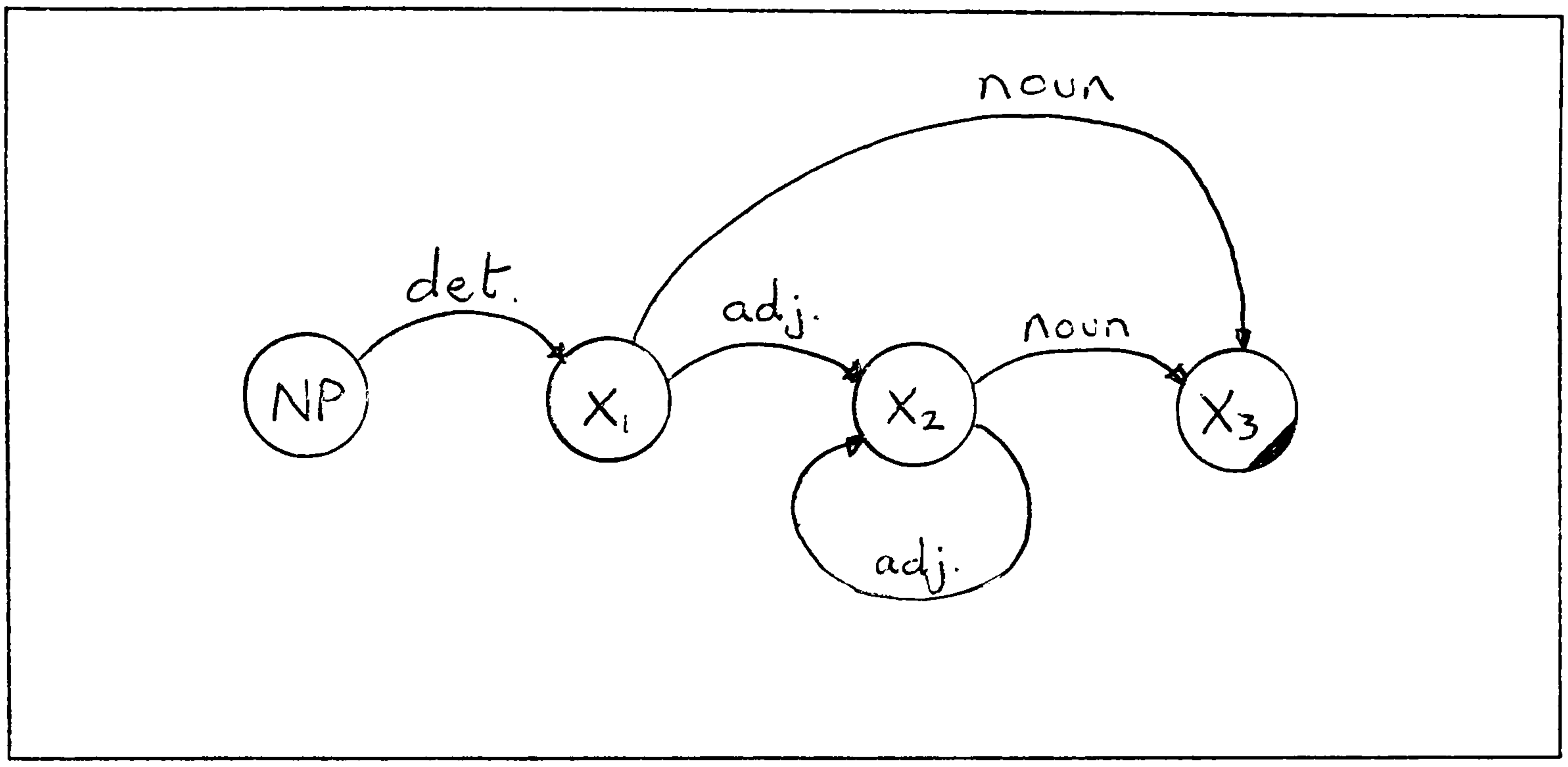
The grammar was tested with many input strings. Some examples are given in appendix C, with analyses. As the grammar was developed in a short period of time, the emphasis was to ensure that no valid strings were rejected. However, in certain cases strings which are ungrammatical produce a successful parse. This is a shortcoming of the grammar.

7.3.3.5 Summary

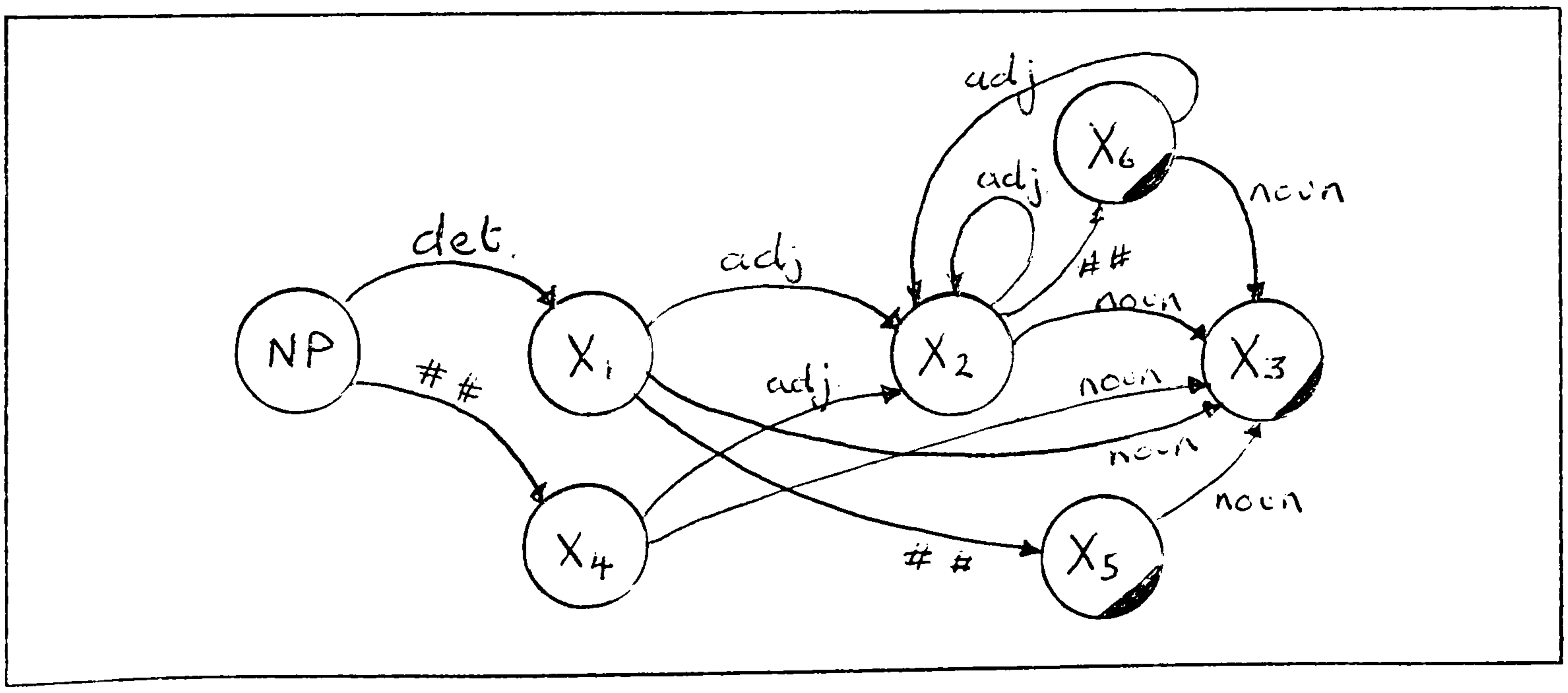
From this brief description it can be seen that the underlying structure of a grammar used in parsing the output from Palantype transcription systems is one where sentences are analysed into elements like subject verb, etc. All the elements are defined syntactically in terms of recursive transition networks, but no attempt has been made at this stage to do semantic analysis.

7.3.4 Allowing for non English word structures

Consider the transition network below and the phrase "the PRE TI girl".



It is clear that a parse of the string would fail at PRE . However, if the structure PRE TI was enclosed in delimeters e.g. #PRE TI#, and the transition network was modified as below, the parse would be successful.



This transition network would also allow the following

pretty girl
The pretty # #
The # #
girl
pretty #
The # # red # # ball

7.3.5 Predicting syntactic categories

Using the above transition network, arriving at state X4 would cause the delimited structure to be labelled "determiner". However, arriving at states X5 or X6 is a little more complicated. If the next symbol in the input string is neither a noun or an adjective, then the delimited structure is labelled "noun", otherwise it is labelled "adjective".

7.4 THE PARSER: SYNICS

The recursive transition networks that have been defined were expressed in a tabular form in order to 'drive' SYNICS (Edmonds & Guest, 1978), a powerful translator writing system which has been used to implement the syntax analysis in this application. SYNICS was chosen for several reasons;

1. It is an inherent feature of natural language that there will be ambiguous sentences which have several distinct analysis paths

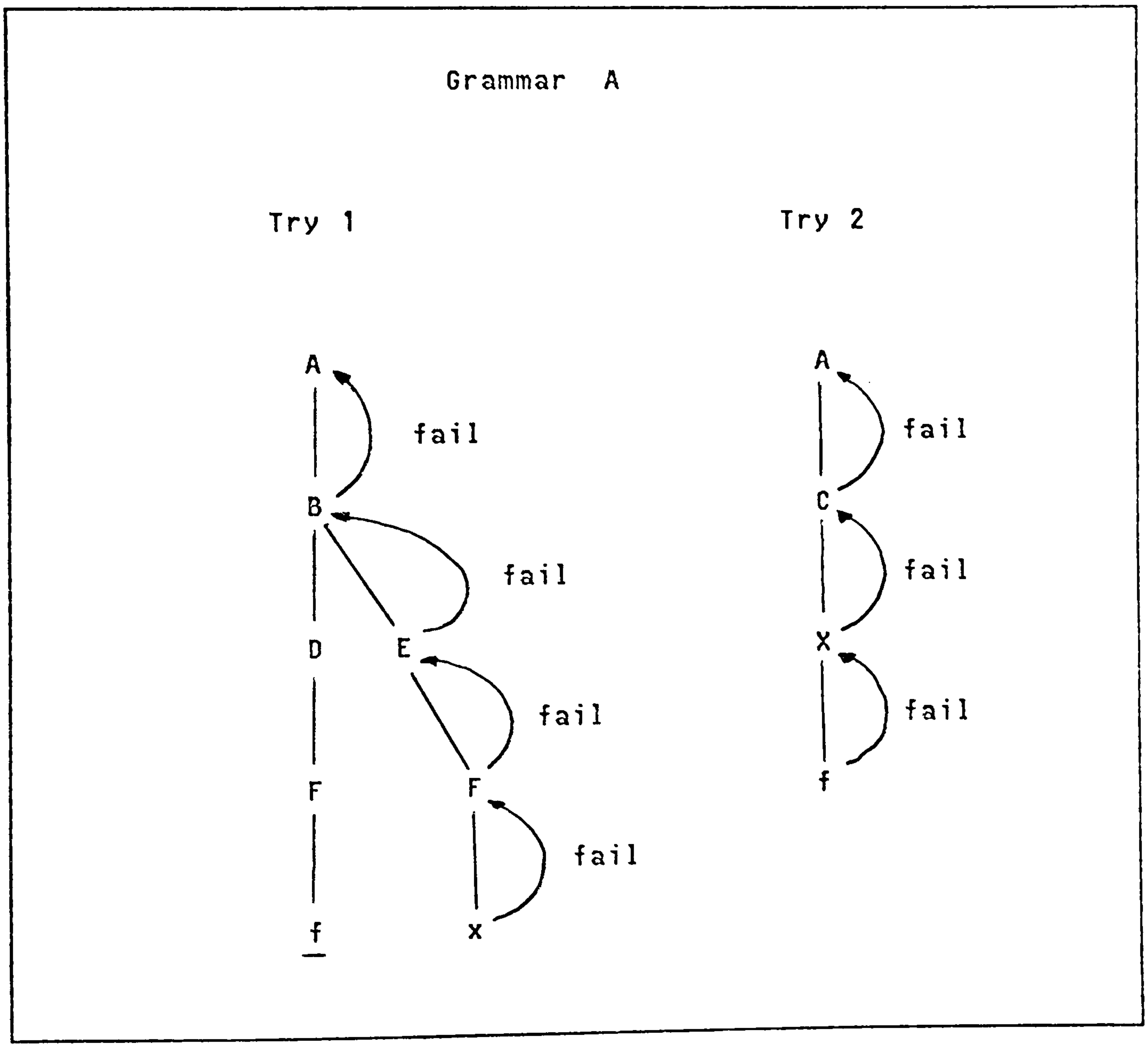
through the network. Woods,1970 states therefore that the transition network model is a fundamentally non-deterministic mechanism, and any parser must be capable of following any and all analysis paths for any given sentence.

SYNICS has certain 'special' features, and two which are particularly relevant in the present context are its ability to perform selective backup (Guest,1982), and its ability to find multiple analyses (if they exist) of the input string. In the first case, the user of SYNICS is allowed to select the paths through the Grammar on which "extra" backup is required, by placing a "+" instead of the normal OR symbol "/" in the definition rules. Normally, with the default backup algorithm, when the parser encounters a node which fails to find a match in the input string, it backs up to its parent and tries an alternative rule. However, when selective backup is implemented and a preceding alternative succeeds, the parser notes that if at a later stage, a fail occurs whilst the 'selected' node is still in the tree, then the alternatives to that node should be tried. The example below shows the difference between a grammar incorporating selective backup and one incorporating the default algorithm. Consider the two Grammars A and B, and a parse of the input string:- f x .

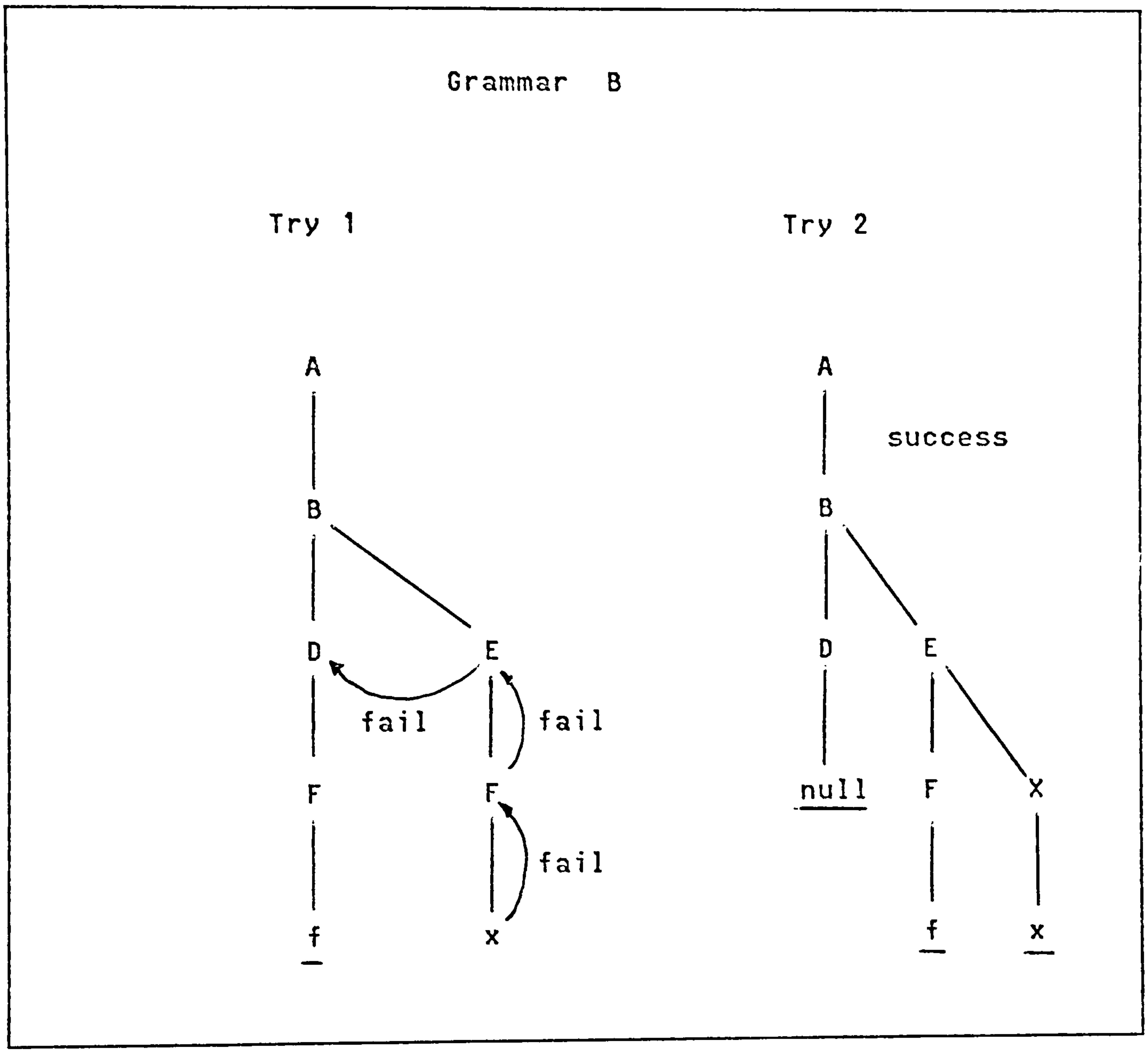
```
Grammar  A              Grammar  B

A=B/C/.                 A=B/C/.
B=DE/.                  B=DE/.
C=X/.                   C=X/.
 D=F//.                 D=F+/.
E=FX/.                  E=FX/.
F="f"/.                 F="f"/.
X="x"/.                 X="x"/.
```

A parse of the string "f x" would produce the following parse trees:-



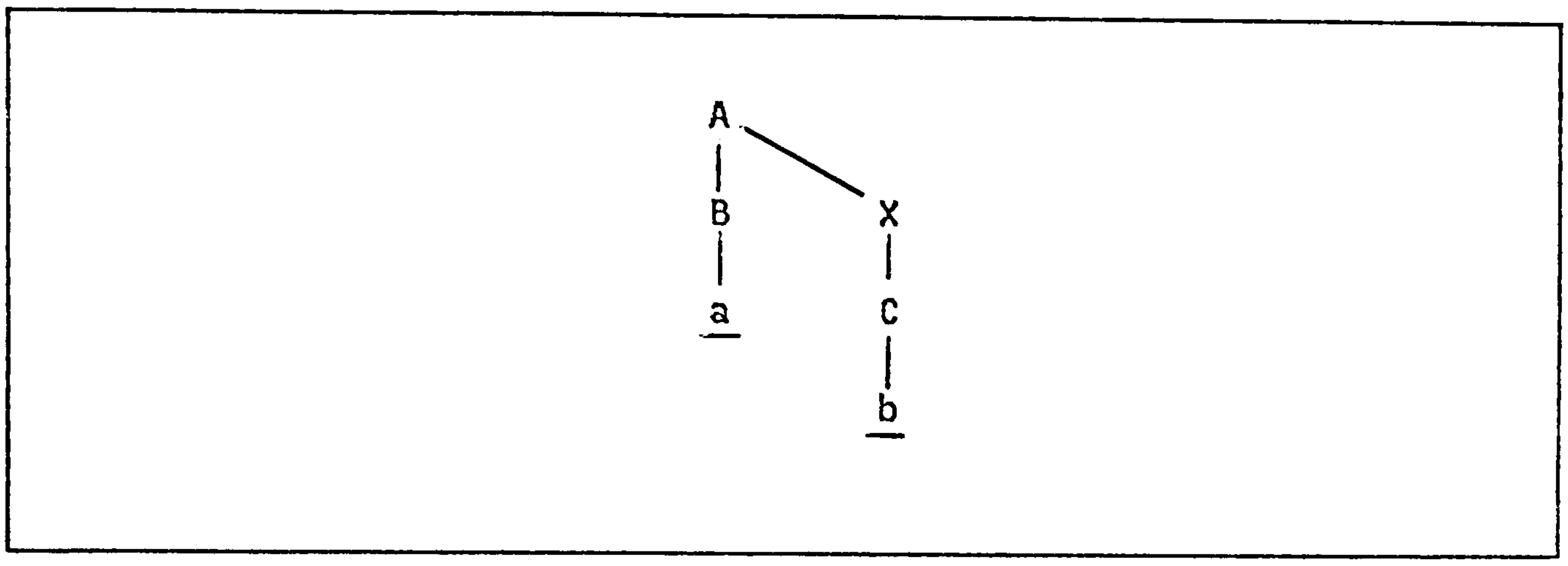
In the first try of Grammar A, the attempt to match the "f" is achieved by rule D. However, this leaves rule E unable to match the "x". In the second try rule C is unable to match the "f". In the tree of Grammar B, when the parser sees that rule E does not match the "x", it checks to see if any node in the tree has been marked as selective backup. Rule D has been so marked, and a successful parse is achieved in try 2.



