AD-A021929

NRL Report 7948

AD/A021 929

# Automatic Translation of English Text to Phonetics by Means of Letter-to-Sound Rules

HONEY SUE ELOVITZ, RODNEY W. JOHNSON,
ASTRID MCHUGH, AND JOHN E. SHORE

*Information Systems Staff*
*Communication Sciences Division*

January 21, 1976

NAVAL RESEARCH LABORATORY
Washington, D.C.

Approved for public release; distribution unlimited.

| | | |
|---|---|---|
| **REPORT DOCUMENTATION PAGE** | | READ INSTRUCTIONS<br>BEFORE COMPLETING FORM |
| 1. REPORT NUMBER<br>NRL Report 7948 | 2. GOVT ACCESSION NO. | 3. RECIPIENT'S CATALOG NUMBER |
| 4. TITLE (and Subtitle)<br>AUTOMATIC TRANSLATION OF ENGLISH TEXT TO PHONETICS BY MEANS OF LETTER-TO-SOUND RULES | | 5. TYPE OF REPORT & PERIOD COVERED<br>Interim report on a continuing NRL Problem<br>6. PERFORMING ORG. REPORT NUMBER |
| 7. AUTHOR(s)<br>Honey Sue Elovitz, Rodney W. Johnson, Astrid McHugh, and John E. Shore | | 8. CONTRACT OR GRANT NUMBER(s) |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS<br>Naval Research Laboratory<br>Washington, D.C. 20375 | | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS<br>NRL Problem B02-15<br>RF21-222-401-4356, 62721N |
| 11. CONTROLLING OFFICE NAME AND ADDRESS<br>Department of the Navy<br>Office of Naval Research<br>Arlington, Va. 22217 | | 12. REPORT DATE<br>January 21, 1976<br>13. NUMBER OF PAGES<br>101 |
| 14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) | | 15. SECURITY CLASS. (of this report)<br>Unclassified<br>15a. DECLASSIFICATION/DOWNGRADING SCHEDULE |

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Speech synthesis
Computer voice output
Text-to-speech translation
Man-machine communication

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Speech synthesizers for computer voice output are most useful when not restricted to a prestored vocabulary. The simplest approach to unrestricted text-to-speech translation uses a small set of letter-to-sound rules, each specifying a pronunciation for one or more letters in some context. Unless this approach yields sufficient intelligibility, routine addition of text-to-speech translation to computer systems is unlikely, since more elaborate approaches embodying large pronunciation dictionaries or linguistic analysis require too much of the available computing resources.

(continued)

(continued from block 20)

The work described here demonstrates the practicality of routine text-to-speech translation. A set of 329 letter-to-sound rules has been developed. These translate English text into the International Phonetic Alphabet (IPA), producing correct pronunciations for approximately 90% of the words in an average text sample. Most of the remaining 10% have single errors easily correctable by the listener. Another set of rules translates IPA into the phonetic coding for a particular commercial speech synthesizer.

This report describes the technical approach used and the support hardware and software developed. It gives overall performance figures, detailed statistics showing the importance of each rule, and listings of a translation program and a program used in rule development.

## CONTENTS

# AUTOMATIC TRANSLATION OF ENGLISH TEXT TO PHONETICS BY MEANS OF LETTER-TO-SOUND RULES

## INTRODUCTION

Hardware to produce synthetic speech existed in various forms as early as 1939. At the New York World's Fair in that year, Homer Dudley exhibited his Synthetic Speaker [1], the ancestor of many of the more successful speech synthesizers now in use. Today phonetically programmable synthesizers of reasonable intelligibility are commerically available for a few thousand dollars. Such devices have stimulated widespread interest in computer voice output for various civilian and Department of Defense (DoD) applications. A further impetus to DoD interest is resulting from the development of narrowband digital voice-transmission systems, such as NRL's Linear Predictive Coder [2], and the likelihood of their widespread future use. These speech-transmission systems include a synthesizer that could also be used for computer voice output.

Among the most promising applications of computer voice output are:

- ways to transmit information from English-language data bases to remote locations by telephone,
- a channel of communication with busy operators of computer-controlled systems who have to give most of their attention to complicated visual displays and would find extraneous text messages intolerable, and
- reading machines for the blind.