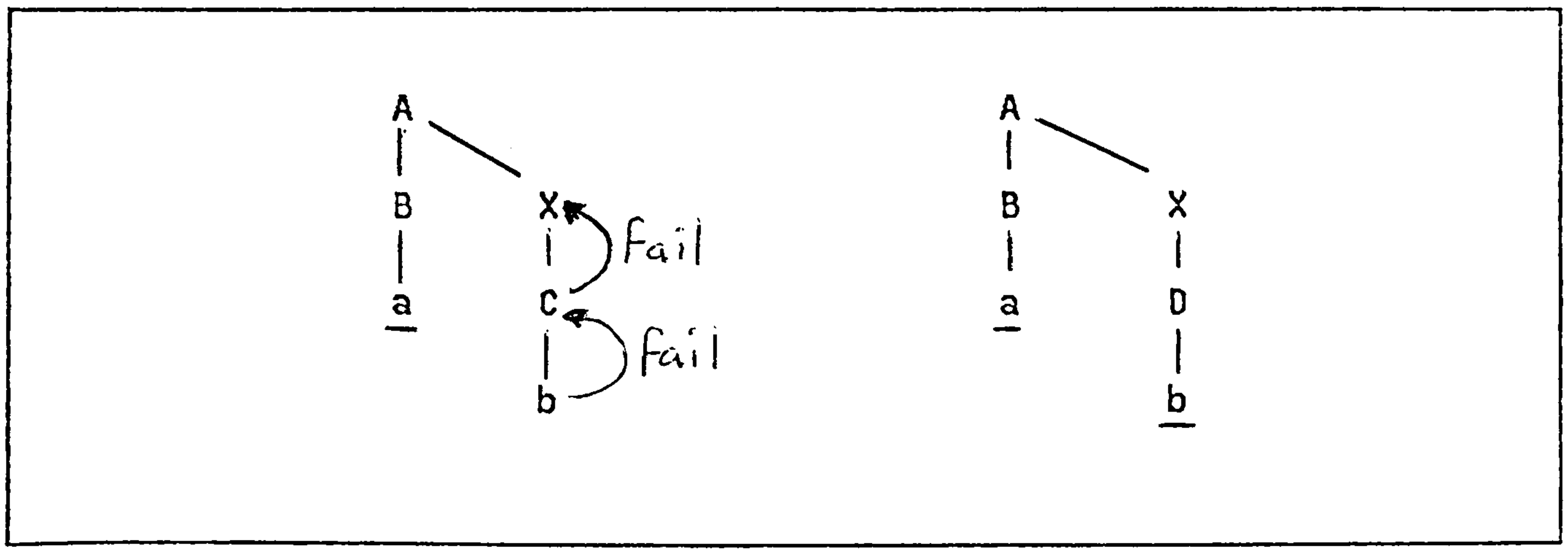
The ability of SYNICS to perform multiple analyses is achieved by issuing the command 'REDO' after the parser has produced a successful analysis of the input string. REDO causes the parser to treat the last node in the successful tree as a FAIL node. Consider the following example:-

```
A=BX/.
B="a"/.
C="b"/"c"/.
D="b"/.
X=C/D/.
```

A parse of the string "a b" would produce the following tree:-



Now if the REDO command was issued, it would cause the last node, C to fail. SYNICS would now try the alternative rule and produce the following trees:-



It is a combination of these two special features of SYNICS which enable the parser to follow any and all analysis paths for any given sentence.

2. The recursive transition network is equivalent to a push down store automaton, with the ability to suspend the processing of a constituent at a given level while using the same grammar to process an embedded constituent.

In the above example of Grammar 1 and 2, A was pushed on the stack while B was processed which in turn was pushed on the stack while D E were processed and so on.

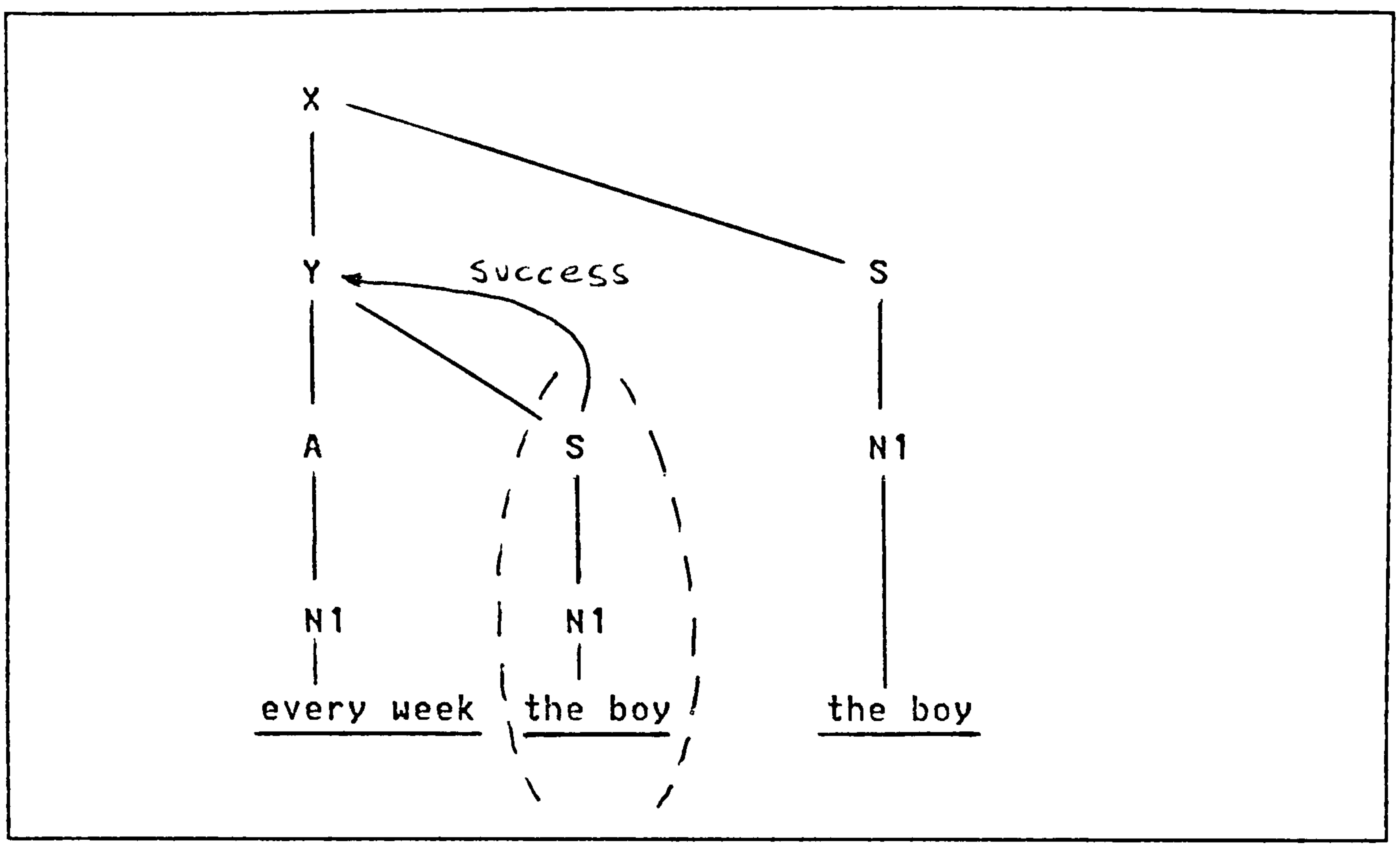
3. The augmented transition network adds to each arc of the transition network an arbitrary condition which must be satisfied in order for the arc to be followed.

SYNICS has an 'if then else' construct which facilitates the testing of any chosen condition. It also has a very powerful feature in the concept of 'interrupt nodes' which when encountered by the parser, transfer control over to a user supplied subroutine where list checking or complex semantic processing could be performed for example. Consider the following example:-

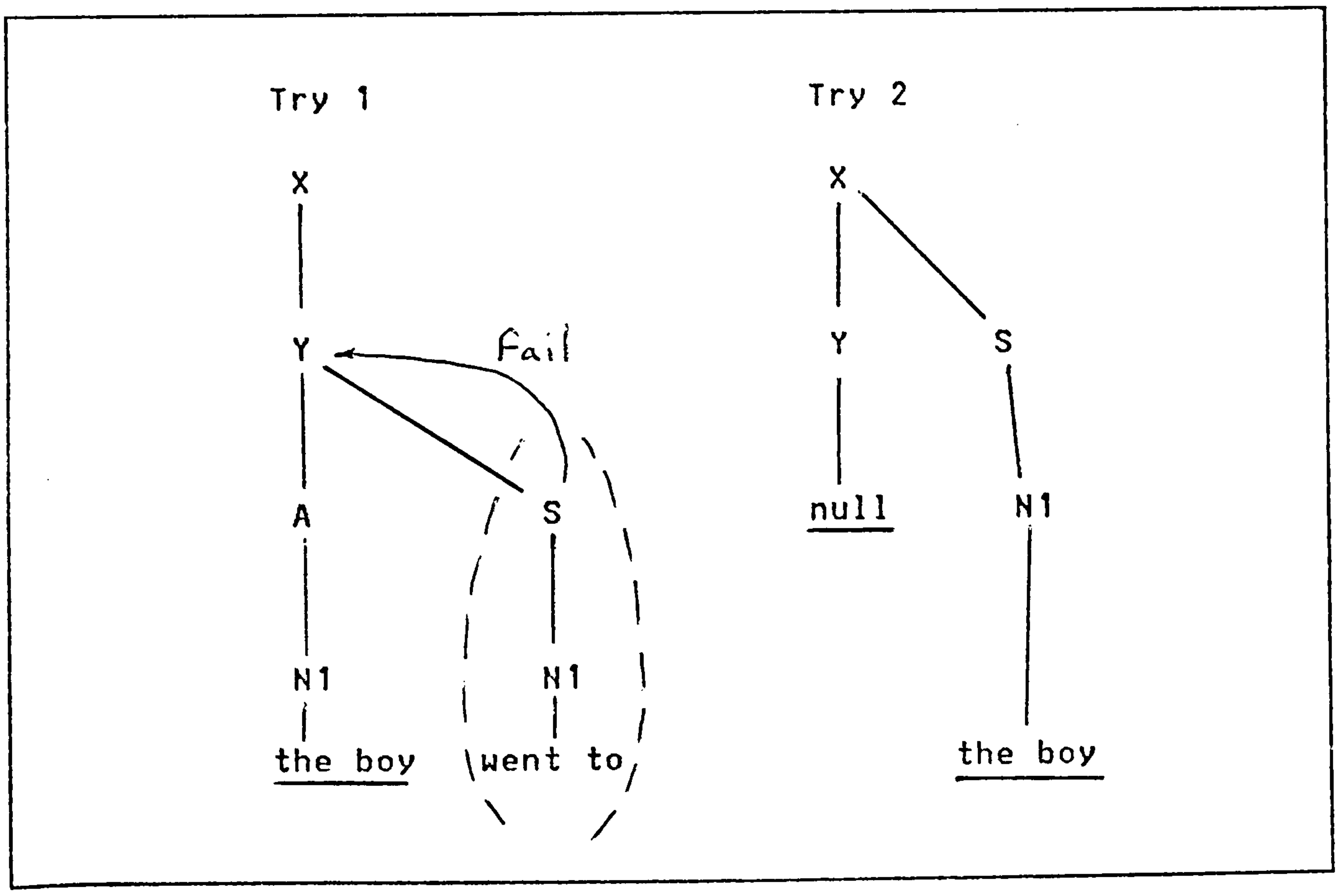
```
          X=YS[C1C2]/.
          Y=if AS then A[C1]/else /.
S subject =N1[C1]/.
A advbial =N1[C1]/.
N1 nounphrase ="every week"/"the boy"/.
```

In this example, the interrupt node is $1 and, when encountered by the parser, transfers control over to a subroutine which tests whether or not the current word in the input string is in a list of nouns. The subroutine returns a flag to say whether the node succeeded or not.

Rule Y shows a use of the 'if then else' construct. A parse of the string " every week the boy went to ......", would produce the following partial tree:-



However, a parse of the string "the boy went to ...." shows how the 'if then else' construct has been used to test for an optional adverbial before the subject of a sentence.



4. The augmented transition networks have a set of structure building actions to be executed if an arc is followed. This is fairly equivalent to the transformational grammar which has the ability to move fragments of the sentence structure around (so that their positions in the deep structure are different from those in the surface structure) and also copy and delete fragments of sentence structure. The actions on constituents depends on the context in which those constituents occur.

By using the semantic facilities in SYNICS one can move fragments of sentence structure around, for example consider the following grammar:-

G1*

```
S1=N1 V1 [L2"SUBJECT"C1]/
   N1 V2 [L2"OBJECT"C1]/.
V1=V3 N1 [L2"VERB"L1"OBJECT"]/.
 V2=A1 V3 "by" N1 [L "VERB"L1"SUBJECT"]/.
 V3="kicked"/.
 A1="was"/.
 N1="the" N2/.
 N2="ball"/"man"/.
```

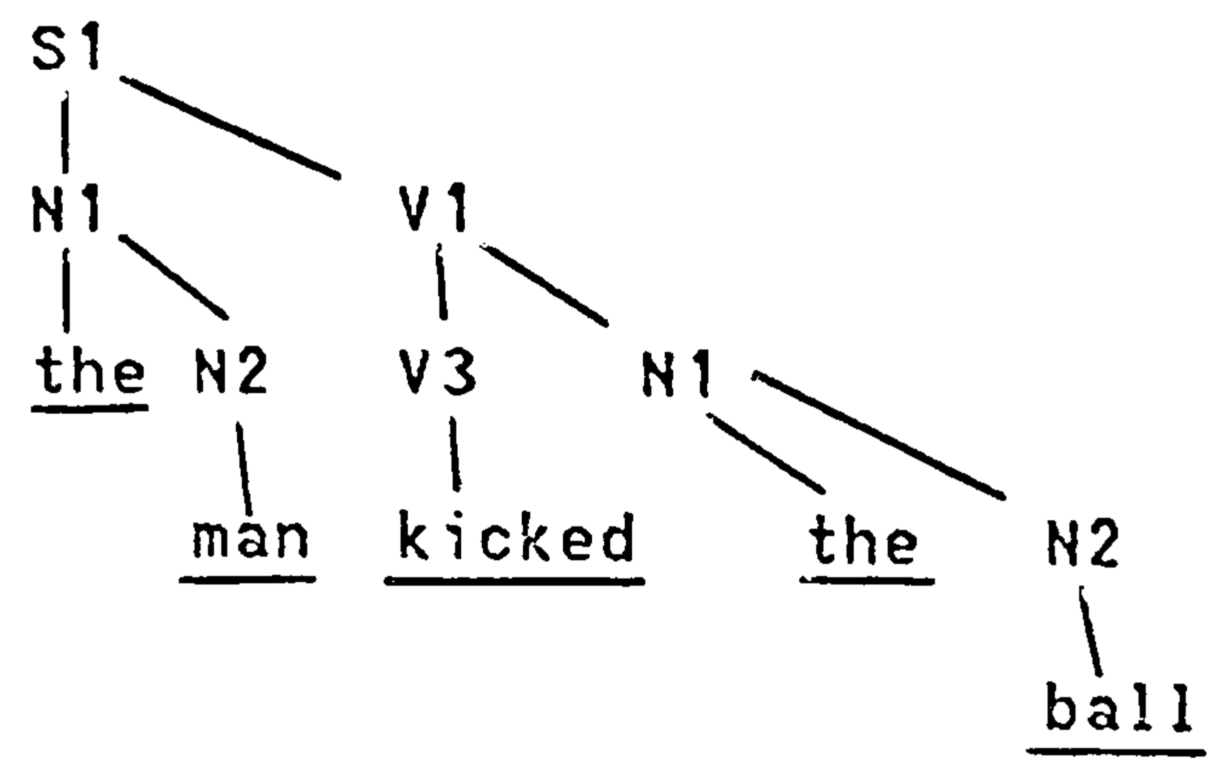
Although the two sentences "the man kicked the ball" and "the ball was kicked by the man" have the same deep structure, the above SYNICS statements allow this deep structure to be transformed into differently ordered surface structures, i.e. S V O and O V S. See fig. 7G.

G2* as G1 but

```
S1 = N1 V1 [L2 "SUBJECT" C1]/
     N1 V2 [C1 L2 "OBJECT"]/.
```

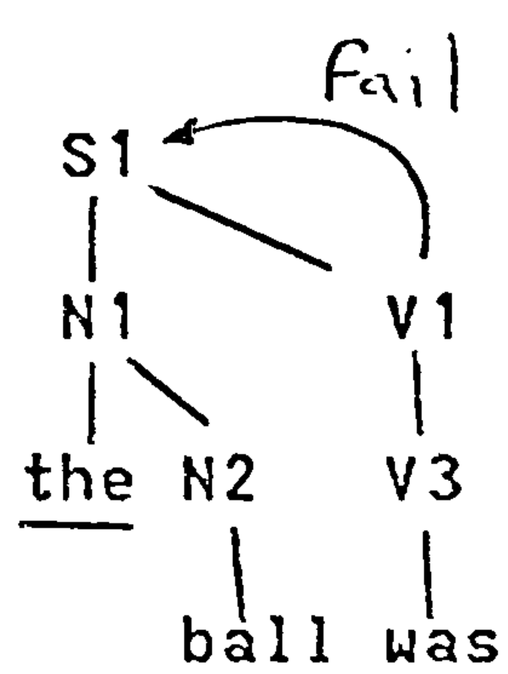
and

```
V2 = A1 V3 "by" N1 [L1 "SUBJECT" L4 "VERB"]/.
```

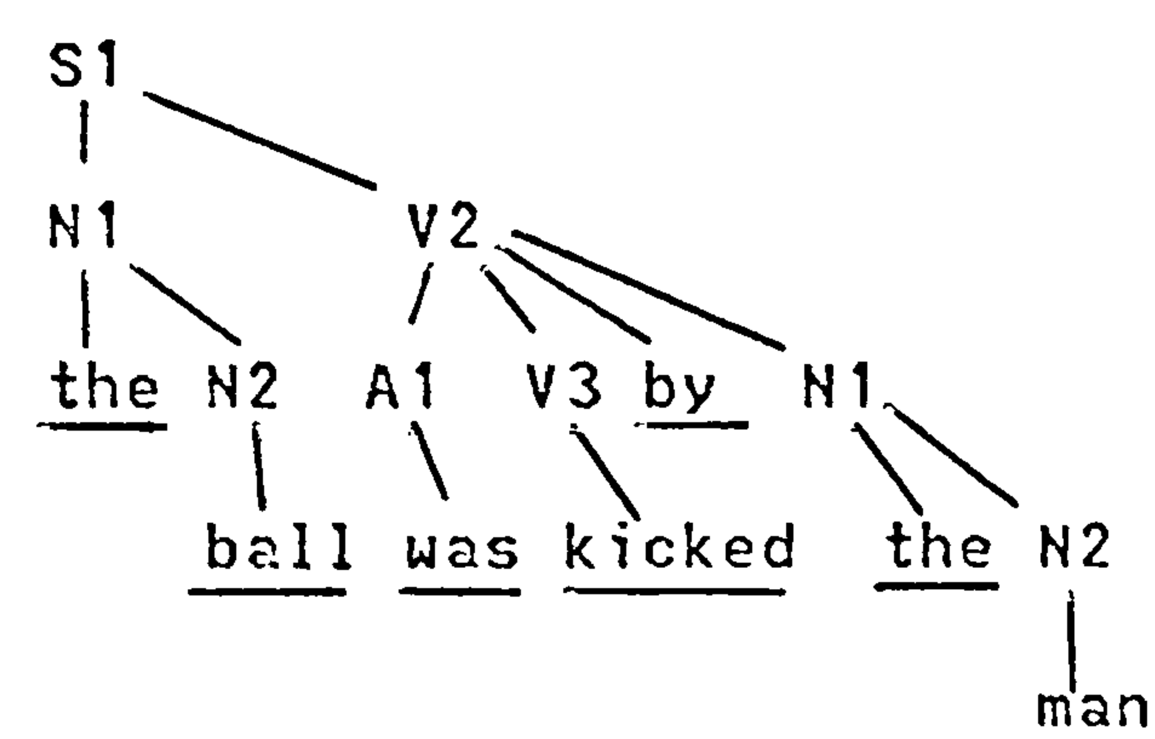


G1&G2
output: the man SUBJECT kicked VERB the ball OBJECT

Try 1



Try 2



G1output: the ball OBJECT kicked VERB the man SUBJECT
G2 output: the man SUBJECT kicked VERB the ball OBJECT

Fig.7G SYNICS output

Chapter 8

SYNTAX ANALYSIS IN THE CORRECTION OF ERRORS

8.1 INTRODUCTION

In the previous chapter a set of syntactic rules and a parser were described in outline. This chapter describes how the rules and parser were used to help correct operator errors.

A shortcoming of the correction procedures described in Chapter 6 was that sometimes the chord which produced the first successful re-transcription was not the chord which the operator originally intended. The correct chord was sometimes lower down the ordered list, or had a greater key difference value than the chord which produced the successful transcription.

An example to illustrate this is as follows. If the operator was attempting to type the chords for "the ten bases", which are /TH, T/EN, P+/E. and S/S, but typed /IS instead of S/S then the transcription program on scanning /IS would find that it was not second level sucessor to P+/E.. Also, P+/E. is not a word in its own right, nor is it a successor , in any tree, to the chord T/EN. At this point, the transcription program has detected the presence of an error in the input string, and so invokes a correction procedure. The correction procedure assumes that /IS is in error and should have had a direct match in the list of second level successors to P+/E. (Fig.8A). The error metric produces two lists of chords, each having a key difference value of 1 and 2 respectively from /IS, as shown in figure 8A.

Second level successors to P+/E.	Key difference of 1	Key difference of 2
P+/I (baby, etc.)	+/I	P+/IS
P+/IS (babies)	S/IS	CH/IS
P+J/ISH (babyish)	etc.	etc.
C/N (bacon)		
CR/I (bakery)		
CR/IS (bakeries		
C/IN (ba ing)		
S/L (basal)		
S/S (bases)		
S/I.G (bases)		
S/NS (basing)		
S/IS (basis)		
TH/S (bathes)		
etc.		

Fig. 8A Second level successors to P+/E.

The weighting routine inspects the lists of "key difference 1" chords and decides that the operator is more likely to make the deletion error of forgetting to press the first S in the chord S/S than to make the substitution error, an I for an S in the chord S/IS. Consequently the weighting routine places SIS above SS so that the "key difference 1" list now looks like this:

Key difference of 1

SIS

SS

etc.

The correction procedure substitutes the chord at the top of the "key difference 1" list for the chord which it has assumed to be erroneous, in its position in the input string. The input string is now as follows: TH, TEN, P+E. and SIS. A transcription is performed and the English words "the ten basis" is produced. As it is a successful transcription, i.e. it produces English words without detecting any more errors, the chord S/IS is accepted as the chord the operator originally intended. The chord which the operator actually intended, S/S, is lower down the key difference list and consequently does not get the opportunity of a transcription.

To try and overcome this problem the idea of accepting the first chord which produced a successful transcription was extended to include the following condition: the chord is only accepted if the words produced by the successful transcription form part of a phrase which when subjected to syntax analysis produces a successful parse.

It is clear then, if this condition was applied to the above example, "the ten basis" would not produce a successful parse since "basis" is a singular noun. The correction procedure on substituting the next chord from the "key difference 1" list finds that a transcription produces the words "the ten bases", which is syntactically correct. S/S is therefore accepted as the chord the operator originally indended.

Before incorporating syntax analysis in a correction procedure and re-running all the Palantype data, it was necessary to consider two important factors; the quality of the text and the unit that was being submitted for syntax analysis.

8.2 THE QUALITY OF TEXT BEING SUBMITTED FOR SYNTAX ANALYSIS

Parsing the output from Palantype transcription systems poses a different set of problems to parsing "book type" English text. Consequently, different parsing strategies need to be developed in order to obtain a meaningful syntactic analysis of the output text. These different strategies are necesary because of the following reasons:

i) The output from Palantype transcription systems does not consist wholly of English words. This is due to the fact that when an error is too difficult to correct, the transcription program resorts to using the transliteration process which produces a pseudo-phonetic translation of the Palantype chord or chords. For example, the chord P+RI. would be output as "brea".

ii) A keying mistake can cause a different English word to the intended one to be output. For example, if the operator typed CE.M instead of C+E.M, then the word "came" would appear in the output instead of "game".

iii) The operator sometimes fails to press a chord at all. Missing chords can result in whole words being omitted from the output text, and can cause more transliteration due to incomplete words.

iv) The occurrence of homophonic ambiguities. If from a list of alternative homophones, e.g. cheque/check/Czech, the wrong word is selected, the text could be rendered meaningless.

v) As Palantype is a phonetic shorthand, chords which are sometimes incorrectly joined together will produce English words. For example, "sell" "fish" becomes "selfish".

vi) The high degree of error and ambiguity present in speech itself. As the Palantype operator is recording verbatim, these errors are transferred through the transcription system to appear in the output.

vii) There is only a very limited amount of punctuation in the output from Palantype transcription systems. Usually only periods and commas are found in the text and even these are frequently omitted.

It was mentioned in Chapter 7 that the implications of all this for a parser was that its input would at some points be lists of alternatives. As will be seen in what follows, these alternatives have been presented separately. This was done for implementational reasons; it took less time to implement than putting the extra intelligence into the parser.

8.3 WHAT UNIT TO SUBMIT FOR SYNTAX ANALYSIS

As an error can occur anywhere in a sentence, the number of words processed since the last period was encountered (and to which the words produced by a successful re-transcription are to be added), will be variable. A problem arises therefore of what unit to submit for syntax analysis. If all the words processed since the last period up to and including the words produced by a successful re-transcription are submitted and there happen to be a lot, then the more prone they will be to the factors mentioned above. Alternatively, submitting only the words produced by the successful re-transcription (on average, two words) may not be effective since there are often too few to distinguish between different syntactic units. After studying output text from different operators, the unit decided upon was the part of the current sentence which succeeded the latest comma ...... encountered, up to the most recent word built by the "history and future" routine, and including the words produced by a successful re-transcription. For example, if the words produced by a successful re-transcription are "big peanut", then in the following text the phrase "the big elephant ate the big peanut" would be submitted to syntax analysis.

.......at the zoo, the big elephant ate the big peanut

Similarly, if the words produced by a successful re-transcription are "elephant ate", then in the following text the phrase "The elephant ate" would be submitted for syntax analysis.

.......at the zoo. The elephant ate

Note in this latter example there is no comma in the current sentence. If there was a keying mistake in the first word of a sentence, then the corrected word was not submitted for syntax analysis. The accepted word was the one most likely intended by the operator.

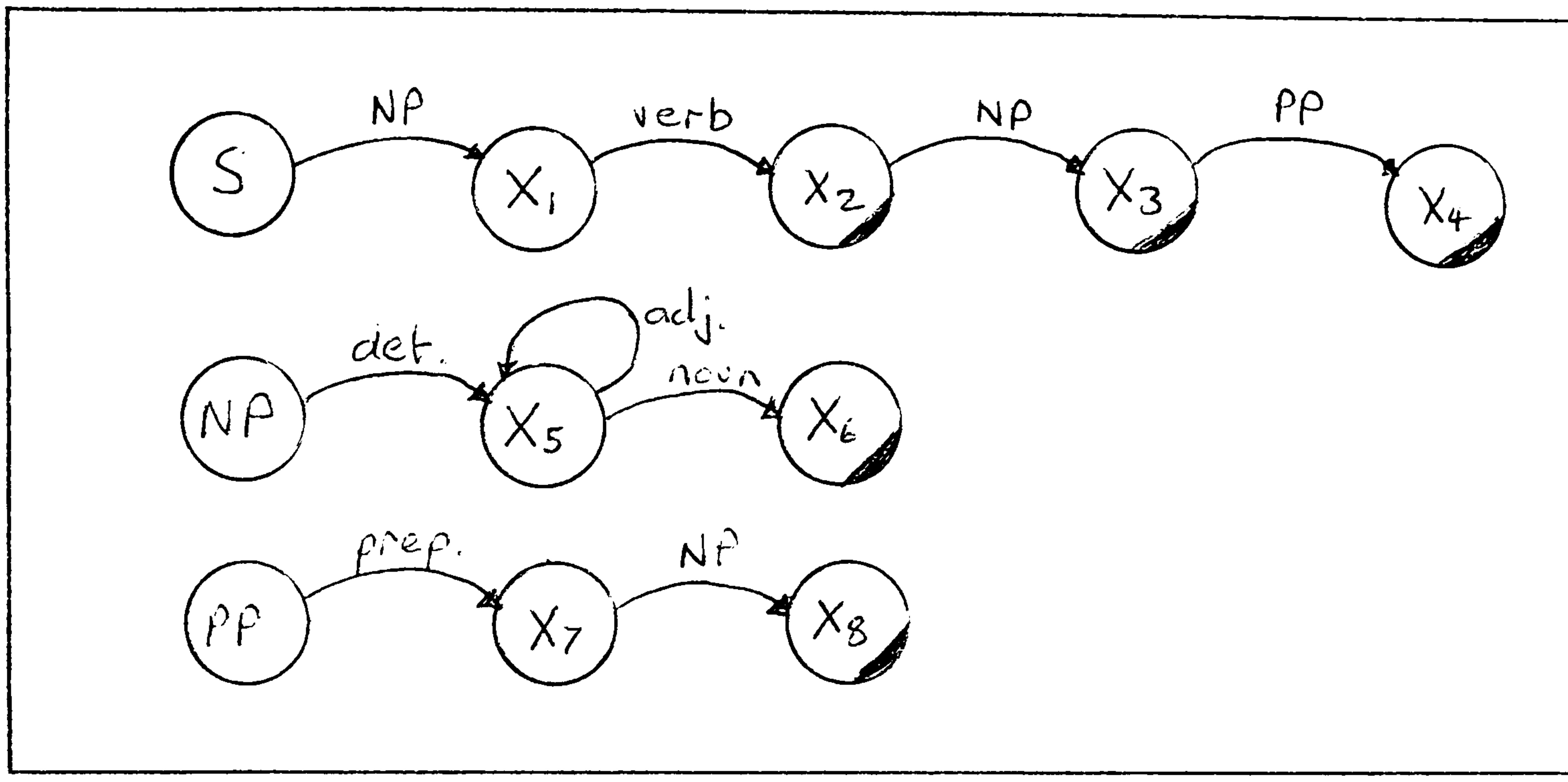
One reason for this choice of unit was because the only punctuation marks which appeared in the analysed output texts were the period and the comma. Also, one dominant use of the comma described in (3) is to "separate closely associated units within a sentence which are so constituted that each part has the elements necessary to operate as a sentence in its own right". This seemed to be especially true in Palantype output text.

8.3.1 A Feature of the Syntactic Rules

It was decided that even if a unit did not have the elements necessary to operate as a sentence in its own right, it would still produce a successful parse if it had built a valid parse tree when the end of the unit was encountered. This meant that a special character had to be appended to the end of each unit being submitted for syntax analysis, to inform the parser that it had reached the end of the input string.

Consider the following simple transition network:

where non/terminals are in UPPER-CASE, terminals are in lower-case and X2, X3, X4, X6 and X8 are final states.



For this transition network, the following strings would produce successful parses:

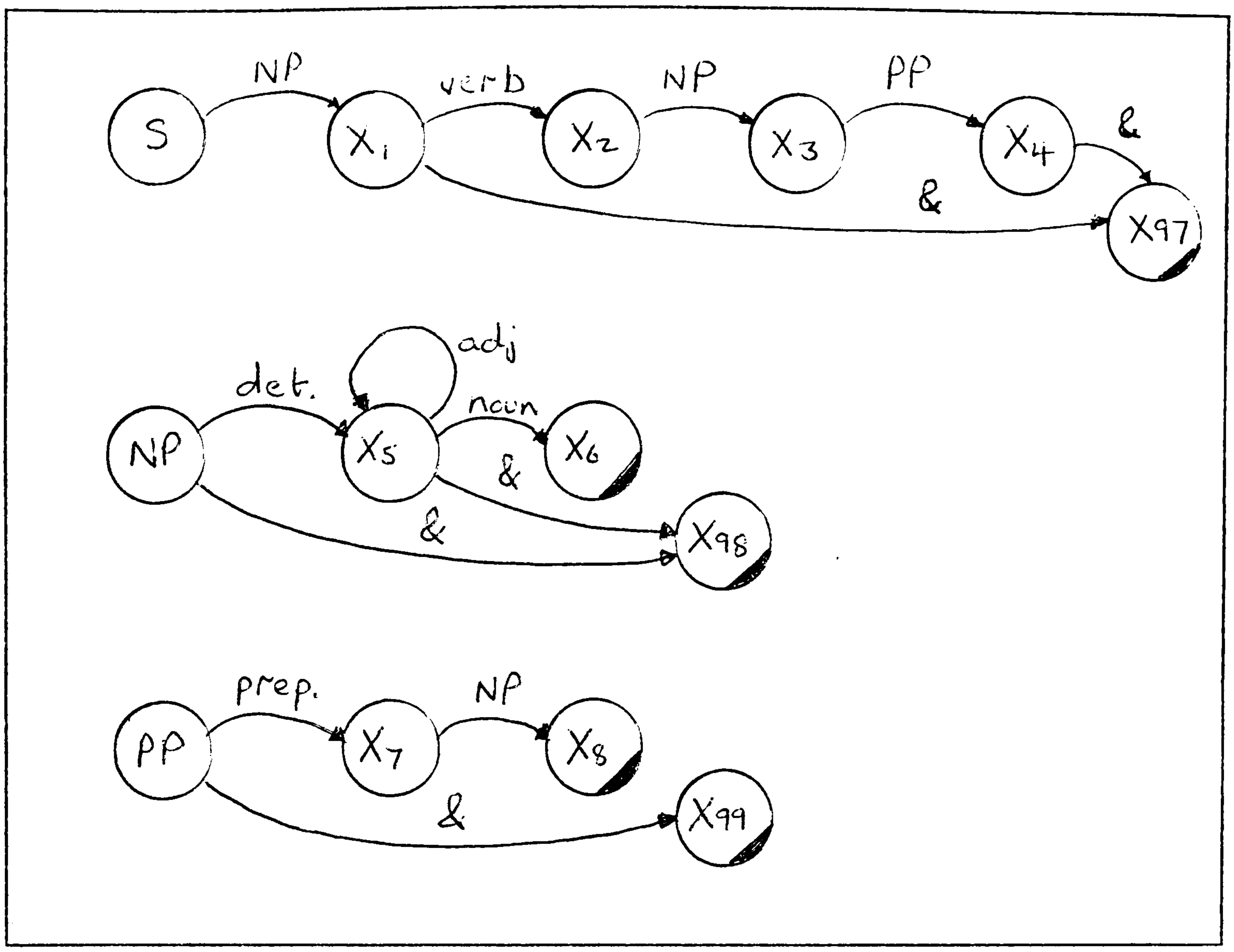
The big elephant ate

The big elephant ate the peanut

The big elephant ate the peanut in the park

etc.

To allow incomplete sentences to produce successful parses the above transition network would have to be modified so that every terminal arc has an alternative arc which will be followed to a final state if the end-of-unit marker is encountered. (The character "&" was chosen as the end-of-unit marker.) This would produce the following transition network:



where "&" is the end-of-unit marker and X6, X8, X97, X98 and X99 are final states.

With this transition network, the following strings would produce successful parses:

the big elephant ate &

the big elephant ate the peanut &

the big elephant ate the peanut in the park &

the big &

the big elephant &

the big elephant ate the &

the elephant ate the big peanut in &

whereas, for example, the following would produce parse fails:

the big elephant ate (no end-of-unit marker)

the big ate &

the ate &

the big elephant ate the peanut in ate &

the elephant the the peanut &

etc.

N.B. When the end-of-unit marker is scanned by the parser, the pointer to the input string is not advanced.

8.3.2 Multiple Errors

Sometimes, due to the factors mentioned earlier affecting the quality of text produced by Palantype transcription systems, there is more than one error in the unit which is being submitted for syntax analysis, or the unit contains a homophonic ambiguity. When a unit contained an homophonic ambiguity as well as an error, it was submitted for syntax analysis with only one of the ambiguous alternatives included in the unit, i.e. the most frequently used one in everyday English. This signifies of course that error correction and homophonic ambiguity resolution are being done separately. As a result of multiple errors in a unit, it may fail a parse before the parser has arrived at the words produced by a successful retranscription. For example, the words produced by a successful retranscription in the following text, are "the park".

........at the zoo. The big elephant late the peanut in the park

On submitting "The big elephant late the peanut in the park" to syntax analysis, the parse fails at the word "late", and so we are not able to determine whether "the park" was a syntactically correct part of the unit being parsed.

Under these circumstances, the strategy decided upon was to ignore all the words up to and including the word at which the parse failed, and re-submit the remaining words to syntax analysis as a separate unit. Therefore, in the above example, the words "the peanut in the park" is submitted for syntax analysis. Adopting this strategy means that a parse can start anywhere in a sentence and this factor must be considered in the development of the grammar and parser.