In such applications the potential utility of computer-controlled speech synthesizers is greatly enhanced if the speech is not restricted to a prestored vocabulary.

Among the numerous approaches to providing such unrestricted text-to-speech translation, the simplest is to use a small set of letter-to-sound rules to guess at the pronunciation of any word. Each rule specifies a phonetic correspondence to one or more letters. In some cases the letter's context is used to determine which rule should be applied. An example is the elementary school rule "when two vowels go walking, the first one does the talking," which indicates that when one vowel is followed by another, the first is transcribed into the long vowel phoneme whereas the second vowel is silent and receives no phonetic symbol. In other cases no context is necessary, as with the letter j, which usually receives the /dʒ/ phoneme. (The International Phonetic Alphabet (IPA)* will be used to denote English phonemes and indicate pronunciations.)

A more complicated approach, and one requiring much more storage, uses a large pronunciation dictionary supplemented by various sets of rules. Words are isolated from the text and looked up in the dictionary. If the lookup fails, various rules are used to break the word into constituent parts for which there are dictionary entries. Finally, if all else fails, letter-to-sound rules are used to guess at the pronunciation.

Manuscript submitted October 29, 1975.

*Table 1.

A yet more elaborate approach adds syntactic analysis of sentences to the preceding in order to determine the part of speech of each word. This resolves the pronunciation ambiguities of words like approximate (adjective or verb?) and house (noun or verb?). Finally, well beyond the current state of the art, one could imagine an approach incorporating a semantic analysis sophisticated enough to decide whether unionized refers to unions or ionization.

To be attractive as a routine addition to computer systems, text-to-speech translation cannot require a large fraction of the available computational resources. This constraint, which is particularly strong for real-time military systems, precludes approaches that embody large pronouncing dictionaries or linguistic analysis programs. Thus routine use of text-to-speech translation is likely only if sufficient intelligibility can be attained with a limited set of letter-to-sound rules.

We report here on work that has demonstrated the practicality of routine text-to-speech translation. We have developed a set of 329 letter-to-sound rules that translate English text into the International Phonetic Alphabet (IPA). Using the 50,000-word Standard Corpus of Present-Day Edited American English ("Brown Corpus") [3], we have determined that the rules will produce correct pronunciations for approximately 90% of the words in an average sample of English text. Typically the remaining 10% have single errors that in most cases can easily be mentally corrected by the listener. A separate set of rules was developed to translate from IPA into a phonetic encoding compatible with a particular commercial speech synthesizer (Federal Screw Works Votrax VS-6).

In the next section we discuss previous work in text-to-speech translation. The technical approach used in the NRL system is described in the third section as are the support hardware and software that we developed. Our results are summarized in the fourth section. Together with overall performance figures, we give detailed statistics that show the importance of each rule. Our conclusions and our plans for further work are discussed in the fifth section. Descriptions and listings of two SNOBOL programs that were important for our work are included as appendixes. A third appendix contains some remarks on the improvement in these programs' performance that followed our changing from an interpreted version of SNOBOL to a compiled version, FASBOL.

## SOME EXISTING TEXT-TO-SPEECH SYSTEMS

Text-to-speech systems have been built ranging in complexity from letter-to-sound rule systems to dictionary-lookup systems with syntactic analysis. We will describe three briefly: those developed at MIT, the University of Keele, and Bell Telephone Laboratories. None that we encountered however completely satisfy all the criteria we imposed:

- The implementation must be straightforward, for reasons given in the Introduction, requiring little space for the program and none at all for large dictionaries;
- The translation rules must be easily modifiable, both to allow for development and improvement of the rules and to permit the system to be tailored to a variety of special applications;
- The system should not be tied to a particular hardware synthesizer;

- There should be an objective measure of the system's performance.

## MIT System

Allen and Lee have reported on research in automatic text translation at the Massachusetts Institute of Technology [4-8]. The MIT system not only confronts the text-to-speech conversion problem but attempts to read printed text using a character recognizer. The MIT system includes a parts-of-speech preprocessor to aid in the pronunciation of such homographs as refuse, appropriate, and lives. After parts-of-speech analysis, the system, using a phrase analyzer module, assigns such prosodic features as inflection and stress to the phonetic transcription. The resulting string of phonemes and prosodic features is transformed to the signals needed to operate the synthesizer, designed in the MIT laboratory.