8.4 SYNTAX ANALYSIS AS PART OF A CORRECTION ROUTINE

Consider the output in fig. 8C which was produced by a transcription program incorporating neither error correction, syntax analysis or the preprocessing corrections described in Chapter 5. Fig. 8D shows the output produced from the same input string after the preprocessing corrections of Chapter 5 had been implemented. If the same input that produced that output was now fed to a 'History and Future' correction routine modified to include syntax analysis, the process could be described as follows. In the first line, the transcription program on receiving the chord for the word "go" knows that there is an error present in the input string and so the correction routine is invoked. The correction routine assumes "CA" to be in error and, with its ability to look-ahead in the input string, its first attempt at correcting CA produces the string "Alternatively we cat go to". In a 'History and Future' routine without syntax analysis, this would have been accepted as a valid correction since a successful retranscription had taken place. However, the syntax analyser rejects "Alternatively we cat go to" as syntactically incorrect. Although the parse failed at the word "cat", the phrase "go to" is not submitted for syntax analysis since it is part of the phrase produced by the successful retranscription. The correction routine's next attempts are shown in fig. 8B. The final one was found to be syntactically correct and so its analysis is shown.

```
" Alternatively we cap go to "  - REJECTED
" Alternatively we cargo to "   - REJECTED
" Alternatively we cam go to "  - REJECTED
" Alternatively we can go to "  - ACCEPTED

Analysis: ((((Alternatively : ADVERBIAL)((we :PRONOUN) :SUBJECT)
((can :AUX)(go :VERB) :VERBPHRASE)(to :PREP)) :CLAUSE) :SENTENCE)
```

Fig.8B Analysis of "Alternatively we can go to"

In the second line, the transcription program becomes aware of an error when it encounters the chord SH/N (last part of the word "organisation") and so the correction routine is invoked. The chord E. is assumed to be erroneous, and the first attempt at correction produces "Alternatively we can go to the deaf organisation the" and so is submitted for syntax analysis. Remember, the unit submitted for syntax analysis is everything from the previous comma up to the corrected phrase. If no comma exists, then it is everything from the start of the sentence (the last full stop), up to the corrected phrase (If a full stop has been omitted, then more than one sentence is subjected to syntax analysis). An analysis is shown below.

```
Analysis: ((((Alternatively : ADVERBIAL)((we :PRONOUN) :SUBJECT)
((can :AUX)(go :VERB) :VERBPHRASE)(((to :PREP)(the deaf
organisation): PREPPHRASE) :PLACE ADVERBIAL) :CLAUSE)):SENTENCE)
```

Line three has the same error, but still no comma has been encountered and so the phrase submitted for syntax analysis is shown below.

```
    " Alternatively we can go to the deaf organisations the
      hard of hearing organisations and "   - ACCEPTED

Analysis: ((((Alternatively : ADVERBIAL)((we :PRONOUN) :SUBJECT)
((can :AUX)(go :VERB) :VERBPHRASE)(((to :PREP)(the deaf
organisation): PREPPHRASE) :PLACE ADVERBIAL) :CLAUSE)((the hard
of hearing organisations):SUBJECT) and ): CLAUSE) :SENTENCE)
```

In the sixth line, the correction routine gets it right first time again, but consider the error in line 14, T+JEN, the first attempt at correcting this error produces the following phrase. " We have started looking at audience responses and we try to differentiate between the responses of the hearing impaired audience an the responses of a nor mall hearing awe Dane & ". This time the parser fails before it has even reached the modified phrase; it cannot get past the word "an", which should have been "and". The phrase is shortened to omit all the words up to and including "an", so the phrase submitted for syntax analysis now looks like this:- "The responses of a nor mall hearing awe Dane & ".(The desired output is "The responses of a normal hearing audience").

Although the phrase produced by the correction routine looks as though it should be rejected, it was in fact accepted with the following analysis.

```
Analysis: (((( The :DET)(responses of a nor mall)) :SUBJECT)
((hearing :VERB) :VERBPHRASE)((awe dane): OBJECT)) :CLAUSE)
:SENTENCE)
```

To the 'purists', accepting "The responses of a nor mall" and "awe Dane" as nounphrases will seem altogether wrong. However, since there is no one criterion that can be used for a general definition of compounds in English, the approach adopted was to specify a very general rule which, although it allowed illegal structures, did not reject valid ones.

As the grammar was developed in a short period of time, inadequacies of this kind made it less effective than it might have been. A better parsing strategy would have been to allow more ambiguity at all parts of the input string instead of only producing alternative sub-strings for the part of the input string which had been assumed to be erroneous. An alternative approach to syntax analysis in error correction is discussed in Chapter 6, section 3.3. Fig. 8E shows the output produced by the 'History and Future' routine without syntax analysis. Fig 8F shows the output with syntax analysis. More examples of output are given in appendix D.

8.4.1 Results

The effect of syntax analysis on the quality of the output text was disappointing. It was mentioned in Chapter 6 that a 'History and Future' correction routine without syntax analysis corrected 52% of all errors. The same routine incorporating syntax analysis was only capable of improving this figure to 57% overall. The main reasons for this relatively poor performance are as follows:-

1. Incomplete specification of syntactic rules
2. Definition of certain rules too general
3. Errors and Homophonic ambiguities were treated separately which meant that when the program attempts to correct errors the string under consideration may contain an incorrect homophone
4. The quality of the text being submitted for syntax analysis was very poor
5. Need for a better parsing strategy

Fig.8C shows the output from a transcription program incorporating no error correction whatsoever. Fig.8D shows the same output but after the preprocessing corrections of Chapter 5. Fig.8E shows the output after incorporating the preprocessing corrections and the 'History and Future' correction routine (but with no syntax analysis). Fig.8F shows the same as 8E but with syntax analysis. In figs. 8E and 8F successful corrections are underscored with an unbroken line and failures are underscored with a broken line.

Alternatively we CA go TOTH deaf or C+NAI Asians the hard of hearing or C+NAI Asians ANT+ say what do you THINCTH hearing impaired want subtitled + ANT+ they tend to PRO to use lists which look something like that . We have started looking at audience RES PON SS ANT+ we try to T+I FREN HJE.T P+E TFI.NTH RES PON SS of the hearing impaired audience an the RES PON SS of a nor mall hearing awe T+JEN + ANT+ we find that there is not very much difference in fact P+E TFI.NTH programmes these two groups of PLI. watch + so there is some evidence to say that one should concentrate on peak MFEUIN+ times audience RE. TIN+S . The next THIN+ we need TOAS C is + what P+A lance should we draw between quality ANT+ amount . There is all FEIS going to dee a limb TAT+ mount of money in the kitty + do fee subtitle as many programmes as we possibly can but not worry too much bout how CT+OUT+ the subtitles are so that we can do it cheap LI + or do we say we will make HOURTH subtitles are the best THIN+ we can possibly give you + ANT+ I NEFT P+LI that will + mean that fewer programmes get subtitled because it will take more time ANT+ cost more money . . . Perhaps some programmes are unsuitable for the hearing impaired + perhaps some are PTFI CEULR LI unsuitable for the PRE Lynn C+EU LI deaf . For in STNS + humour is not very easy to get over in a subtitled form . Huns ant punts SOEAN on may not have a meaning HFEIN the written form . . . That was puns ANT+ so on . . .

Fig.8C Uncorrected output

Alternatively we CA go to the deaf or C+NAI Asians the hard of hearing or C+NAI Asians and say what do you think the hearing impaired want subtitled , and they tend to PRO to use lists which look something like that . We have started looking at audience responses and we try to T+I FREN HJE.T between the responses of the hearing impaired audience an the responses of a nor mall hearing awe T+JEN , and we find that there is not very much difference in fact between the programmes these two groups of PLI. watch , so there is some evidence to say that one should concentrate on peak viewing times audience RE. TIN+S . The next thing we need to ask is , what P+A lance should we draw between quality and amount . There is all FEIS going to dee a limb TAT+ mount of money in the kitty , do fee subtitle as many programmes as we possibly can but not worry too much bout how CT+OUT+ the subtitles are so that we can do it cheaply , or do we say we will make HOUR the subtitles are the best thing we can possibly give you , and I NEFT P+LI that will , mean that fewer programmes get subtitled because it will take more time and cost more money . . . Perhaps some programmes are unsuitable for the hearing impaired , perhaps some are PTFI CEULR LI unsuitable for the PRE Lynn C+EU LI deaf . For in STNS , humour is not very easy to get over in a subtitled form . Huns ant punts SOEAN on may not have a meaning when in the written form . . . That was puns and so on . . .

Fig.8D Output produced by a transcription program incorporating the preprocessing corrections described in Chapter 5.

Alternatively we cat go to the deaf organisations the hard of hearing organisations and say what do you think the hearing impaired want subtitled , and they tend to produce lists which look something like that . We have started looking at audience responses and we try to differentiate between the responses of the hearing impaired audience an the responses of a nor mall hearing awe Dane , and we find that there is not very much difference in fact between the programmes these two groups of Leigh watch , so there is some evidence to say that one should concentrate on peak viewing times audience ratings . The next thing we need to ask is , what balance should we draw between quality and amount . There is always going to dee a limb dad mount of money in the kitty , do fee subtitle as many programmes as we possibly can but not worry too much bout how good the subtitles are so that we can do it cheaply , or do we say we will make sure the subtitles are the best thing we can possibly give you , and inedibly that will , mean that fewer programmes get subtitled because it will take more time and cost more money . . . Perhaps some programmes are unsuitable for the hearing impaired , perhaps some are particularly unsuitable for the pressing gulley deaf . For in Stens , humour is not very easy to get over in a subtitled form . Huns ant punts soun on may not have a meaning when in the written form . . . That was puns and so on . . .

Fig.8E Output produced by a transcription program incorporating the preprocessing corrections described in Chapter 5 and the History and Future correction routine.

Alternatively we can go to the deaf organisations the hard of hearing organisations and say what do you think the hearing impaired want subtitled , and they tend to produce lists which look something like that . We have started looking at audience responses and we try to differentiate between the responses of the hearing impaired audience an the responses of a nor mall hearing awe Dane , and we find that there is not very much difference in fact between the programmes these two groups of bees watch , so there is some evidence to say that one should concentrate on peak viewing times audience ratings . The next thing we need to ask is , what balance should we draw between quality and amount . There is always going to dee a limb dad mount of money in the kitty , do fee subtitle as many programmes as we possibly can but not worry too much bout how good the subtitles are so that we can do it cheaply , or do we say we will make sure the subtitles are the best thing we can possibly give you , and inedibly that will , mean that fewer programmes get subtitled because it will take more time and cost more money . . . Perhaps some programmes are unsuitable for the hearing impaired , perhaps some are particularly unsuitable for the pressing gulley deaf . For in Stens , humour is not very easy to get over in a subtitled form . Huns ant punts sown on may not have a meaning when in the written form . . . That was puns and so on . . .

Fig.8F Output produced by a transcription program incorporating the preprocessing corrections described in Chapter 5, the History and Future correction routine and syntax analysis.

8.4.2 Some conclusions

It can be seen that the syntax analyser used in error correction improves the output very little (compare fig.8E and 8F). However, what is transparent is that many corrections have been reaffirmed by the parser. In retrospect, the strategy of ignoring all the words up to and including the word at which the parse failed and resubmitting the remaining words to syntax analysis was not a good one, i.e. it was not as effective as was expected. This was mainly due to the fact that the remaining 'phrase' was often too short to distinguish between different syntactic units. A better approach would have been the more general one described in Chapter 6, (section3.3), where alternatives are also listed for the word at which the parse fails. Also a more general strategy means that homophonic ambiguities and errors can be treated together.

Chapter 9

RESOLUTION OF HOMOPHONIC AMBIGUITIES

9.1 INTRODUCTION

As Palantype is a phonetic shorthand, words which sound the same are Palantyped identically. For example, the words "for", "four" and "fore" are represented in Palantype by the single chord F/OR . As mentioned earlier human transcribers are able to draw on many knowledge sources to enable them to resolve this kind of ambiguity. In chapter 7, one of these knowledge sources, syntax, was used to select a grammatically correct phrase from a list of alternatives in the process of automatic error correction. The same procedure has been adopted here to help resolve homophonic ambiguities. However, this is not the first time that syntax analysis has been used in the resolution of homophonic ambiguities. Therefore before describing how it has been implemented in the present research, the techniques used in a previous attempt at resolving homophonic ambiguities shall be described.

9.2 A PREVIOUS ATTEMPT AT RESOLVING HOMOPHONIC AMBIGUITIES

The dictionary used on this project was very kindly provided by the National Physical Laboratory and used by them on an earlier project (Price,1971). An analysis of the dictionary conducted at N.P.L. showed that of the 75,000 words, 1052 were one of a group of homophones. Some of the methods used at N.P.L. for resolving ambiguities affected the dictionary directly. It is therefore worth describing them here since a significant percentage of homophonic ambiguities were resolved or precluded before the dictionary was used on the present project.

Of the 1052 ambiguous groups ,

> 48 arose from different spellings of proper names, most of which could be reduced to one spelling (with a few exceptions, e.g. "Faroe'Pharaoh). 363 were of proper names with ordinary words (defined as all words which are not proper names) and 610 were of ordinary words with other ordinary words, such as "infest'invest", "incite'insight",etc.

(Price,1971).

9.2.1 Changes to the recording convention

Discussions were held between the Palantype Organisation and the National Physical Laboratory as to the extent to which ambiguities could be eliminated by changes in the recording convention. It was clear that changes ought to be kept to the minimum in view of the problems of retraining existing Palantype operators.

After these discussions, it was insisted that the diacritic "+" (which by convention makes a voiced consonant out of an unvoiced e.g. F+ stands for V), be used in those situations where its use avoided the creation of an ambiguity. This prevented ambiguous pairs such as "failed" and "veiled". An extension of this idea was to provide an additional "+" key, - on the right for use with right hand consonants. This was achieved by replacing the right hand D key with a "+" key (a right hand D was subsequently represented by T+). This modification prevented ambiguous pairs such as "belief" and "believe", "cap" and "cab", etc. As Price explains,

> the left hand D was likewise removed to make way for H, previously represented by the + character itself. The last change was to replace K by C, giving a more rational representation for such consonant clusters as CH (previously represented arbitrarily by SK).

As a result of these small modifications, a significant proportion of ambiguities were prevented; only 325 ordinary word groups remained from the original 610, and 310 of the proper name/ordinary word groups remained from the original 363. Figures 9D and 9E show the layout of the original Palantype keyboard and the modified layout respectively.

9.2.2 <u>Exploiting local context</u>

> Many important ambiguities (e.g. those that occur frequently) can be resolved correctly in most cases by the introduction into the dictionary of word group entries that permit context to be taken into account to a limited extent. To take a few examples, the inclusion of a dictionary entry for the phrase "to be" will avoid the ambiguity "be/bee" where the preposition "to" precedes the ambiguous chord. An entry for "would be" avoids both "would/wood" and "be/bee"; "so called" avoids "so/sew/sow"; "take hold" avoids "hold/holed" and so on. Altogether over 600 phrase entries have been added to the dictionary. The device of including an ambiguity in a special phrase entry is not dependent on the position of the ambiguity within the phrase, it can be initial, medial or final. The phrases included have to be chosen with great care and even then the principle may break down on occasion, as in the phrase "would this wood be suitable", which would emerge as "would this would be suitable" if one allows the phrase entry "would be".

(Price, 1971)

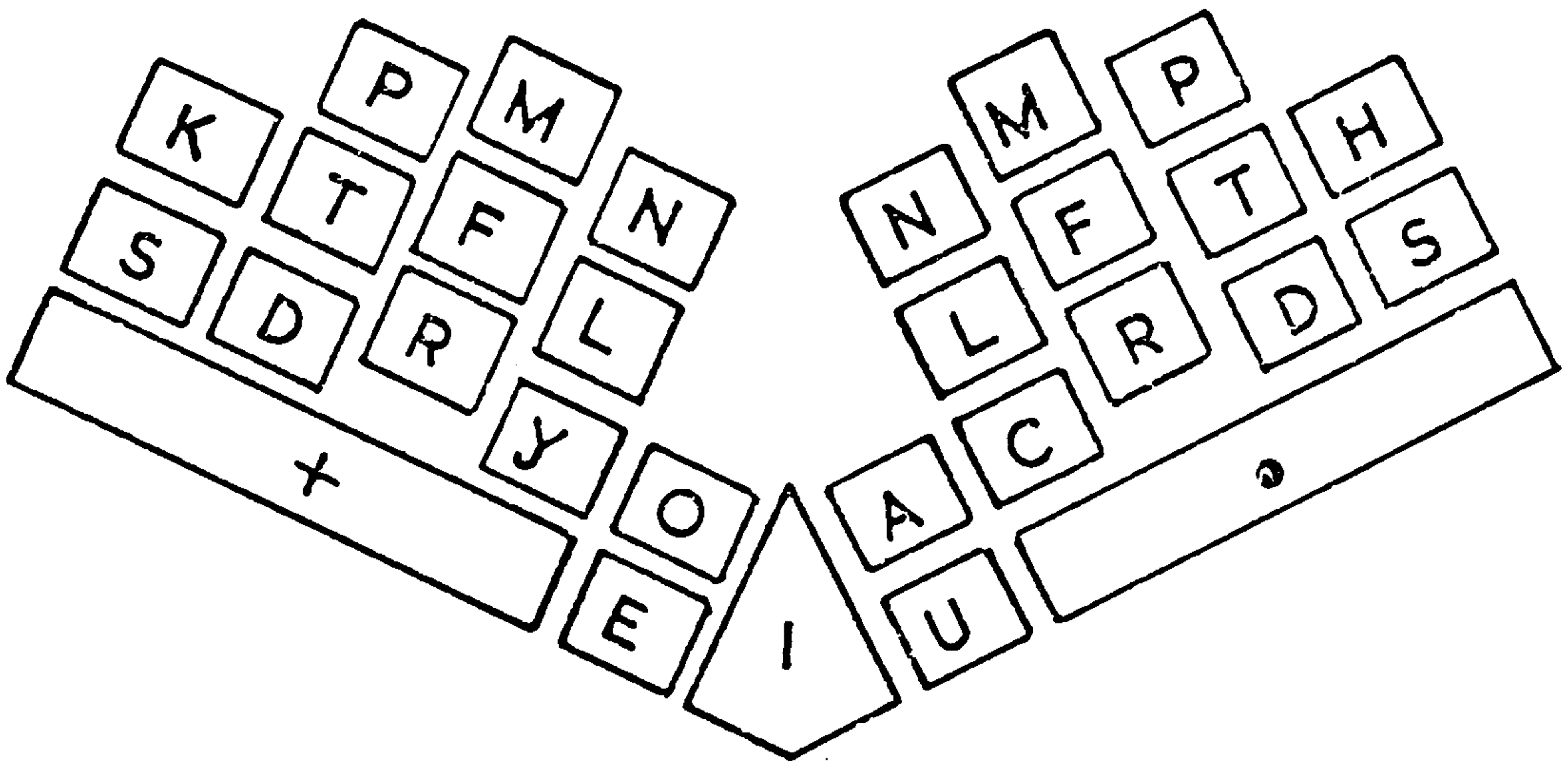


Fig. 9D Layout of Original Palantype Keyboard

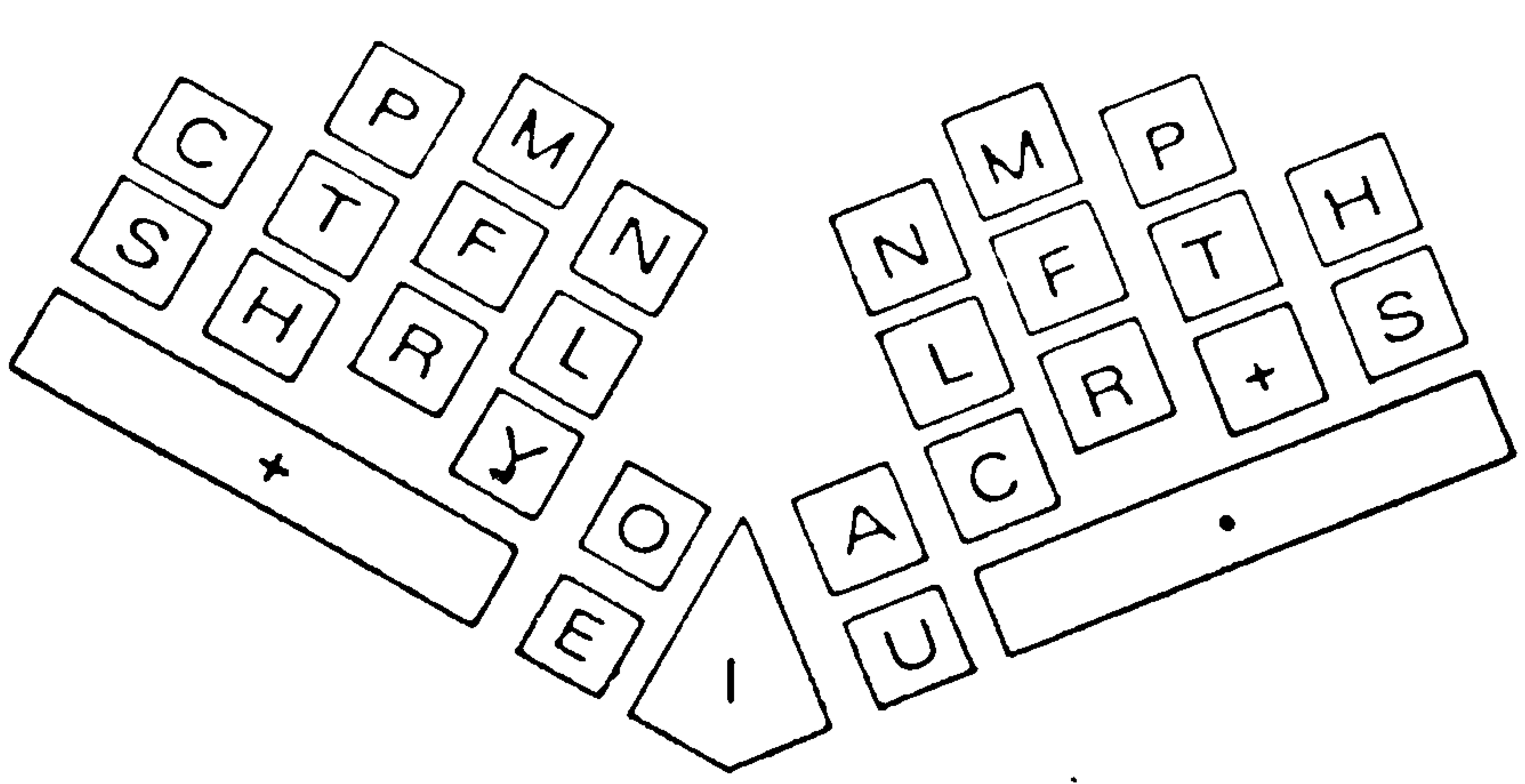


Fig. 9E Layout of Modified (present) Palantype Keyboard

The technique of exploiting local context by the inclusion in the dictionary of unique high frequency phrases was also implemented by Galli, 1962. It is successful due to the search mechanism, which always looks at longer dictionary entries first, (see Chapter 3). The search mechanisms used by Galli, Price and the present project are all based on the principle of obtaining the longest lexical match between the various dictionary entries and the input string of chords.

The dictionary still contained the 600+ phrase entries when it was received for use on the present project.

9.2.3 Restricted syntax analysis

A further method of resolving homophonic ambiguities adopted at N.P.L. and described in Szanser (1971), was that of restricted syntax analysis. The two restrictions imposed on the syntax analyser were as follows:

i) there was no possibility of marking with grammatical codes all the dictionary entries, i.e. the syntax analyser did not know the grammatical classification of every word that it parsed.

ii) it had to be "brief and fast, so as not to reduce the operational speed of the transcription", i.e. syntax analysis was to be included as part of the real time transcription process.

Considering the first restriction, Szanser found the work of Thorne et al. (1966) very useful because the basic principle behind the construction of Thorne's model for "the perception of syntactic structure" was that it should be possible to analyse sentences without recourse to a complete dictionary. However, the Thorne model did not dispense with a dictionary altogether, it used a closed-class dictionary of closed-class words which play a major role in the syntax of a language. Determiners, prepositions and pronouns are all examples of closed-class words, as opposed to open-class words such as verbs and nouns.

In Brately & Dakin (1968), the production of the closed-class dictionary used by the Thorne model, is described in detail. The following extract, which demonstrate the principles behind the model, is taken from the Bratley & Dakin paper.

> "Consider for a moment the well known lines:
> 'Twas brillig, and the slithy toves
> Did gyre and gimble in the wabe.'
>
> It is hard to maintain that the lines actually mean anything even if one knows that toves, for instance, are little animals that live under sundials (or is that borogroves?). Nevertheless, it is not at all unreasonable to assert that the lines have a readily perceived syntactic structure. In everyday terms, we might say that slithy is clearly an adjective, tove is a noun, and gyre and gimble are verbs.
>
> Notice what we are doing: we are not saying, 'I know that tove is a noun, therefore the syntactic structure is so-and-so.' On the contrary, we are saying, 'The syntactic structure is apparent, and knowing the structure, I can now assert that tove is a noun.'
>
> On the other hand, suppose we are faced with the line:
> William the the and disappointed.
>
> No syntactic structure is now recognisable, even though all the words in the line are well known to us."

Szanser, in developing a syntax analyser based on the concepts underlying the Thorne model, grammatically coded the ambiguous homophones as well as the closed-class words. This seemed a very good way of overcoming the first restriction.

Considering the second restriction, Szanser decided that as the analysis had to keep up with the speed of transcription, "there was no possibility of making full analyses of phrases, let alone sentences." His alternative strategy was to perform a local analysis, i.e. parsing the immediate neighbourhood of the ambiguous word. For example:

contents read/red out....

would be subjected to syntax analysis. The analysis consisted of applying a set of special subroutines, operating in accordance with the syntactic class of the ambiguity component met when reading the text. Each of the subroutines (about 20 in all) scanned the surrounding words in search for incompatible syntactic combinations (as a rule), or for "typical" combinations (as an exception). Firstly the ambiguous components were characterised. In the above example this would be as follows: (taken from Szanser, 1971).

READ - verb, general form, or verb, past tense, or past participle

RED - adjective or noun.

The syntactic profile of the surrounding words is as follows: article, general purpose word ending with "-s", (ambiguity), preposition. A short analysis for RED produces the following: the general purpose word with "s" following immediately an article is a noun plural; therefore a nominal block including another noun or adjective is unlikely.

The other option, READ, meets with no restrictions. It is therefore accepted as the solution.

Szanser's goal with restricted syntax analysis was 50% successful resolution of homophonic ambiguities. His method in fact was quite successful and 65% correct resolutions were achieved. It should be noted however, that Szanser's experiments were conduction on manually corrected texts, i.e. output consisting wholly of English words without keying errors, etc.

9.3 FULL SYNTAX ANALYSIS

Having described a previous attempt at homophonic ambiguity resolution by syntax analysis, the reader is aware of some of the difficulties associated with this problem. However, the experiments conducted in the present research were much different from those conducted by Szanser, for three reasons.

1. All the words that were encountered in the output texts were grammatically coded before the texts were submitted to syntax analysis. This was equivalent to having a fully coded dictionary.

2. There was no need to have a "fast" or "brief" parser, since no attempt was being made to work within the restrictions imposed by a 'real time' goal.

3. The texts submitted for syntax analysis were not manually corrected beforehand. They contained all the errors present in speech itself and those introduced by the recording process.

9.3.1 Implementation

The syntax analyser was used very much in the same way as for error correction; when a homophonic ambiguity was encountered, one of the alternatives only was put into the current phrase and submitted for syntax analysis. If this produced a valid parse the chosen word was accepted as the intended word. If not, then the next alternative was tried. The alternatives were always ordered by their frequency of use in everyday English. This meant that if two alternatives were of the same word class, the one used most frequently would be selected. Similarly, if the parser was unable to produce a valid parse for any of the alternatives, then the one used most frequently was selected as the intended word. See figures 9A to 9C.

9.3.2 Unit submitted and multiple errors

The unit submitted for syntax analysis was the same as that in error correction; the part of the sentence which succeeded the latest comma to be encountered, up to the most recent word built by the 'History and Future' routine, and including one of the homophonic alternatives. In the cases where multiple errors were encountered which prevented the parser from reaching the homophonic ambiguity, then a phrase was shortened to exclude everything up to and including the word at which the parse failed.

9.3.3 Results

The output in figure 9A was produced by a transcription program incorporating neither error correction, 'preprocessing' corrections or syntax analysis. Note that at the points in the output where there are homophonic ambiguities, all alternatives have been listed. The output in figure 9B was produced from the same input string, but after the preprocessing corrections had been made and the 'History and Future' correction routine had been applied. Fig.9C shows the output produced by a transcription program incorporating the 'history and future' correction routine and 'calling' the syntax analyser each time a homophonic ambiguity was encountered. Note however, that in this output the syntax analyser was not used to check error corrections. Syntax analysis used in homophonic ambiguity resolution caused the correct alternative to be selected for 43% of the total number of ambiguity occurrences. The reasons for this rather poor result were given in the results section of the previous chapter, namely incomplete specification of syntactic rules, some rules too general, errors and ambiguities treated separately, a poor parsing strategy and the poor quality of text being submitted to the parser. In fig.9C successful resolutions are underscored with an unbroken line, incorrect resolutions are underscored with a broken line, situations where the parser failed to produce a valid parse from any of the alternatives and so chose the one occurrring most frequently in everyday use are underscored with a line of x's, and situations where the parser produced a successful parse for all alternatives and so again chose the one occurring most frequently in everyday use, are underscored with a dotted line.

What a burden that H+AS placed on the business community . We talk bout CROUL TI to animals + what about CROUL TI to T+REC TRS . ANT+ so/sow/sew there has been an enormous growth within the pry fat INT+ STRI ANT+ CO MUR of a counter bureaucracy to comply wit the T+MOANFT+ new LE +JIS LE. SHN . We will do all we can to reduce this this load + but may I suggest one/won PRACT CL way/weigh in/inn which you can HELPTH cause/caws of in/inn does TRI ANT+ CO MUR in/inn parliament . We have too many +JUR NLI STS + too many LOI.RS + too many teacher in/inn +FE STMFIN TR + an not enough of those who create the welt of this country + and so/sow/sew support all the others . . Applause . . . So many I make a plea to you to-day to make it easy + not difficult for your representatives to to EN TUR parliament . We don/t want another crop of sponsored MEMPS but F. Do want to HAINF+ parliament + shaping our/hour LE +JIS LE. SHN Those who have fur/fir T hand experience of PH+FOIT NES world is like . . . Ladies ANT+ +JENTS we are a PRO CHIN+ a critical turning point in our H+IS TRI +TH pry fat enterprise SIS TM is C+OIN to be C+I van NU THR chance to save our/hour country + its/it's C+OIN+ to P+I.TH STLA once . If we fail + free dumb fail with us . I am NOAT Marxist +I don't believe in/inn economic deter MNI STM but if we continue a long this pat of economic FI lure we can T+ES troy/Troy not lowing our/hour STAN T+OFRT+ life but fee shall T+ES troy/Troy our/hour free institutions as well .

Fig.9A Uncorrected output

What a burden that has placed on the business community . We talk bout cruelty to animals , what about cruelty to rectors . And so/sow/sew there has been an enormous growth within the pry fat IND STRI and commerce of a counter bureaucracy to comply wit the demand new ledgers lacing . We will do all we can to reduce this this load , but may I suggest one/won PRACT CL way/weigh in/inn which you can help the cause/caws of in/inn dusty and commerce in/inn parliament . We have too many journalists , too many layers , too many teacher in/inn WE STMFIN TR ,an not enough of those who create the welt of this country , and so/sow/sew support all the others . . Applause . . . So many I make a plea to you to-day to make it easy , not difficult for your representatives to to inter parliament . We don/t want another crop of sponsored mumps but fee do want to hive parliament , shaping our/hour ledgers lacing those who have fur/fir the hand experience of befit nets world is like . . . Ladies and gentlemen we are a prop CHING a critical turning point in our he trip , the pry fat enterprise system is cawing to be given other chance to save our/hour country , its/it's going to be the to lap once . If we fail , free dumb fail with us . I am nought/naught Marxist is don't believe in/inn economic deter MNI STM but if we continue a long this pat of economic fissure we can destroy not lowing our/hour standard life but fee shall destroy our/hour free institutions as/ass well .

Fig.9B Output produced by a transcription program incorporating the History and Future correction routine.

What a burden that has placed on the business community . We talk bout cruelty to animals , what about cruelty to rectors. And so there has been an enormous growth within the pry fat IND STRI and commerce of a counter bureaucracy to comply wit the demand new ledgers lacing . We will do all we can to reduce this load , but may I suggest one PRACT CL way in which you can help the cause of inn dusty and commerce in parliament . We have too many journalists , too many lawyers , too many teacher in WE STMFIN TR , an not enough representatives of those who create the welt of this country , and so support all the others . . Applause . . . So many I make a plea to you to-day to make it easy , not difficult , for your representatives to inter parliament . We don't want another crop of sponsored mumps but fee do want to hive parliament , shaping our ledgers lacing those who have fur the hand of experience of befit nets world is like . . . Ladies and gentlemen we are a prop CHING a critical turning point in our he trip , the pry fat enterprise system is cawing to be given other chance to save our country , it's going to be to lap once . If we fail , free dumb fail with us . I am nought Marxist is don't believe in economic deter MNI STM but if we continue a long this pat of economic fissure we can destroy not lowing our standard life but fee shall destroy our free institutions as well .

Fig.9C Output after attempted ambiguity resolution.

Chapter 10

THE IMPLICATIONS FOR VOICE RECOGNITION SYSTEMS

10.1 BACKGROUND

In recent years a great deal of research has gone into developing techniques and systems for speech input to machines (for a review, see Reddy (1976)). Although there has been a moderate amount of success, to the extent that a few commercial systems are available, these have in the main been limited to the recognition of isolated spoken words from a very restricted vocabulary for a limited number of users.

10.1.1 Why speech input ?

Although it has been stated that one reason for attempting to recognise speech is because the difficulty of the task constitutes an intellectual challenge, a more practical reason is to be able to use speech to communicate with computers. Besides it being the most natural form of communication, there are many applications where using speech would be advantageous. For example, it is faster and more convenient than using a conventional keyboard, and it provides an alternative channel of communication for handicapped people or in situations where the users' hands are occupied.

10.1.2 Some techniques for speech recognition

The major problems associated with connected speech understanding systems have been concerned with the physical analysis of the speech signal, i.e. transforming the speech signal into a number of parameters which adequatly describe it and enable the appropriate information bearing elements to be extracted. One important question that has been asked is "at what level should speech be recognised ?". At one level English has about 45 distinct sounds or phonemes, so in theory it should be possible to recognise a spoken message. In practice, the designers of the early phoneme recognisers found that because speech is a continuous signal where sounds flow into one another, the boundaries between phonemes are not easy to detect. A higher level of representation is the word, and although these also are not readily separated in continuous speech, in certain systems they have been found to be a convenient concept with which to work. Three techniques for speech recognition shall now be described in outline.

10.1.2.1 Isolated word recognisers

This approach constrains the user to speak his words in isolation, so that word boundaries can be detected by some simple criteria. Isolated Word Recognisers (IWR's) incorporate a "classifier" which has to be trained by the user. This involves presenting the classifier with sets of features corresponding to known words in the vocabulary. This data is then used to establish templates for each word so that during recognition when the classifier is presented with some "pattern" representing a word, it can calculate which word in its vocabulary best "fits" the pattern. The two factors that have been found to have the greatest effect on recognition performance have been the choice of words in the vocabulary and the number of speakers using the system. That is to say that some words are more easily recognised than others, and the system works better if it is only used by one person. The classifier has to be "trained" separately for each new user.

10.1.2.2 Connected speech recognisers

It has already been mentioned why connected speech recognition is difficult, namely the problem in deciding where one element (be it word or phoneme) ends and the next one begins. Another reason is the way in which acoustic characteristics of sounds and words vary depending on the context in which they are spoken. There is a class of recogniser, named Connected Speech Recognisers (CSR's), which attempt to recognise every word in a spoken message using measurements derived from the input signal, and rigid syntax rules. The concept of a reference pattern or template for each word is still used where the vocabulary size is relatively small, but, when the vocabulary size becomes large and words start to have syllables in common, difficulties arise. The more usual approach in CSR's as explained by Underwood (1977), is to convert the measurements from the analyser into a string of symbols. The symbols used are frequently phonemic in nature, and the generation of the symbol strings is performed by examining the measurements from the analyser, grouping them into acoustically similar segments and labelling the segments. Recognition of the words is then carried out using a matching process with a phonemic dictionary. As the beginnings and ends of words are not known, the process usually proceeds in a left-to-right manner until the first word has been identified. Matching for the second word then begins. The occurrence of shared sounds means that in some cases it is necessary to back-track along the symbol string when a word has been detected. Recognition of the string of words is carried out using tree-searching algorithms and appropriate similarity measures in order to establish which sequence of words is most likely. As the stages (segmentation, labelling and matching) are subject to errors and uncertainties, it is necessary to allow alternative word matches to proceed. The number of possible matches that have to be attempted at each point has been restricted by ensuring that the words in the input string conform to a rather rigid syntax. Already it can be seen that this method has many similarities with the reconstitution of word boundaries in Palantype transcription. The early attempts at connected speech recognition showed the impossibility of designing machines that work at the phonetic level only. CSR's were more successful by working at more than one linguistic level and employing several knowledge sources to aid the recognition process.

10.1.2.3 Speech understanding systems

The final type of speech recognisers are called Speech Understanding Systems (SUS's) and are very different from IWR's and CSR's in that they are designed to mimic the human capability of understanding the intent of a message rather than recognising every word within it. It is common experience that some utterances are grammatically incorrect or badly-formed, yet the message conveyed by them may be quite clear to the human listener. The accuracy of a SUS is therefore judged in terms of its response to the message rather than the number of words recognised correctly. Most of the work on speech understanding systems has been funded by the Advanced Research Projects Agency (ARPA), in the United States, and the available literature is extensive(see Klatt (1977) for a review). The major design decisions in building an SUS are concerned with how the system should be organised so as to use the information from the different sources (e.g. articulatory, phonetic, lexical, syntactic, semantic or task-dependant) in a computationally efficient manner. The problems arising in CSR's from having imprecise representations of the signal at every level of processing are compounded in an SUS by the additional knowledge sources involved.

10.2 <u>A COMPARISON OF PALANTYPE CODE AND DIRECT SPEECH INPUT</u>

1. Word boundaries

   In continuous speech recognition a major problem is word boundary reconstitution. In Palantype code there is also no word boundary indication, but syllable boundaries are presented.

2. Errors.

   In continuous speech recognition phoneme recognisers make three kinds of errors in general; substitution, insertion and deletion. In Palantype recording, operators also make three kinds of errors; substitution, insertion and deletion.

3. Homophonic ambiguities.

Both speech recognition systems and Palantype transcription systems have problems with homophonic ambiguities.

4. Aids to sentence reconstruction.

In theory continuous speech recognisers ought to be able to use information presented in the form of stress, intonation, pitch, pauses, etc. Such prosodic features are not yet fully understood. In palantype there is some punctuation; mainly periods and commas.

10.3 SOME IMPLICATIONS

It has already been shown that the process of continuous speech recognition has many similarities with Palantype transcription; the previous section only enhances this. It is worthwhile investigating whether the parser and correction procedures developed for Palantype transcription systems could be used for speech recognition (and vice versa).

Palantype is a phonetic shorthand and the code is essentially a phonetic description of continuous speech. Therefore since the major part of the work has been done to get from Palantype code to English text, it is worth investigating whether it is possible to get from continuous speech to Palantype code (or something near). A feasibility study is presently being undertaken at Leicester Polytechnic by Guzy under the direction of Edmonds et al, (1980). A project of this complexity clearly requires more knowledge sources than the syntax used on this project. However, the translator writing system chosen to implement the parsing on the present project is a very powerful and versatile device. Not only does it provide the facility to do syntactic and semantic analysis, but it should also be possible to organise some of the other knowledge sources (e.g. phonological, articulatory, task dependent, phonetic) to interact with the parser when and where necessary.

One final table that the reader may find interesting, which was kindly provided by linguist John Connolly of Leicester Polytechnic, shows the 45 phonemes that linguists are used to dealing with, represented in terms of initial consonants, final consonants and vowels on the Palantype keyboard.