The grapheme-to-phoneme translator uses a typical dictionary-lookup approach with a set of letter-to-sound rules. One word is isolated from the input text and looked up in a dictionary. If the word is found and has no alternate transcriptions, the result is passed to the phrase analyzer, assigned prosodic features, and passed for speech-synthesizer parametrization. If an alternate transcription is encountered, the parts-of-speech information obtained by the parts-of-speech preprocessor is used to determine which transcription is to be used. This result is then passed along the translation chain.

When a word is not found in the dictionary, an attempt is made to partition the word into morphs and isolate affixes. The individual morphs are then looked up in the dictionary. If they are found, the result is passed along for stress analysis and synthesizer parametrization as before. When all else fails, the set of letter-to-sound rules is applied to the original input word.

Currently the MIT system contains a dictionary of 11,000 words and a set of approximately 400 letter-to-sound rules [9]. The phrase analyzer does not parse a sentence completely, but techniques to assign prosodic features are being investigated. Each item in the dictionary requires parts-of-speech information and alternate transcriptions along with various internal flags. Consequently the amount of external computer storage can grow quite large. Lee estimates that a 32,000-word dictionary requires approximately 4 million bits [4]. Additionally the internal storage for such a translation program could become quite large when new features such as syntax analysis and prosodic feature assignment are added. A comprehensive list of the letter-to-sound rules has not been published, nor has a quantitative evaluation of the system's performance.

## University of Keele System

The system developed at the University of Keele in England by Ainsworth [10] is a letter-to-sound-rule system that converts text punched on paper tape to symbols used to generate parameters to control a speech synthesizer. Ainsworth does the translation to speech in the following steps.

1. Segmentation into breath groups,
2. Translation to phonemes via letter-to-sound rules,
3. Lexical stress assignment,
4. Speech synthesizer parametrization.

Step 1 inserts pauses at convenient locations, to provide more natural sounding speech. A translation buffer of about 50 characters is filled until a punctuation mark is encountered. This buffer becomes a breath group. If the buffer is filled before a punctuation mark is encountered, the buffer is search for a conjunction, and the buffer up to the conjunction becomes a breath group. If a conjunction does not occur, an auxiliary verb, a preposition, or an article is searched for. Otherwise the entire contents of the buffer becomes a breath group.

Step 2 provides the translation of input text to phonemes. Ainsworth's rules are intended to produce a dialect of British English. These rules are context sensitive, and the order of their application is critical. For example, the rules

(o)ing /əu/

and (oi) /ɔi/

occur in that order among the rules for translating the letter o. The first illustrates context dependence; it states that o, in the context of following ing, is pronounced as /əu/, like the o in going in Ainsworth's dialect. The order is important since going matches both rules. In such a case the first matching rule is used; if the order were reversed, the oi in going would be transcribed as /ɔi/, the sound of the oi in coin. Ainsworth's rules were the starting point for the development of the rules used by the NRL system.

Ainsworth reports performance measures based on 1000-word passages from three sources: a textbook, a novel, and a newspaper. His figures show 92% of the words in the first sample correctly translated, 89% in the second, and 89% of the third. Listening tests using the same three passages showed scores ranging from 50% to 90% of words correctly understood.

The rules are embodied as a section of PDP-8 assembly code with numerous conditional branches testing the symbol being translated and its neighbors [11]. Changing the rules would presumably involve rewriting part of the assembly code and reassembling.

## Bell Telephone Laboratories System

Another system for translating text to speech by letter-to-sound rules has been described by McIlroy [12] at Bell Telephone Laboratories. McIlroy's system contains more than 750 letter-to-sound rules, which include 100 words, 580 word fragments, and 70 letters and occupies 11,000 bytes in a PDP 11/45. This is the typical approach taken by a letter-to-sound rule system. The system has a small 100-word exception dictionary, with the remainder being context sensitive translations (the 580 word fragments).

The approach taken is to isolate a word from the input text and attempt to find it in the exception dictionary. If the word is not found, capital letters are converted to lower-case letters and leading and trailing punctuation eliminated. The dictionary is then searched for the converted word. If it still is not found, a final s is removed and final ie is changed to y when appropriate. The altered word is looked up. If none of the above procedures succeeds in finding the word in the dictionary, letter-to-sound rules are applied.

McIlroy's rules specify not only phonetic output but alterations to be made in the input string. For instance, his qu rule outputs a synthesizer code corresponding to the /k/ phoneme and also rewrites the input string so that w appears instead of u. This additional complication allows his war rule to give the right pronounciation to the a, not only in war, but in quart.

McIlroy reported that the program performed satisfactorily for 97% of the 2000 most common words listed in the Brown Corpus [3] and performed satisfactorily for 88% of the tail consisting of a 1% sample of the Corpus remainder. McIlroy does not report the criterion of satisfactory performance used.

The 750 rules mentioned are contained in tables in the program and are fairly easy to modify. A number of others however are embedded in the program code. These include rules for marking medial and final silent e, common suffixes, certain potential long vowels, and voiced s. The system directly generates codes for a particular synthesizer; no IPA transcription is produced.

## THE NRL SYSTEM

As was discussed in the Introduction, the NRL system is designed to test the conjecture that acceptable intelligibility can be obtained with a limited set of letter-to-sound rules. The implementation algorithm is simpler than either McIlroy's or Ainsworth's in that it involves fewer ad hoc preprocessing steps before the application of the rules. McIlroy's final-s stripping and ie-to-y conversion are absent, as is his lookup in an exceptions dictionary. Instead of an exceptions dictionary, we have included, for each word needing individual treatment, a rule giving its correct pronunciation; such single-word rules make up about a sixth of the full set. Ainsworth's breath-group segmentation is also absent, although we include some rules that convert punctuation into pauses of various lengths. The NRL system, like Ainsworth's, but unlike McIlroy's, does no rewriting of the input string and produces IPA as the output of the rules. The decision to use IPA was due to our desire not to be tied to a particular synthesizer; the text-to-phonetics information is contained in device-independent rules, and only the more direct phonetics-to-synthesizer rules need to be changed when it is desired to change to a new synthesizer.

Because we required a convenient means of changing the rules in the course of their development, we have not immediately proceeded to a hand-coded system (like Ainsworth's) which incorporates the rules in the form of assembly code. Among the research tools we have developed is a translation program in SNOBOL, to be described more fully, which contains the rules as a text string easily modifiable even by someone with no knowledge of SNOBOL.

## Research Tools — Hardware

Our work so far has used a commercial speech synthesizer, a Federal Screw Works Votrax VS-6 audio-response unit. It can produce 63 basic speech sounds (called "phonemes" by the manufacturer) at four different pitch levels (inflections) and string them together to form continuous speech. Although the Votrax "phonemes" do not correspond exactly to the phonemes of English, one can set up a fairly straightforward mapping from a phonemic transcription to Votrax codes.

We used the synthesizer with a system of support devices that provide for convenient input, output, and manipulation of phonetic texts. The speech-synthesis laboratory system (Fig. 1) includes a minicomputer and a collection of peripheral devices. Besides the speech synthesizer, there are a phonetic keyboard, a terminal with twin digital magnetic-tape cassette units, a cathode-ray-tube (CRT) terminal, a teletype with paper-tape punch and reader, and a modem for communication with NRL's PDP-10.