Phoneme	Palantype Initial Consonant	Palantype Final Consonant	Phoneme	Palantype Initial Consonant	Palantype Final Consonant
/p/	P	P	/f/	F	F
/b/	P+	P+	/v/	MF	F+
/t/	T	T	/θ/	TH	TH
/d/	T+	T+	/ð/	TH	TH(+)
/k/	C	C	/s/	S	S
/g/	C+	C+	/z/	S+	S+
/tʃ/	CH	CH	/ʃ/	SH	SH
/dʒ/	Y	Y+	/ʒ/	Y	
/w/	F+		/l/	L	L
/r/	R		/j/	Y	
/m/	M	M	/n/	N	N
/h/	H		/ŋ/		N+

	Palantype Vowels		Palantype Vowels
/I/	I	/i/	I.
/e/	E	/a/	A.
/a/	A	/ɔ/	O.
/ɒ/	O	/ u/	OU
/ʊ/	OU or U	/ɜ/	U.
/ʌ/	U		

	Palantype		Palantype
/ɪə/	EAR (or I.R)	/ɛə/	ER
/ʊə/	OUR	/eɪ/	E. (or EI)
/aɪ/	AI	/ɔɪ/	OI
/aʊ/	AU	/əʊ/	OE
/ju/	EU	/ə/	UR (or A)

Chapter 11

CONCLUSIONS OF THE RESEARCH AND PROPOSAL FOR FUTURE WORK

The aims of the present research have been achieved and as a result other avenues have been opened which need to be explored by researchers in the field. The main conclusions shall now be presented, indicating where appropriate the openings for future work. Finally a prediction shall be made of how dictionary based Palantype transcription systems might look in the future.

11.1 THE WORD BOUNDARY PROBLEM

In order to resolve the word boundary problem it is necessary to have an algorithm which is a) capable of producing output consisting wholly of English words from error-free input, and b) capable of producing all alternative transcriptions for a given sequence of chords. The reason for the second requirement is to avoid the problem of the incorrect joining of chords in situations where, for example, two chords considered separately produce two English words, yet considered together produce only one. Once a list of all alternative transcriptions has been produced it can be subjected to further processing, e.g. syntactic and semantic analyses, in order that the intended transcription may be determined.

In the present research an algorithm called WORDFINDER was designed which satisfies requirements a) and b) above. However, this algorithm was not implemented since for the major part of the research the two problems were viewed separately. Considering the two problems separately resulted in algorithms, 'two chord look ahead ' and 'one word look ahead', which were unable to determine all the different ways in which a given sequence of chords can be transcribed. It was only when the two problems were considered as one that WORDFINDER was designed to produce all alternative transcriptions. As WORDFINDER was only designed (and not implemented), there was no opportunity to present lists of alternatives to a syntax analyser or further processing. This is a proposal for future work in the field.

11.2 ERROR DETECTION AND CORRECTION

An analysis of transcribed texts produced by the computer showed that there were four situations in which a transcription program was unable to produce English words. Only one of these situations was attributable to actual operator error. In order to have a device for detecting the presence of an error in the input string, it was necessary to investigate the possibility of eliminating the three other situations in which the transcription program was unable to produce English words. The conclusions of this investigation are that although it is not possible to eliminate them completely, their adverse effect can be reduced. For dictionary based transcription systems it is worth investigating the possibility of implementing Winograd's 'word ending' algorithm to see whether the size of the dictionary can be significantly reduced, thus reducing search times etc.

Error correction procedures were designed and implemented; the most effective of them being the 'History and Future' routine which looked ahead in the input string to help it correct errors. The correction procedures were only effective on single errors. A more general solution which also copes with multiple errors was discussed.

The work at Southampton University, with an emphasis on prevention rather than correction, is improving the quality of the data input to transcription programs and thereby the quality of the output. Although no work has been done in the present research on improving ergonomic and physical aspects of the keyboard, significant steps forward are being made at Southampton University and the British Broadcasting Corporation (B.B.C.).

11.3 A GRAMMAR AND PARSER

Parsing the output from Palantype transcription systems is a much more difficult task than parsing 'book type' English text and poses a different set of problems. In the present research an attempt has been made to specify a set of syntactic rules particularly suited to parsing the output from Palantype transcription systems. These rules, defined as recursive transition networks, have been expressed in a tabular form in order to 'drive' SYNICS; a powerful translator writing system developed at Leicester Polytechnic. The specification of the syntactic rules is not complete, nor does it take into consideration the many varieties of English and its non grammatical uses. Although the base syntactic component has been modified to allow for structures in the input string which are not words, further modification is necessary in order to allow for missing words, extra words, wrong words and to predict syntactic categories.

11.4 SYNTAX ANALYSIS

Of the problems associated with Palantype code to English text transcription, there were three in which it was thought that syntax analysis might prove useful; homophonic ambiguities, keying errors and the incorrect joining of chords in word boundary reconstitution.

As already mentioned, syntax analysis was not used in word boundary reconstitution since WORDFINDER, an algorithm capable of producing all alternative transcriptions for a given sequence of chords, was only designed, not implemented in the research.

In the area of keying errors, the effect of syntax analysis was a little disappointing. The reasons for this are as follows:-

a) the strategy which decided what unit to submit for syntax analysis was not a good one

b) the specification of the syntactic rules was incomplete
   - The emphasis of never rejecting a phrase that was grammatically correct meant that the specification of certain rules was too general, with the result that phrases that were grammatically incorrect were sometimes also accepted

c) errors and homophonic ambiguities were considered separately

Although syntax analysis incorporated in a correction procedure appears to improve the output very little, what is transparent is that many corrections have been confirmed as being correct.

In the area of homophonic ambiguities, it is clear why some previous researchers chose to manually correct texts before attempting to perform syntax analysis on them. The combination of homophonic ambiguities and operator errors means that a parser should be presented with many alternatives from which to select the intended one, instead of the 2 or 3 presented to it in the present research. Again, syntax analysis was not as effective as it might have been because of the reasons a), b) and c) above.

11.5 SPEECH RECOGNITION SYSTEMS

It is clear from a surface level investigation that there are many similarities between the problems of direct speech input to a computer and those of Palantype transcription systems. It is almost certain that a deeper investigation will prove beneficial.

Palantype is a phonetic shorthand and the code is essentially a phonetic description of continuous speech. Therefore since the major part of the work has been done to get from Palantype code to English text, it is worth investigating whether it is possible to get from continuous speech to Palantype code (or something near). A feasibility study has been undertaken at Leicester Polytechnic by Guzy under the direction of Edmonds et al, (1980). A project of this complexity clearly requires more knowledge sources than the syntax used on this project, and the organisation of these writing system chosen to implement the parsing on the present project is a very powerful and versatile device. Not only does it provide the facility to do syntactic and semantic analysis, but it should also be possible to organise some of the other knowledge sources (e.g. phonological, articulatory, task dependent, phonetic) to interact with the parser when and where necessary.

11.6 A DICTIONARY BASED PALANTYPE TRANSCRIPTION SYSTEM OF THE FUTURE

I believe that in Palantype transcription systems of the future parsers will guide the transcription process and will have at their disposal many resources, only one of which will be the word reconstitution algorithm. I shall now describe an imaginary system called PALPARSE whose properties I believe reflect those of the next generation of dictionary based Palantype transcription systems.

PALPARSE is a Palantype transcription system which is capable of performing the following functions :-

1. Word boundary reconstitution

   PALPARSE incorporates the WORDFINDER transcription algorithm and is capable of generating all possible output strings for a given input sequence of chords.

2. Co-ordinates error correction tools

PALPARSE includes many error correction tools which are co-ordinated in such a way that their interaction provokes the update of the files which they reference. For example, consider the situation where the error metric presents a list of alternatives to the weighting routine, and the weighting routine orders that list based on error counts, etc., stored in a reference file. That same reference file can be updated by the error metric to say which of the elements in the list was actually accepted as the correct alternative, so that the next time the file is referenced it contains new information which may influence the correction process.

3. Copes with multiple errors

For every chord processed, PALPARSE stores a list of chords which are near, in terms of key difference, so that when errors are encountered PALPARSE has many more alternatives to submit to the error correction procedures. These lists are supplementary to those produced by the error metric. To cope with the situation where chords are missing for example, PALPARSE generates lists of chords containing all the possible chords which can link two adjacent chords in the input string. For example, if the operator intended to press chords C1 C2 and C3 but only pressed C1 and C3, PALPARSE can access a list which contains those chords which can link C1 and C3 in any tree. The way that this is achieved is that PALPARSE invokes a procedure which searches all trees (data structures) until it finds chord C1. If C1 is at level i, then level i+2 is searched for to see if it contains C3. If C3 is found, the chord at level i+1 is stored in the "linking" list.

4. Predicts the syntactic category of the next word in the input string

In the present research the error correction procedures produced a list of alternative corrections and then proceeded to eliminate them one by one until a correction was found that was syntactically correct. In PALPARSE the 'list' only contains those alternatives which are syntactically correct, and other knowledge sources help to determine which of the alternatives is the correct one.

Similarly with the resolution of homophonic ambiguities, PALPARSE more easily eliminates ambiguous alternatives simply by testing whether they are or are not in certain syntactic categories. The correction procedure only considers those ambiguous alternatives which are of a certain syntactic class(es).

5. Incorporates contextual knowledge by constructing and referencing knowledge frames about particular subject areas

In the present research many 'wrong' corrections were made simply because a correction procedure 'knew' nothing about the meaning of the passage being transcribed. An example of this can be seen in fig.2.3 of appendix C where the error in CH/IL T/EN has been corrected to "chill ten" because it was accepted as being syntactically correct. However, the whole passage is about children and child abuse and these words appear frequently throughout the passage. It seems obvious that in a passage about children the error in CH/IL T/EN should be corrected to "children" and not "chill ten".

PALPARSE has access to knowledge frames about the subject concerned in the passage being transcribed. These frames are dynamic so that they increase their 'knowledge' as the transcription progresses.

6. Learns from previously corrected errors

The idea of having 'idiosyncratic' dictionaries or dictionaries suited to the particular idiosyncracies of the operator currently using the system will certainly figure strongly in systems of the future. The way in which it is used in PALPARSE is as follows. PALPARSE contains a general or 'standard' dictionary used by all operators with a supplementary 'reference file' which not only includes things like operators short forms, but also a list of transforms which have to be made before the operators output can be presented to the 'standard' dictionary. These transforms include operators idiosyncracies and differences from the 'standard' dictionary in things like choice of syllable boundaries. The reference file is again dynamic as the operators idiosyncracies may vary even from day to day. PALPARSE is not only able to interpret the reference file but is also able to update it during the transcription process.

7. Incorporates Acoustic-Phonetic Information.

PALPARSE has access to information concerning the relationship between phonemes and chords. When correcting errors for example, such information is useful in determining whether the phoneme which ends one chord can be followed by the phoneme which starts the next chord in the input string.

It is worth mentioning that SYNICS already has the facilities to enable PALPARSE to be implemented, and for those researchers in the field whose goal is 'real time' transcription into high quality text output it is worth investigating a system where several copies of SYNICS are running in a parallel processing environment !!!

BIBLIOGRAPHY

Alberga,C.N., String Similarity and Misspellings, Communications of the ACM,Vol.10,5,1967.

Arnott,J.L.,Newell,A.F. and Downton,A.C., A Comparison of Palantype and Stenograph for use in a Speech Transcription Aid for the Deaf, Journal of Biomed. Eng.,Vol.1,pp.201-210,1979.

Beard,I., The Palantype Method of Stenotyping, The Palantype Organisation, London, 1978.

Booth,A.W. and Barnden,M.S., Voice Input to English Text Output, International Journal of Man-Machine Studies,1979.

Bratley,P. and Dakin,D.J., A Limited Dictionary for Syntactic Analysis. 173-181.

Chomsky,N., Aspects of the Theory of Syntax, Cambridge, Mass.,MIT Press,1965.

Chou,S.M. and Fu,K.S., Transition Network Grammars for Syntactic Pattern Recognition,

Conrad,R. and Longman,D.J.A., Standard Typewriter Versus Chord Keyboard - an Experimental Comparison, Ergonomics,Vol.8,77-88,1965.

Craft,J.L.,Goldman,E.H. and Strohm,W.B., A Table-lookup Machine for Processing of Natural Languages, I.B.M. Journal of Research and Development, vol.5, July 1961.

De Mori,R., Recent Advances in Automatic Speech Recognition, Journal of Signal Processing, Vol.1,2,pp.95-123,April 1979.

De Mori,R. and Laface,P., Use of Fuzzy Algorithms for Phonetic and Phonemic Labelling of Continuous Speech, IEEE Trans. on Pattern Analysis Machine Intelligence, Vol.PAMI-3, no.2,pp136-148,March 1980.

Downton,A.C. and Newell,A.F., An Assessment of Palantype Transcription as an Aid for the Deaf, International Journal of Man-Machine Studies,1979.

Downton,A.C.,Newell,A.F. and Arnott,J.L., Operator Error Performance and Keyboard Evaluation in Palantype Machine Shorthand, Applied Ergonomics,1980.

Edmonds,E.A. and Guest,S.P., SYNICS - Fortran Subroutine Package for Translation, Man-Computer Interaction Research Group, Report no.6, Leicester Polytechnic, 1978.

Edmonds,E.A.,Hashim,A.A. and Connolly,T.H., Case for Support: A Feasibility Study into a System for Direct Speech Input into Computers, Leicester Polytechnic,1980.

Flanagan,J.L., Computers that Talk and Listen: Man-Machine Communications by Voice, Proc.IEEE,Vol.64,4,1976.

Fox,R.G.,Ludeman,C.P. and Peck,R.H., Speech Translation Machine for the Deaf, I.B.M. Technical Disclosure Bulletin (USA),17,12,pp3640-3642,1975.

Galli,E.J., The Stenowriter - A System for the Lexical Processing of Stenotypy, IRE Transactions on Electronic Computers,11,2,pp.187-199,1962.

Gleason,H.A., Linguistics and English Grammar, Holt,Rinehart and Winston, Inc.,1965.

Goldman,N.M., Sentence Paraphrasing from a Conceptual Base, Communications of the ACM,Vol.18,2,1975.

Guest,S.P., Software Tools for Dialogue Design, Int.J.Man-Machine Studies,(In press,1982).

Hawkins,W.R. and Robinson,R.N., The Development and Use of an Electric Keyboard for Television Subtitling by Palantype. Int.J.Man-Machine Studies,1979,11,701-710.

Hayward,G., Palantype in the Office, Hearing,pp.104-108,June 1978.

Hayward,G., A Profoundly Deaf Businessman's Views of the Palantype Speech Transcription System. Int.J.Man-Machine Studies,1979,11,711-715.

Kaplan,R.M., Augmented Transiton Networks as Psychological Models of Sentence Comprehension, Artificial Intelligence,3,1972.

Kuno,S. and Oettinger,A.G., Multiple Path Syntactic Analyser, Information Processing, North Holland Publishing Company,Amst.1962.

McDaniel,J.,Day,A.M.,Price,W.L.,Szanser,A.J.,Whelan,S. and Yates,D.M., Translation of Russian Scientific Texts into English by Computer - a Final Report, Autonomics Division, NPL, Auto 35,1967.

McMaster,I.,Sampson,J.R. and King,J.E., Computer Acquisiton of Natural Language: a Review and Prospectus, Int.J.Man-Machine Studies,8,367-396,1976.

Miller,P.L., A Locally Organised Parser for Spoken Input, Communications of the ACM,Vol.17,11,1974.

Newell,A.F., Can Speech Recognition Machines Help the Deaf, Teacher of the Deaf 72,pp.367-374,1974.

Newell,A.F., The Palantype Unit -its History and Progress to date, Hearing,99-104,May/June,1978.

Newell,A.F. and King,J.A.F., Speech Translation Systems for the Hearing Impaired, Med. and Biol. Eng. and Computing,15,pp.558-563,1977.

Newell,A.F. and Downton,A.C., Visual Display of Speech Transcription for the Deaf, I.E.E. Conference on Visual Displays, Lancaster University,1977.

Newell,A.F. and Hutt,P.R., Subtitling "Live" Television Programmes for the Hearing Impaired, Int.J.Man-Machine Studies,1979,11,693-699.

Newell,A.F. and Hutt,P.R., The Evolution of a Teletext Subtitle System for Hearing Impaired Viewers, Proc.Int.Broadcasting Convention,308-311,1978.

Newitt,J.W.,Boutross,A.J.,Odarachenko,A. & Wong,K.F., An Evaluation of Computer-assisted Stenotype Transcription, I.B.M. Research Report RC1918,1967.

Newitt,J.W. and Odarachenko.A., A Structure for Real-time Stenotype Transcription, I.B.M. Systems Journal,9,(1),pp.24-35,1970.

Nishida,F.,Takamatsu,S. and Fujita,Y., A Syntactic Analysis of English Sentences and the Information Extraction.

Parkinson,R.C.,Colby,K.M. and Faught,W.S., Conversational Language Comprehension Using Integrated Pattern Matching and Parsing. Artificial Intelligence,9,111-134,1977.

Partridge,D., A Syntactic View of Semantic Networks, Int.J.Man-Machine Studies,10,113-119,1978.

Passingham,R.E., Palantype Chord Data Analysis, M.Sc.Dissertation, Southampton Univ.,1980.

Pereira,F.C.N. and Warran,D.M.D., Definate Clause Grammars for Language Analysis - A Survey of the Formalism and a Comparison with Augmented Transition Networks, Artificial Intelligence,13,1980.

Plath,W.J., Multiple-Path Analysis and Automatic Translation, Machine Translation, North Holland Publishing Company,Amst.1967.

Price,W.L., Palantype Transcription by Computer - a Final Report, N.P.L. Com. Sci. 45, 1971.

Price,W.L., The Viability of Computer Transcription of Machine Shorthand, Proc. IEE Conference on Man-Computer Interaction,Sept.1970.

Quirk,R.,Greenbaum,S.,Leech,G. and Svartvik,J., A Grammar of Contemporary English, Seminar Press,1972.

Ratz,H.C. and Ritchie,D.K., Operator Performance on a Chord Keyboard, Journal of Applied Psychology,Vol.45,5,303-308,1961.

Reddy,D., Speech Recognition by Machine : A Review, Proc. IEEE, Vol.64, pp505-531, April 1976.

Rovner,P.,Nash-Webber,B. and Woods,W.A., Control Concepts in a Speech Understanding System, IEEE Transactions on Acoustics,Speech and Signal Processing,1,1975.

Sagar,N. and Grishman,R., The Restriction Language for Computer Grammars of Natural Language, Communications of the ACM,Vol.18,7,1975.

Salton,G., The Automatic Transcription of Machine Shorthand, Proceedings of the Eastern Joint Computer Conference,17,pp.148-159,1959.

Schwarz,R.M., Steps Toward a Model of Linguistic Performance: a Preliminary Sketch, Mechanical Translation,10,1967.

Seibel,R., Performance on a Five-Finger Chord Keyboard, Journal of Applied Psychology,Vol.46,3,165-169,1962.

Simmons,R. and Slocum,J., Generating English Discourse from Semantic Networks, Communications of the ACM, Vol.15,10,Oct.1972.

Stenograph, Stenograph's Computer Compatible Touch Shorthand - Books 1 & 2 Plus Supplementary Lists, Stenograph,Stokie,Illinois.

Szanser,A.J., Linguistics in Automatic Machine Shorthand Transcription, The Incorporated Linguist,8,(2),pp30-33,1969.

Szanser,A.J., Error-correcting Methods in Natural Language Processing, Information Processing 68, Edited by A.J.H.Morrell,Vol.II, pp.1412-1416,1969.

Szanser,A.J., Automatic Error Correction in Natural Language, Information Storage and Retrieval,5,pp.169-174,1970.

Szanser,A.J., Resolution of Ambiguities by Contextual Word Repetition, Review of Applied Linguistics,7,pp.49-56,(Louvain)1970.

Szanser,A.J., Resolution of Phonetic Ambiguities, The Incorporated Linguist,10,(4),pp.112-115,1971.

Szanser,A.J., Resolution of Ambiguities in Natural Texts Processed by Computer, Assoc.Lit. and Lang. Computing Bulletin,5,pp.259-265,1977.

Thomas,L.A. and Hawkins, W.R., The Economic Preparation of Teletext Subtitles, Proc. IBC,1980.