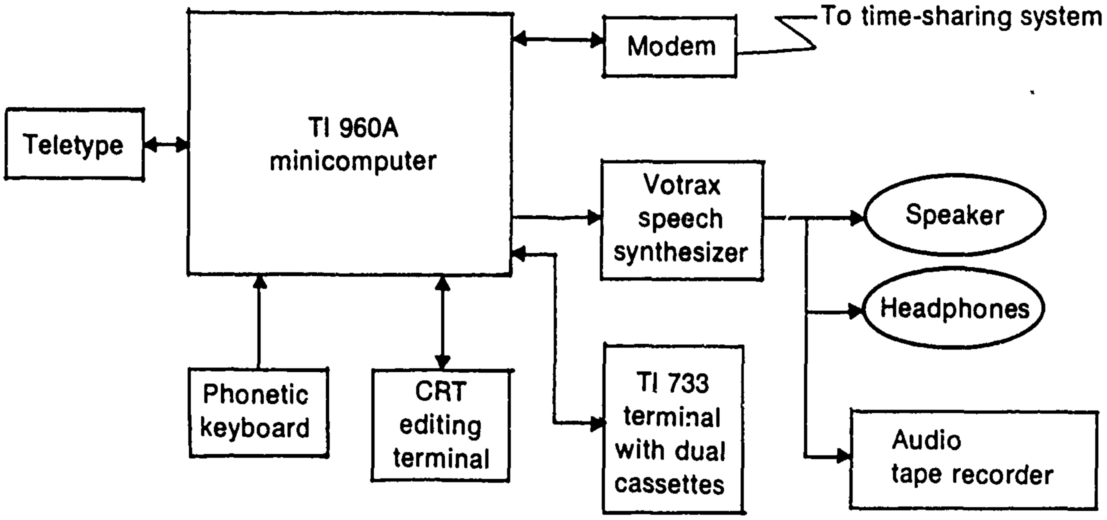


Fig. 1 — The Naval Research Laboratory's speech laboratory system

The phonetic keyboard, made by Federal Screw Works for use with the Votrax synthesizer, has a key for each phoneme, four inflection keys, and a few control keys.

The terminal is a Texas Instruments (TI) 733 Silent 700 data terminal, used for typing commands to control the system, for entering phonetic texts and other messages, and for printing out messages and error reports. The cassette units record messages on tape and play them back. The teletype is a backup for the TI 733 terminal and permits paper tapes to be punched and read.

Editing is the function of the CRT terminal, a Delta Data Systems TelTerm video-display terminal. Messages can be sent to the screen by the system or typed there directly from the CRT keyboard, characters can be added or deleted, and the resulting message can be sent back to the system for transmission to another device. For example a phonetic message can be composed on the screen, edited, and spoken out by the Votrax; it can then be edited further and spoken out again. A permanent copy can be printed on the TI 733 or teletype, recorded digitally on the TI 733's cassette unit, or recorded on an audio tape recorder.

The minicomputer is a TI 960A computer with 12,000 16-bit words of memory. It receives messages from the peripheral devices, transmits messages to the devices, holds messages in buffers in its memory, and translates messages to formats appropriate to the various peripheral devices. The messages are transferred and translated in response to commands that are usually entered from the TI 733 terminal keyboard. It is possible however to specify another peripheral device or a memory buffer as the source for commands.

The modem links the TI 960A to a remote time-sharing computer when computations are needed beyond the current capabilities of the TI 960A software. Among these computations is the translation of English text to phonetics, which is handled by a SNOBOL program running on NRL's PDP-10. The procedure is to link to the PDP-10 by telephone, start the SNOBOL program, send it an English-text message from the terminal, and record on a cassette the phonetic text received in reply. The cassette is then played back for editing, speaking out through the Votrax, and the like.

### Research Tools -- Software

TRANS, the translation program mentioned, accepts text, applies the translation rules, and returns the translated results. Input may come from the terminal or a text file; output may be sent to a file, the terminal printer, or the cassette unit. The complete translation from English to Votrax codes may be requested, or the English-to-IPA or IPA-to-Votrax pass may be requested separately. TRANS is described more completely in Appendix A.

The rules are kept in character strings in a form easy for human beings to read and write. They are interpreted by the program. Each rule has the form

A[B] C=D

which is essentially the same form as Ainsworth's. The meaning is "The character string B, occurring with left context A and right context C, gets the pronunciation D."

D consists of IPA symbols — or rather a capitalized latin-letter representation of IPA to cater to computer character sets (Table 1). B is a letter or text fragment to be translated. A and C are patterns; like B they may be strings of letters and other characters, but some special symbols denote classes of strings such as "voiced consonant" and "vowel cluster." Table 2 lists the symbols that have such special interpretations. Blanks are significant, because they identify the beginnings and ends of words.

Table 1
Latin-Letter Representation of IPA

| Standard IPA | Representation | Example | Standard IPA | Representation | Example |
|---|---|---|---|---|---|
| i | IY | beet | g | G | goat |
| ɪ | IH | bit | f | F | fault |
| e | EY | gate | v | V | vault |
| ɛ | EH | get | θ | TH | ether |
| æ | AE | fat | ð | DH | either |
| a | AA | father | s | S | sue |
| ɔ | AO | lawn | z | Z | zoo |
| o | OW | lone | ʃ | SH | leash |
| ʊ | UH | full | ʒ | ZH | leisure |
| u | UW | fool | h | HH | how |
| ɝ, ɚ | ER | murder | m | M | sum |
| ə | AX | about | n | N | sun |
| ʌ | AH | but | ŋ | NX | sung |
| aɪ | AY | hide | l | L | laugh |
| aʊ | AW | how | w | W | wear |
| ɔɪ | OY | toy | j | Y | young |
| p | P | pack | r | R | rate |
| b | B | back | tʃ | CH | char |
| t | T | time | dʒ | JH | jar |
| d | D | dime | hw | WH | where |
| k | K | coat | | | |

Table 2
Special Symbols Appearing in the English-to-IPA Translation Rules

| Symbol | Meaning |
|---|---|
| # | One or more vowels* |
| * | One or more consonants† |
| • | One of B, D, V, G, J, L, M, N, R, W, and Z: a voiced consonant |
| $ | One consonant followed by an E or I |
| % | One of (ER, E, ES, ED, ING, ELY): a suffix |
| & | One of (S, C, G, Z, X, J, CH, SH): a sibilant |
| @ | One of (T, S, R, D, L, Z, N, J, TH, CH, SH): a consonant influencing the sound of a following long u (cf. rule and mule) |
| ∧ | One consonant |
| + | One of (E, I, Y): a front vowel |
| : | Zero or more consonants |

*Vowels are A, E, I, O, U, Y.
†Consonants are B, C, D, F, G, H, J, K, L, M, N, P, Q, R, S, T, V, W, X, Z.

For example, a typical rule is

' C[O]M=/AA/',

which means that an O after an initial C and before an M gets the pronunciation /a/, the a-sound in father. Another rule is

' :[E] =/IY]',

where the colon denotes any sequence of zero or more consonants, which means that final e, if the only vowel in a word, gets the long-e sound /i/ of be and she.

The translation algorithm scans input text from left to right and, for each character scanned, sequentially searches the rules pertinent to that character until it finds one whose left-hand side matches the text at the correct position. It outputs the right-hand side, passes over the characters bracketed in the rule, and resumes the scan with the next character of text. The input string is never altered.

To illustrate the operation of the algorithm, we will describe a worked example: the translation of RATIO using the English-to-IPA rules from the program listing of TRANS in Appendix A.

To the left of the first character, R, the program adds a blank to delimit the word, and the scan starts with the R, as we indicate with a pointer: $_{\uparrow}$RATIO. The program searches the R rules -- the rules with R as the first character between brackets. The first R rule, ' [RE]^#=/R IY/', fails to match, since it requires that R be followed by E. The next, and last, R rule, '[R]=/R/', is the default; it matches any R not matched by earlier rules. Consequently, /R/ goes into the output string, and the scan moves past the R to A: R$_{\uparrow}$ATIO.

The search of the A rules turns up no match before '[A]^+#=/EY/', which applies when A is followed by a single consonant, a front vowel (E, I, or Y), and another vowel. The program adds /EY/ to the output and moves the pointer past the A to T: RA$_{\uparrow}$TIO.

The first T rule that matches is '[TI]O=/SH/'. Consequently, /SH/ goes into the output, and the pointer moves past TI to O: RATI$_{\uparrow}$O. The program does not search the I rules, since the I occurs inside the brackets with the T; the string TI as a whole gets the pronunciation /SH/ and no output phonemes correspond to I alone.

The first match among the O rules is '[O]=/OW/'; the program outputs /OW/ and moves the pointer past the O to the blank at the end of the word: RATIO$_{\uparrow}$. The output string is /R/ /EY/ /SH/ /OW/, which represents the IPA /reʃo/, the correct transcription [12]. If the translation continued, the next matching rule would be in the set that passes blanks, commas, periods, and other punctuation into the output string as /⟨ ⟩/, /⟨,⟩/, /⟨.⟩/, etc. The program would output /⟨ ⟩/ and move the pointer past the blank to the beginning of the next word, if any.

The IPA output string is the input to a second pass that uses the same algorithm and rules of the same form to translate IPA to Votrax codes. The IPA-to-Votrax rules are fewer and more straightforward than the English-to-IPA rules (for example, '[T]=[T]'). Since the synthesizer automatically varies the pronunciation of its "phonemes" to suit various contexts, the rules need not contain much context dependence. Some context-dependent rules have been included however to implement the manufacturer's suggestions about liquids, particularly L, adjacent to certain vowels. The complete set of rules is contained in the program listings of TRANS in Appendix A.