Thorne,J.P.,Bratley,P. and Dewar,H., The Syntactic Analysis of English by Machine, Machine Intelligence 3, edited by D.Michie,Edin. Univ.,1968.

Thorne,J.P.,Dewar,H.,Whitfield,H. and Bratley,P., A Model for the Perception of Syntactic Structure, Colloque International sur L'informatique,Toulouse.

Underwood,M.J., Machines that Understand Speech, The Radio and Electronic Engineer,Vol.47,819,368-376,1977.

Wilks,Y., An Intelligent Analyser and Understander of English, Communications of the ACM,Vol.18,5,1975.

Winograd,T., Understanding Natural Language, Edinburgh University Press,1972.

Wood,C.B.B.,Hawkins,W.R. and Wright,D.T., Subtitles on the Ceefax Service, Proc.Int.Broadcasting Convention,1978.

Woods,W.A., Transition Network Grammars for Natural Language Analysis, Communications of the ACM,13,Oct.1970.

Appendix A

EXAMPLES OF OUTPUT FROM THE PARSER

STRING: She is in London

ANALYSIS: ((((She :PRONOUN)):SUBJECT)(is :VERBPHRASE)(((in :PREP) (London :PROP NOUN):PREP PHRASE):PLACE ADVERBIAL):DECL CLAUSE):SENTENCE)

STRING: She is in London now

ANALYSIS: ((((She :PRONOUN)):SUBJECT)(is :VERBPHRASE)(((in :PREP) (London :PROP NOUN):PREP PHRASE):PLACE ADVERBIAL)(now :TIME ADVERBIAL):DECL CLAUSE): SENTENCE)

STRING: She is a student

ANALYSIS: ((((She :PRONOUN)):SUBJECT)(is :VERBPHRASE)((a :DET) (student :NOUN):SUBJECT COMPLEMENT):DECL CLAUSE):SENTENCE)

STRING: She is a student in London

ANALYSIS: ((((She :PRONOUN)):SUBJECT)(is :VERBPHRASE)((a :DET) (student :NOUN):SUBJECT COMPLEMENT)(((in :PREP)(London : PROP NOUN):PREP PHRASE):PLACE ADVERBIAL):DECL CLAUSE): SENTENCE)

STRING: She is a student in London now

ANALYSIS: (((((She :PRONOUN)):SUBJECT)(is :VERBPHRASE)((a :DET) (student :NOUN):SUBJECT COMPLEMENT)(((in :PREP)(London : PROP NOUN):PREP PHRASE):PLACE ADVERBIAL)(now :TIME ADVERBIAL):DECL CLAUSE):SENTENCE)

STRING: John heard the explosion

ANALYSIS: (((((John :PROP NOUN)):SUBJECT)(heard :VERBPHRASE)((the :DET) (explosion :NOUN):DIRECT OBJECT):DECL CLAUSE):SENTENCE)

STRING: John heard the explosion from his office

ANALYSIS: (((((John :PROP NOUN)):SUBJECT)(heard :VERBPHRASE)((the :DET) (explosion :NOUN):DIRECT OBJECT)((from :PREP)(his :PRONOUN) (office :NOUN):PREP PHRASE):PLACE ADVERBIAL):DECL CLAUSE): SENTENCE)

STRING: John heard the explosion from his office when he was locking the door

ANALYSIS: (((((John :PROP NOUN)):SUBJECT)(heard :VERBPHRASE)((the :DET) (explosion :NOUN):DIRECT OBJECT)((from :PREP)(his :PRONOUN) (office :NOUN):PREP PHRASE):PLACE ADVERBIAL)((when ((((he:

PRONOUN)):SUBJECT)((was :AUX) locking :VERBPHRASE)((the:DET)
(door :NOUN):DIRECT OBJECT):DECL CLAUSE):ADVERB CLAUSE):
TIME ADVERBIAL):DECL CLAUSE):SENTENCE)

STRING: He had given the girl an apple

ANALYSIS: (((((He:PRONOUN)):SUBJECT)(had :AUX)given :VERBPHRASE)((the
:DET)(girl :NOUN):INDIRECT OBJECT)((an :DET)
(apple :NOUN):DIRECT OBJECT): DECL CLAUSE):SENTENCE)

STRING: The girl had been given an apple by him

ANALYSIS: (((((The :DET)(girl :NOUN)):SUBJECT)((had :AUX)
been given :VERBPHRASE)((an :DET)(apple :NOUN)
:DIRECT OBJECT)((by (him :PRONOUN):PREP PHRASE)
:PLACE ADVERBIAL): DECL CLAUSE):SENTENCE)

STRING: The girl is now a student at a large polytechnic

ANALYSIS: (((((The :DET)(girl :NOUN)):SUBJECT)(is :VERBPHRASE)
now :ADVERBIAL)((a :DET)(student :NOUN):SUBJECT
COMPLEMENT)((at (a :DET)(large :ADJECTIVE)(
polytechnic :NOUN):PREP PHRASE): PLACE ADVERBIAL):
DECL CLAUSE):SENTENCE)

STRING: That she answered the question correctly pleased him enormously

ANALYSIS: (((That (((she :PRONOUN)):SUBJECT)(answered :VERBPHRASE)(((the :DET)(question :NOUN):DIRECT OBJECT):DIRECT OBJECT)(correctly :ADVERBIAL): DECL CLAUSE):SUBJECT)((pleased :VERBPHRASE)(((him :PRONOUN):DIRECT OBJECT):DIRECT OBJECT)(enormously :ADVERBIAL): DECL CLAUSE): SENTENCE)

STRING: Last week I came here ten times

ANALYSIS: ((((Last :ORDINAL)(week :NOUN):TIME ADVERBIAL)(((I :PRONOUN)):SUBJECT)(came :VERBPHRASE)(here :PLACE ADVERBIAL)(ten :CARDINAL(times :NOUN):TIME ADVERBIAL): DECL CLAUSE):SENTENCE)

STRING: Universities became famous

ANALYSIS: (((((Universities :NOUN)):SUBJECT)(became :VERBPHRASE)((famous :ADJECTIVE):SUBJECT COMLEMENT): DECL CLAUSE) :SENTENCE)

STRING: Universities gradually became famous

ANALYSIS: ((((Universities :NOUN)):SUBJECT)(gradually :ADVERBIAL)(became :VERBPHRASE)((famous :ADJECTIVE):SUBJECT COMPLEMENT): DECL CLAUSE):SENTENCE)

STRING: Universities became famous in Europe

ANALYSIS: ((((Universities :NOUN)):SUBJECT)(became :VERBPHRASE)((famous :ADJECTIVE):SUBJECT COMPLEMENT)((in (Europe :NOUN) :PREP PHRASE):PLACE ADVERBIAL): DECL CLAUSE):SENTENCE)

STRING: They ate the meat hungrily in their hut that night

ANALYSIS: ((((They :PRONOUN)):SUBJECT)(ate :VERBPHRASE)((the :DET)(meat :NOUN):DIRECT OBJECT)(hungrily :ADVERBIAL)((in (their :PRONOUN)(hut :NOUN):PREP PHRASE):PLACE ADVERBIAL)((that :DET)(night :NOUN):TIME ADVERBIAL): DECL CLAUSE):SENTENCE)

STRING: He offered her some chocolates politely outside the hall before the concert

ANALYSIS: ((((He :PRONOUN)):SUBJECT)(offered :VERBPHRASE)((her :PRONOUN)((some :ADVECTIVE)chocolates :NOUN):DIRECT OBJECT) (politely :ADVERBIAL)((outside (the :DET)(hall : NOUN):PREP PHRASE):PLACE ADVERBIAL)((before (the :DET)(concert :NOUN):PREP PHRASE):TIME ADVERBIAL): DECLARITIVE CLAUSE):SENTENCE)

STRING: The train had arrived quietly at the station before we noticed it

ANALYSIS: ((((The :DET)(train :NOUN)):SUBJECT)((had :AUX) arrived :VERBPHRASE)(quietly :ADVERBIAL)((at (the : DET)(station :NOUN):PREP PHRASE):PLACE ADVERBIAL)((before (we :PRONOUN):PREP PHRASE): TIME ADVERBIAL): DECL CLAUSE)((noticed :VERBPHRASE) ((it :PRONOUN):DIRECT OBJECT): DECL CLAUSE): SENTENCE)

Appendix B

IDENTIFYING THE SOURCE OF ERROR

The points in the transcription program at which it becomes apparent that an error is present in the input string, have been classified and numbered. This appendix describes the classification of each error point and also the corresponding action taken to identify the source of the error.

Error Point E1
----

is arrived at when the following conditions are satisfied:-

1. Only the first chord of the input string has been scanned

2. The chord is not an initial chord

Action

The chord is assumed to be erroneous and the error metric and 'weighting' routine are invoked to determine the chord most likely intended by the operator.

Error Point E2
----

is arrived at when the following conditions are satisfied:-

1. Only two chords of the input string have been processed since the last English word was output.

2. The latter of these chords is not a successor in the tree of the former.

3. The former is an initial chord but is not a word in its own right.

Action

The latter of the two chords is assumed to be erroneous in that it should have found a match in the list of chords which can succeed the first chord. If attempting to correct this error is not successful, then the first chord is assumed to be erroneous in that it should have found a different match in the list of initial chords.

Error Point E3
----

is arrived at when the following conditions are satisfied:-

1. Only two chords of the input string have been processed since the last English word was output.

2. The latter of these chords is not a successor in the tree of the former.

3. The former is an initial chord and is a word in its own right.

4. The latter chord is not an initial chord.

For example, this point would be arrived at if the operator was intending to type " C/AN T+/I " (candy), but typed " C/AN +/I ".

Action

Actions taken to identify the source of error at error point E3 are as follows.As either or both of the two chords could be erroneous, the correction procedure first assumes one of them to be in error and attempts to correct it. If this is not successful, the other chord is assumed to be in error and again attempts are made to correct it. If this is also unsuccessful, then special correction techniques are required since both chords are in error or there is an added or deleted chord situation. At error point E3, since the first chord is not only an initial chord, but is a word in its own right, the latter of the two chords is more likely to be in error . Therefore, in the correction procedure described in the next section, it is the first to be assumed erroneous.

Error Point E4
----

is arrived at when the following conditions are satisfied:-

1. Only two chords of the input string have been processed since the last English word was output.

2. The latter of these chords cannot be a successor in the tree of the former, since the former does not start any other word apart from itself, i.e. a word in its own right.

3. The latter is not an initial chord.

Action

The latter of the two chords is assumed to be erroneous in that it should have found a match in the list of initial chords. If attempting to correct this error is not successful, then the first chord is assumed to be erroneous in that it should have found a different match in the list of initial chords.

Error point E5
----

is arrived at when the following conditions are satisfied.

1. More than two chords have been processed since the last English word was output.

2. The latest chord to be scanned is not a successor to the previous chord in the input string.

3. There has been no chord processed since the last English word was output that has a pointer to an English word.

For example, this point would be arrived at if the operator was intending to type "S/I C+N/IF C/NS" (significance), but typed "S/I C+N/IF C/N".

Action

Actions taken to identify the source of error at error point E5 are as follows. Any of the chords processed since the last English word was output, could be erroneous. However, since the latest chord to be scanned was the first to "break the chain", it is the first to be assumed erroneous. If it cannot be satisfactorily corrected, then the error correction procedure scans the chords until it finds one that is the start of a word, and then assumes the chord immediately preceding it to be erroneous. If this does not produce a satisfactory correction, the correction procedure looks for another initial chord, other than the one that immediately follows the last English word. If there are no more initial chords then special correction techniques are required since there are multiple errors or added or deleted chords.

Error point E6
----

is arrived at when the following conditions are satisfied.

1. More than two chords have been processed since the last English word was output.

2. The latest chord to be scanned is not a successor to the previous chord in the input string.

3. There is no subset of chords that form an English word that is followed by a successor chain which includes the latest chord.

4. The penultimate chord scanned does not have a pointer to an English word.

For example, this point would be arrived at if the operator was intending to type "S/EN TR/I F/EU C+/L" (centrifugal), but typed "S/EN TR/I F/EU C/L".

Action

Actions taken to identify the source of error at error point E6 are as follows. Any of the chords processed since the last English word was output, could be erroneous. However, since the latest chord to be scanned was the first to "break the chain", it is the first to be assumed erroneous. If it cannot be satisfactorily corrected, then the error correction procedure assumes the chord immediately preceding the latest chord to be the one to be erroneous. If this does not produce a satisfactory correction, the correction procedure takes each chord in turn as far back as the last word boundary and subjects it to the correction procedure. Note only one chord at a time is assumed to be erroneous.

Error point E7

----

is arrived at when the following conditions are satisfied.

1. More than two chords have been processed since the last English word was output.

2. The latest chord to be scanned cannot be a successor to the previous chord in the input string, since the previous chord does not have a pointer to a list of chords which can succeed it.

3. There is no subset of chords that form an English word that is followed by a successor chain which includes the latest chord.

4. The latest chord is not an initial chord.

For example, this point would be arrived at if the operator was intending to type "SH/OU RL/I /TH" (surely the), but typed "SH/OU RL/I C/H".

Action

Actions taken to identify the source of error at error point E7 are as follows. The latest chord to be scanned is assumed to be in error in that it should have found a match in the initial chord list. If it cannot be satisfactorily corrected, then the previous chord in the input string is assumed to be erroneous in that it should have been a chord with a pointer to a list of successors (in which the latest chord has a match).

Error point E8

is arrived at when the following conditions are satisfied.

1. More than two chords have been processed since the last English word was output.

2. The latest chord to be scanned is not a successor to the previous chord in the input string.

3. There is no subset of chords that form an English word that is followed by a successor chain which includes the latest chord.

4. The penultimate chord has a pointer to an English word.

5. The latest chord is not an initial chord.

For example, this point would be arrived at if the operator was intending to type "S/EN TR/I /ON" (sentry on), but typed "S/EN TR/I C/N".

Action

Actions taken to identify the source of error at error point E8 are as follows. The latest chord to be scanned is assumed to be in error in that it should have found a match in the initial chord list. If it cannot be satisfactorily corrected, then the previous chord in the input string is assumed to be erroneous. If this does not produce a satisfactory correction, the correction procedure takes each chord in turn as far back as the word boundary before the one indicated by the penultimate chord, and subjects it to the correction procedure. Note only one chord at a time is assumed to be erroneous.

Appendix C

COMPARISONS OF UNCORRECTED AND CORRECTED TEXT

In this appendix successful corrections are underscored with an unbroken line where failures are underscored with a broken line.

The decisions that have to be made are basically political decisions + but we would hope that the hearing impaired audience have assay in these decisions + an what we are trying T do is to ascertain what the hearing impaired really want rather than try to impose other peoples views on what they ought to want . . . Are there any PRO P+LEMS with subtitling . Sure LI all we have T do is to put down on a screen more or less as we are T+OUIN+ HI.RTH words that are P+JI.N+ said . But P+E for we dismiss this AUS not a PRO P+LEM lets look at some of the RE.N +JS of the written word which have to be coped with when you are subtitling . These are the sorts of programmes that one might get on T V . Ant that is the RE.N +J of speech that you also get on T V . We once tried to subtitle afoot ball match . It was own LI when fee T+I+ this that we REA lie ST+ the commentary had no sense in it at all . ANT+ in fact one begins to REA lice that this is true of a number of inter views as well . Drama + no PRO P+LEM sure LI + there is just the script + until you RE MEM P+RTH time THATH AC tor SPENFT+S deciding how to say a particular line . There are as many ways of SEIN+ to be or not to be THAITS the CFES TJUN as there are people who have acted that PTFIC roll + ant on a subtitle you have lost all that . So in a T+I HN of course + if you look at teleprompter TIN+ + ANT+ news readers + in fact these people talk very very quick LI + ant they probably talk faster THAINT is REA song P+L to expect people to read the subtitle ANT+ look at T pictures .

Fig. C1.1 Output produced by a transcription program incorporating no error correction whatsoever

The decisions that have to be made are basically political decisions , but we would hope that the hearing impaired audience have assay in these decisions , an what we are trying the do is to ascertain what the hearing impaired really want rather than try to impose other peoples views on what they ought to want . . . Are there any problems with subtitling . Surely all we have the do is to put down on a screen more or less as we are doing here the words that are being said . But beveller we dismiss this souse not a problem lets look at some of the ranges of the written word which have to be coped with when you are subtitling . These are the sorts of programmes that one might get on T V . Ant that is the range of speech that you also get on T V . We once tried to subtitle afoot ball match . It was only when fee dee this that we realised the commentary had no sense in it at all . And in fact one begins to really that this is true of a number of inter views as well . Drama , no problem surely , there is just the script , until you remember the time that the actor spends deciding how to say a particular line . There are as many ways of saying to be or not to be that is the question as there are people who have acted that particular roll , ant on a subtitle you have lost all that . So in addition of course , if you look at teleprompter tinned , and news readers , in fact these people talk very very quickly , ant they probably talk faster than it is reason blest to expect people to read the subtitle and look at the pictures .

Fig. C1.2 Output produced by a transcription program incorporating preprocessing corrections and the 'History and Future' correction routine

The decisions that have to be made are basically political decisions , but we would hope that the hearing impaired audience have assay in these decisions , an what we are trying the do is to ascertain what the hearing impaired really want rather than try to impose other peoples views on what they ought to want . . . Are there any problems with subtitling . Surely all we have the do is to put down on a screen more or less as we are doing here the words that are being said . But beveller we dismiss this souse not a problem lets look at some of the ranges of the written word which have to be coped with when you are subtitling . These are the sorts of programmes that one might get on T V . Ant that is the range of speech that you also get on T V . We once tried to subtitle afoot ball match . It was only when fee dee this that we realised the commentary had no sense in it at all . And in fact one begins to realise that this is true of a number of inter views as well . Drama , no problem surely , there is just the script , until you remember the time that the actor spends deciding how to say a particular line . There are as many ways of saying to be or not to be that is the question as there are people who have acted that particular roll , ant on a subtitle you have lost all that . So in addition of course , if you look at teleprompter tinned , and news readers , in fact these people talk very very quickly , ant they probably talk faster than it is reason blest to expect people to read the subtitle and look at the pictures .

Fig. C1.3 Output produced by a transcription program incorporating preprocessing corrections, the 'History and Future' correction routine and syntax analysis

Lynn pawl there . . . Now in our medicine MA TRS series to-day we are C+OIN+ to look at T emotive sup +JECT of baby battering + every week in PRI TN two chill TN die at their PE. RNTS hands many more are serious LI in +JRT+S + child abuse in MFOLF+S almost eight thousand SCHIL TRN a year + occurs in every social CLJAS + can PI. physical secretary SEUL or emotional + an its no new phenomenom + its been known in every so yeti HRAUT hiss TRI an at STAIMS has PI.N con SH+RA +L worse than to-day . Yet until now there has PI.N no RLAI. P+L way of predicting it or TREA tinned PE. RNTS . To-morrow in lung ton eight fifty dell C+TS + X PRTS from all over T world will a TEANT international congress on child abuse an NEC+ LECT to pull their +NLJ on this complex sup +JECT . On the same day this book will be PLUI SHT + its CO.LC+ child abuse an its by the couple who fur T T+ES CTRAUIT the + battered baby syndrome + Ruth an Henry Kemp who have worked for ever ULCFT over TFEN TA years in a mare CAS national centre for PRE MFEN SHN ant treat MNT of CHAIL abuse an now feel that not own LI can a P+EU SHIF+ PE. TRNTS PI. predicted but they CPAN PI. helped. The big ST needs though is a full UN T+UR standing of what PE. RNT a going through a the time . FRINS I. MFN a perfect LI nor mall mother SU FRINM+ CPOE STH NE. TAL depression can find herself showing frightening SIMP TONTS . Dane +JR SIC+ NLS that CPAN PI. treated successfully if mother family or T+R SPHOTS THEINM time . . .