Another program DICT, was used during rule development to insure that a rule change proposed to fix up a dozen mispronounced words would not ruin a hundred others previously translated correctly. DICT accepts a pattern like the left-hand side of a rule but without brackets; it gives the same interpretations as TRANS to the same special symbols. After reading the pattern, DICT searches a file of words and outputs the words that contain a match. The program is described in Appendix B.

DICT must read the entire file of words and convert to SNOBOL internal representation before searching. Although we have a copy of the frequency-ordered list of words in the Brown Corpus [3] on line, core-size restrictions have limited us to searching a few thousand words at a time. DICT was complemented by the on-line text-editing program SOS, which can search an entire text file for patterns. Pattern searching in SOS is less convenient than in DICT; for instance, one cannot specify "consonant" as an element of an SOS search pattern. However with SOS we could search the entire 50,000-word Brown Corpus file.

The Brown Corpus comprises 500 samples of English text written in a wide variety of styles. Each sample is roughly 2000 words long, and the entire Corpus totals slightly more than a million words. The file we use lists the roughly 50,000 individual words occurring in the Corpus, arranged in decreasing order of frequency. The entry for each word contains some items of numerical information, including frequency (the number of occurrences of the word in the Corpus) and number of texts (the number of text samples, among the 500 comprising the Corpus, in which the word occurs).

One output that can be requested from TRANS is a stat file — a file listing every instance of every rule used in translating every word in a text file. A program STAT reads stat files and produced statistics on the relative importance of the rules. For each rule STAT counts the words in whose translation the rule was used, sums the frequencies of those words, and sums the number of text samples, among the 500 in the Corpus, in which each of those words appear. The output comprises these three absolute results together with the relative results obtained by normalizing the absolute ones so that their sums over all rules are 1.

Pre- and postprocessors were written to enable the time-sharing system SORT utility to produce from a stat file a file giving, for each rule, a list of all the words in whose translation the rule was used. This provides a detailed analysis of the interactions of a set of rules. A program for line-by-line comparison of two files was used to compare translations of a text file by different sets of rules. In scoring the results of translating a set of words, a program was used that accepts a user's "good/bad" judgments on translated words and accumulates total and frequency-weighted total scores.

## Rule Development

Our starting point, version 1 of the rules, was a modification of Ainsworth's set. The main alterations were changes in the right-hand sides to Americanize the accent and additions to handle final S, ES, and ED correctly. Then began a development cycle with the following steps:

1. Translate. With version 1 we translated the most frequent 4000 words in the Brown Corpus. With later versions we included samples from deeper in the corpus.

2. Examine results. We had much of the translated output spoken by the synthesizer and listened to it, marking mistakes on a printed listing. Kenyon and Knott's pronouncing dictionary [13] was the arbiter in case of doubt or disagreement as to what constituted a mistake. (The authors' linguistic backgrounds are diverse enough that disagreements were fairly frequent). Later in the project we grew proficient enough at reading the machine representation of IPA to risk checking some samples visually, but we never abandoned the practice of listening to at least part of the output from each version of the rules. The major goal was a good IPA transcription. In the few cases where a correct transcription still sounded strange, the IPA-to-Votrax rules were fixed up when possible, and the problem was otherwise blamed on the synthesizer.

3. Classify errors. We divided the mispronounced words into lists with headings like "TH problem," "Silent E problem," "Long A problem," and "Stress problems." Then we

scanned the lists to identify specific letter patterns being frequently mistranslated.

4. Modify. For a given frequently mistranslated letter pattern, we would find all sufficiently frequent words, mistranslated or not, that matched the pattern. If the correct pronunciations agreed in a majority of cases, or in even a clear plurality of cases, we wrote a new or altered rule tc give that pronunciation; otherwise we tried a more specific context. For example, version 1 had no rule for the EA combination, which has a great variety of pronunciations: great, heart, ready, sea, earth. Most words containing EA showed up on the "EA problem" list. We found the long-e̲ pronunciation /i/ in roughly half of them. The addition of a rule '[EA]=/IY/' was justified, since it improved many words and did not harm the rest. Meat received the correct pronunciation /mit/, and great was no worse as /grit/ than it had been as /grɛ æt/. During the second round of development many EA words still showed up as problems, but a search with DICT turned up the large number now getting the correct pronunciation. Looking for a more specific pattern, we found lots of EAD words on the problem list. A search of the Corpus for EAD words suggested adding a rule '[EA] D=/EH/', which fixes ready, changes one acceptable pronunciation of lead to another, and hurts a few previously correct words like bead. The additions and alterations continued until the accumulation of changes made the interactions between rules hard to keep track of.