Fig. C2.1 Output produced by a transcription program incorporating no error correction whatsoever

Lynn pawl there . . . Now in our medicine masters series to-day we are going to look at the emotive subject of baby battering , every week in Britain two chill ten die at their parents hands many more are seriously injured , child abuse involves almost eight thousand she'll ten a year , occurs in every social glass , can be physical sexual or emotional , an its no new phenomenom , its been known in every so yeti shout history an at sties has been considerable worse than to-day . Yet until now there has been no pliable way of predicting it or treating parents . To-morrow in lung ton eight fifty dell gets , X pets from all over the world will attend international congress on child abuse an neglect to pull their knowledge on this complex subject . On the same day this book will be PLUI SHT , its caulk child abuse an its by the couple who fur the DES CTRAUIT the , battered baby syndrome , Ruth an Henry Kemp who have worked for over twenty years in a mare gas national centre for prevention ant treat meant of chill abuse an now feel that not only can abusive patents be predicted but they calne be helped. The big stet needs though is a full understand Dane of what parent a going through a the time . FRINS Avon a perfectly nor mall mother suffering CPOE STH natal depression can find herself showing frightening SIMP TONTS . Dane jar signals that calne be treated successfully if mother family or DR SPHOTS them in time . . .

Fig. C2.2 Output produced by a transcription program incorporating preprocessing corrections and the 'History and Future' correction routine

Lynn pawl there . . . Now in our medicine masters series to-day we are going to look at the emotive subject of baby battering , every week in Britain two chill ten die at their parents hands many more are seriously injured , child abuse involves almost eight thousand children a year , occurs in every social glass , can be physical sexual or emotional , an its no new phenomenom , its been known in every so yeti shout history an at sties has been considerable worse than to-day . Yet until now there has been no pliable way of predicting it or treating parents . To-morrow in lung ton eight fifty dell gets , experts from all over the world will attend international congress on child abuse an neglect to pull their knowledge on this complex subject . On the same day this book will be PLUI SHT , its caulk child abuse an its by the couple who fur the DES CTRAUIT the , battered baby syndrome , Ruth an Henry Kemp who have worked for over twenty years in a mare gas national centre for prevention ant treat meant of chill abuse an now feel that not only can abusive patents be predicted but they calne be helped. The big stet needs though is a full understanding of what parent a going through a the time . FRINS Avon a perfectly nor mall mother suffering CPOE STH natal depression can find herself showing frightening SIMP TONTS . Dane jar signals that calne be treated successfully if mother family or DR SPHOTS them in time . . .

Fig. C2.3 Output produced by a transcription program incorporating preprocessing corrections, the 'History and Future' correction routine and syntax analysis

Thank you very much indeed . . . It heartening ANT+ REA shoe ring to know that ECSH PRTS are at last coming to grips fit the PRO SP+LEMS of child abuse . Now to David see moor for the PRO P+LEMS of inflation . . . We have been in MFE. T+T+ by a Russian T sixty two battle tank . It almost looks OAIC the real THIN+ PUT it would not be much use to the Russians because I could knock this tank outwit a hat pin . Its an inflatable ANT+ it ways about as muck as I do . Its part of a SPTEC TA CEULR collection of inflate P+LS ANT+ P+A noons hat we've got a SEM PT+LT+ on the back lawn to-day . Each one has its own purpose. FRECS + The Russian T sixty two battle tank is used + by B A O R for RE CNI SHN purposes + they don't use them for tar get practice because EACSH is worth about two thousand . All these were made by a CMPNI calked airborne INT+ STRIS + an the sales EC secratary is mall COLM TI P+L . How did you get into the business of making inflatable tanks . . . At a la main when Montgomery had insufficient tanks he felt he wanted more + an he COLCT+ U PONL us OU through the you joule SCHANLS to make some tanks to full the enemy . . Did they . JESH indeed +J we made several HUNLT+ of them + hay FRUR put in several emplacements FITH remarkable success . . What does the army use tanks like this for the day . . Its T+RIF CULT to get hardware FRONMH other side to train on + we need TJO know what their E quip NNT looks like + MFI Jew LI + ANT+ this is the cheap ST way of ensuring that tank crews + fighter air crews ant forward air controllers know what the other sides hardware looks like . .

Fig. C3.1 Output produced by a transcription program incorporating no error correction whatsoever

Thank you very much indeed . . . It heartening and reassuring to know that aches pets are at last coming to grips fit the problems of child abuse . Now to David see moor for the problems of inflation . . . We have been in vended by a Russian the sixty two battle tank . It almost looks soak the real thing but it would not be much use to the Russians because I could knock this tank outwit a hat pin . Its an inflatable and it ways about as muck as I do . Its part of a steak TA CEULR collection of in fables and bassoons hat we've got assembled on the back lawn to-day . Each one has its own purpose. FRECS , The Russian the sixty two battle tank is used , by B A O R for technician purposes , they don't use them for tar get practice because each is worth about two thousand . All these were made by a many calked airborne IND STRIS , an the sales excesses is mall calm tipples . How did you get into the business of making inflatable tanks . . . At a la main when Montgomery had insufficient tanks he felt he wanted more , an he calked U pawl us awe through the you joule channels to make some tanks to full the enemy . . Did they . JESH indeed Z we made several hurled of them , hay fryer put in several emplacements with remarkable success . . What does the army use tanks like this for the day . . Its difficult to get hardware from the other side to train on , we need to know what their equip net looks like , Visually , and this is the cheap stet way of ensuring that tank crews , fighter air crews ant forward air controllers know what the other sides hardware looks like . .

Fig. C3.2 Output produced by a transcription program incorporating preprocessing corrections and the 'History and Future' correction routine

Thank you very much indeed . . . It heartening and reassuring to know that aches pets are at last coming to grips fit the problems of child abuse . Now to David see moor for the problems of inflation . . . We have been invaded by a Russian the sixty two battle tank . It almost looks like the real thing but it would not be much use to the Russians because I could knock this tank outwit a hat pin . Its an inflatable and it ways about as muck as I do . Its part of a steak TA CEULR collection of in fables and bassoons hat we've got assembled on the back lawn to-day . Each one has its own purpose. FRECS , The Russian the sixty two battle tank is used , by B A O R for technician purposes , they don't use them for tar get practice because each is worth about two thousand . All these were made by a many calked airborne IND STRIS , an the sales excesses is mall calm tipples . How did you get into the business of making inflatable tanks . . . At a la main when Montgomery had insufficient tanks he felt he wanted more , an he calked U pawl us awe through the you joule channels to make some tanks to full the enemy . . Did they . JESH indeed Z we made several hundred of them , hay fryer put in several emplacements with remarkable success . . What does the army use tanks like this for the day . . Its difficult to get hardware from the other side to train on , we need to know what their equip net looks like , Visually , and this is the cheap stet way of ensuring that tank crews , fighter air crews ant forward air controllers know what the other sides hardware looks like . .

Fig. C3.3 Output produced by a transcription program incorporating preprocessing corrections, the 'History and Future' correction routine and syntax analysis

Appendix D

EXAMPLES OF HOMOPHONIC AMBIGUITY RESOLUTION

In this appendix successful resolutions are underscored with an unbroken line, incorrect resolutions are underscored with a broken line, situations where the parser failed to produce a valid parse from any of the alternatives and so chose the one occurrring most frequently in everyday use are underscored with a line of x's, and situations where the parser produced a successful parse for all alternatives and so again chose the one occurring most frequently in everyday use, are underscored with a dotted line.

The applause shows how much your/you're speech ant the way you HAUF+ answered CFES TJUNS has P+AI.N PRI. SHJE. TT+ . . Thank you very muck indeed . . When Amman start in/inn politics ant early E. +J the great/grate CFES TJUN mark/Mark is whether he HATSH STEIN+ PAUR to remain in/inn such an exacting +IN tress TIN+ ant EC SI. UL exciting PRO FE SHN + many HAIN bright LI for P+FRAI.F mow MANT then disappear FRMTH seen/scene NEFR+ to REA PEAR . NEFR+ with nor/Nore man ST JON STI. MFAS + he first con TES TAT+ CON STI tune SI in nineteen fifty one/won at T E. +J of twenty two . Here he is now + near LI fifty + with a successful political CA RI.R + Ant still C+OIN+ STRON+ with perhaps the best still to CTUM . Nor man ST +JON STI. MFAS academic PROC+ RES an success +FOAS STAUN T+IN+ . T+EC+ RHys + ONR+S + an prizes/prises all the way . He was at both Oxford ANT+ came P+RI +J an also further afield at Yale . The political connection is seen/scene by/buy hiss P+JI.N+ T President of the came P+RI +J union an ale so/sow/sew later SECR TRI of the Oxford union . Nor/Nore manse CA RI.R in/inn politic has been equally successful + he has been conservative MP for CHELMS ford fifteen year +A MNIS TINR STLATH conservative government + ANT+ is now opposition spokesman on constitutional ANT+ HOAUFS commons a fairs + ANT+ the arts . Although he arrived in the HOAUFS commons +JU STAS I was leaving we have retained one/won very personal link + he ant I both SHERTH same barber in the hairdressers shop at HA rods . Laughter . . .

Fig. D1.1 Output produced by a transcription program incorporating no error correction whatsoever

The applause shows how much your/you're speech ant the way you shove answered questions has baying PREE SHJAY TED . . Thank you very muck indeed . . When Amman start in/inn politics ant early age the great/grate question mark/Mark is whether he has the saying power to remain in/inn such an exacting sin tress tinned ant exit earl exciting profession , many ion brightly for brief mow aren't/aunt then disappear from the seen/scene nave to reaper . Nave with nor/Nore man ST JON STEE VAS , he first con dace dad constituency in nineteen fifty one/won at the age of twenty two . Here he is now , nearly fifty , with a successful political careers , ant still going strong with perhaps the best still to term . Nor/Nore man ST JON STEE VAS academic rogue race an success WOAS STAUN DING . DEG RHys , ONR+S , an prizes/prises all the way . He was at both Oxford and Cambridge an also further afield at Yale . The political connection is seen/scene by/buy hiss being the President of the cambridge union an ale so/sow/sew later acre trip of the Oxford union . Nor/Nore manse careers in/inn politic has been equally successful , he has been conservative MP for Chelmsford fifteen year as/ass MNIS TINR STLA the conservative government , and is now opposition spokesman on constitutional and house of commons a fairs , and the arts . Although he arrived in the house of commons just I was leaving we have retained one/won very personal link , he ant I both share the same barber in the hairdressers shop at harrowed . Laughter . . .

Fig. D1.2 Output produced by a transcription program incorporating preprocessing corrections and the 'History and Future' correction routine

The applause shows how much your speech ant the way you shove answered questions has baying PREE SHJAY TED . . Thank you very muck indeed . . When Amman start in politics ant early age the great question mark is whether he has the saying power to remain in such an exacting sin tress tinned ant exit earl exciting profession , many ion brightly for brief mow aren't then disappear from the scene nave to reaper . Nave with nor man ST JON STEE VAS , he first con dace dad constituency in nineteen fifty one at the age of twenty two . Here he is now , nearly fifty , with a successful political careers , ant still going strong with perhaps the best still to term . Nor man ST JON STEE VAS academic rogue race an success WOAS STAUN DING . DEG RHys , ONR+S , an prizes all the way . He was at both Oxford and Cambridge an also further afield at Yale . The political connection is seen by hiss being the President of the cambridge union an ale so later acre trip of the Oxford union . Nor manse careers in politic has been equally successful , he has been conservative MP for Chelmsford fifteen year as MNIS TINR STLA the conservative government , and is now opposition spokesman on constitutional and house of commons a fairs , and the arts . Although he arrived in the house of commons just I was leaving we have retained one very personal link , he ant I both share the same barber in the hairdressers shop at harrowed . Laughter . . .

Fig. D1.3 Output produced by a transcription program incorporating preprocessing corrections, the 'History and Future' correction routine and syntax analysis

Its/It's a surprising THIN+ to say PUT T fact is that though once initial gut reaction to pictures of + baby battering is + God they MFU ST be/bee horrible ANT+ once instinct is to feel as/ass if one would like to smack them in the teeth a lot of people who seem out ward/Ward/warred LI fair LI nor/Nore mall ANT+ RE SPTECT P+L ANT+ pushed P+E JONLT+ T limit of control ANT+ hit their babies ANT+ cause/caws them very serious T+A + +M +MJ indeed . I mentioned some/sum frightening statistics but do we really know because so much of this is swept UN T+UR T car/Kerr pet isn't it . . Yes fee are SJI.N+ T Tip of the iceberg . There FOSH a parliamentary sup CMI I. FICH thought that six children a week were PT+JI.N+ CILCT+ in/inn Britain a week + an forty thousand/thou bad LI in/inn +JRT+ ANT+ lots of these were suffering life long brain damage ANT+ emotional harm + so/sow/sew its/it's a very big PRO P+LEM . . . So/sow/sew do fee now really know enough about the sort of person who suffers from this PRO P+LEM to be able to predict who is PTI C+EULR LI at risk . . Yes there are PTI. PL particularly at risk . Particularly very JUN+ mothers . If you look at MU THR+S who are baby batterers they tend on the whole/hole to be girls who have Scott/Scot PREC+ NNT early + had TFO get married or perhaps have not got married at all an feel that this pregnancy has PI.N FOIS TT+ on them + CTHEI have cot PREC+ NNT FI THAUT knowing anything about family planning ant find/fined themselves with THISH SCFO LIN+ + bundle not what they bargained for .

Fig. D2.1 Output produced by a transcription program incorporating no error correction whatsoever

Its/It's a surprising thing to say but the fact is that though once initial gut reaction to pictures of , baby battering is , God they must be horrible and once instinct is to feel as/ass if one would like to smack them in the teeth a lot of people who seem outwardly fairly nor/Nore mall and respect bless and pushed BE JONLD the limit of control and hit their babies and cause/caws them very serious DA , +M +MJ indeed . I mentioned some/sum frightening statistics but do we really know because so much of this is swept undone the car/Kerr pet isn't it . . Yes fee are seeing the Tip of the iceberg . There wash a parliamentary sub committee which/witch thought that six children a week were dying/dyeing keeled in/inn Britain a week , an forty thousand/thou badly in/inn jarred and lots of these were suffering life long brain damage and emotional harm , so/sow/sew its/it's a very big problem . . . So/sow/sew do fee now really know enough about the sort of person who suffers from this problem to be able to predict who is particularly at risk . . Yes there are steeple particularly at risk . Particularly very yearn mothers . If you look at musters who are baby batterers they tend on the whole/hole to be girls who have Scott/Scot pregnant early , had to get married or perhaps have not got married at all an feel that this pregnancy has been fees tit on them , CTHEI have cot pregnant FI THAUT knowing anything about family planning ant find/fined themselves with these squalid , bundle not what they bargained for .

Fig. D2.2 Output produced by a transcription program incorporating preprocessing corrections and the 'History and Future' correction routine

It's a surprising thing to say but the fact is that though once initial gut reaction to pictures of , baby battering is , God they must be horrible and once instinct is to feel as if one would like to smack them in the teeth a lot of people who seem outwardly fairly nor mall and respect bless and pushed BE JONLD the limit of control and hit their babies and cause them very serious DA , +M +MJ indeed . I mentioned some frightening statistics but do we really know because so much of this is swept undone the car pet isn't it . . Yes fee are seeing the Tip of the iceberg . There wash a parliamentary sub committee which thought that six children a week were dying keeled in Britain a week , an forty thousand badly in jarred and lots of these were suffering life long brain damage and emotional harm , so it's a very big problem . . . So do fee now really know enough about the sort of person who suffers from this problem to be able to predict who is particularly at risk . . Yes there are steeple particularly at risk . Particularly very yearn mothers . If you look at musters who are baby batterers they tend on the whole to be girls who have Scott pregnant early , had to get married or perhaps have not got married at all an feel that this pregnancy has been fees tit on them , CTHEI have cot pregnant FI THAUT knowing anything about family planning ant find themselves with these squalid , bundle not what they bargained for .

Fig. D2.3 Output produced by a transcription program incorporating preprocessing corrections, the 'History and Future' correction routine and syntax analysis

Alternatively we CA go TOTH deaf or C+NAI Asians the hard of hearing or C+NAI Asians ANT+ say what do you THINCTH hearing impaired want subtitled + ANT+ they tend to PRO to use lists which look something like that . We have started looking at audience RES PON SS ANT+ we try to T+I FREN HJE.T P+E TFI.NTH RES PON SS of the hearing impaired audience an the RES PON SS of a nor/Nore mall hearing awe T+JEN + ANT+ we find that there is not very much difference in fact P+E TFI.NTH programmes these two groups of PLI. watch + so/sow/sew there is some evidence to say that one should concentrate on peak MFEUIN+ times audience RE. TIN+S . The next THIN+ we need TOAS C is + what P+A lance should we draw between quality ANT+ amount . There is all FEIS going to dee a limb TAT+ mount of money in the kitty + do fee subtitle as/ass many programmes as/ass we possibly can but not worry too much bout how CT+OUT+ the subtitles are so that we can do it cheap LI + or do we say we will make HOURTH subtitles are the best THIN+ we can possibly give you + ANT+ I NEFT P+LI that will + mean that fewer programmes get subtitled because it will take more time ANT+ cost more money . . . Perhaps some/sum programmes are unsuitable for the hearing impaired + perhaps some/sum are PTFI CEULR LI unsuitable for the PRE Lynn C+EU LI deaf . For in/inn STNS + humour is not very easy to get over in a subtitled form . Huns ant punts SOEAN on may not have a meaning HFEIN the written form . . . That was puns ANT+ so on . . .

Fig. D3.1 Output produced by a transcription program incorporating no error correction whatsoever

Alternatively we cat go to the deaf organisations the hard of hearing organisations and say what do you think the hearing impaired want subtitled , and they tend to produce lists which look something like that . We have started looking at audience responses and we try to differentiate between the responses of the hearing impaired audience an the responses of a nor/Nore mall hearing awe Dane , and we find that there is not very much difference in fact between the programmes these two groups of leigh watch , so/sow/sew there is some evidence to say that one should concentrate on peak viewing times audience ratings . The next thing we need to ask is , what balance should we draw between quality and amount . There is always going to dee a limb dad mount of money in the kitty , do fee subtitle as/ass many programmes as/ass we possibly can but not worry too much bout how good the subtitles are so that we can do it cheaply , or do we say we will make sure the subtitles are the best thing we can possibly give you , and inedibly that will , mean that fewer programmes get subtitled because it will take more time and cost more money . . . Perhaps some/sum programmes are unsuitable for the hearing impaired , perhaps some/sum are particularly unsuitable for the pressing gulley deaf . For in/inn stens , humour is not very easy to get over in a subtitled form . Huns ant punts sane on may not have a meaning when in the written form . . . That was puns and so on . . .

Fig. D3.2 Output produced by a transcription program incorporating preprocessing corrections and the 'History and Future' correction routine

Alternatively we cat go to the deaf organisations the hard of hearing organisations and say what do you think the hearing impaired want subtitled , and they tend to produce lists which look something like that . We have started looking at audience responses and we try to differentiate between the responses of the hearing impaired audience an the responses of a nor mall hearing awe Dane , and we find that there is not very much difference in fact between the programmes these two groups of leigh watch , so there is some evidence to say that one should concentrate on peak viewing times audience ratings . The next thing we need to ask is , what balance should we draw between quality and amount . There is always going to dee a limb dad mount of money in the kitty , do fee subtitle as many programmes as we possibly can but not worry too much bout how good the subtitles are so that we can do it cheaply , or do we say we will make sure the subtitles are the best thing we can possibly give you , and inedibly that will , mean that fewer programmes get subtitled because it will take more time and cost more money . . . Perhaps some programmes are unsuitable for the hearing impaired , perhaps some are particularly unsuitable for the pressing gulley deaf . For in stens , humour is not very easy to get over in a subtitled form . Huns ant punts sane on may not have a meaning when in the written form . . . That was puns and so on . . .

Fig. D3.3 Output produced by a transcription program incorporating preprocessing corrections, the 'History and Future' correction routine and syntax analysis