5. Iterate. Having produced a new version, we would start the cycle over by translating several thousand words. We went through the cycle twice, ending with version 3. Before testing version 3 we pruned the rules by looking at the STAT outputs for version 2 and removing rules that were rarely used. Hence the rules for initial PT and initial X, although quite reliable, were thrown out for small importance.

### Testing

We tested version 3 by translating the 8000 most frequent words plus a 1000-word sample selected from the tail of the corpus — words with frequencies of 1 or 2 per million. The first 5000 words and the tail sample were scored like the translations by earlier versions: the criterion for correctness was a good IPA transcription, and, although we did not look up most words in a pronouncing dictionary, Kenyon and Knott [13] was the arbiter when questions arose. Numbers, symbols, and abbreviations were excluded from the scoring. Any transcription accepted by Kenyon and Knott was allowed, not just the preferred. Some deviations were allowed. The horse:hoarse distinction (/ɔr/ vs /or/) was ignored, as were the Mary:merry:marry distinction and similar distinctions involving vowels followed by R. Doubled consonants (/bIttɚ/ instead of /bItɚ/ for bitter) were not counted as errors. Otherwise we tried to be quite strict in scoring consonants and stressed vowels. Sometimes an unstressed vowel translated with the full or stressed pronunciation was classed as a "stress problem" rather than a mistake, if vowel reduction upon stressing would give a good transcription. Thus /æbaUt/ instead of /əbaUt/ for about, though marked as a stress problem, was not scored as an error. Some subjectivity entered here. Stress problems judged less severe than that in about were sometimes not marked at all; more severe ones were sometimes scored as errors.

## RESULTS

Table 3 gives the result of scoring IPA transcriptions of 1000-word samples from the Brown Corpus. The first three columns are based on a count of the number of distinct words correctly translated and the total number translated. The last three columns are based on the sums of the frequencies of the correctly translated words and of all the translated words. The frequencies were obtained from the Corpus; they give the number of times the word appeared and thus represent roughly parts per million. The first rows are based on successive 1000-word samples, starting from the beginning of the Corpus; the last is based on 1000 words selected from the tail of the Corpus (1/18 of the words with 2 occurrences per million and 1/36 of those with 1 per million).

Table 3
Scores and Frequency-Weighted Scores for 1000-Word Samples from the Brown Corpus translated by Version 3 of the Rules

| Sample | No. of Words Scored | No. of Words Correct | Percent Correct | Total Frequency of Words Scored | Total Frequency of Correct Words | Percent Correct (Frequency Weighted) |
|---|---|---|---|---|---|---|
| 1 | 976 | 847 | 86.8 | 691,375 | 664,564 | 96.1 |
| 2 | 974 | 808 | 83.0 | 72,966 | 60,862 | 83.4 |
| 3 | 973 | 744 | 76.5 | 43,664 | 33,401 | 76.5 |
| 4 | 988 | 757 | 76.6 | 30,391 | 23,315 | 76.6 |
| 5 | 971 | 707 | 72.8 | 21,601 | 15,743 | 72.9 |
| Tail | 922 | 599 | 65.0 | 1,295 | 849 | 65.6 |

Table 4 gives similar, cumulative results based on the first 1000, first 2000, first 3000, etc. words of the Corpus; the last line is an estimate, derived from the foregoing, of the results that would have been obtained had the entire Corpus been translated and scored. The upper bounds were computed under the assumption that the error rate observed in the fifth 1000-word sample (Table 3) held constant up to the beginning of the tail sample; the lower bounds assume that the error rate following the first 5000 words is equal to that observed in the tail. The figures 89% to 90% in the last column mean that, assuming the Corpus frequencies are representative, we would expect to correctly translate 89% to 90% of the words in a random sample of English text.

Table 5 gives results for the first 1000 words as translated at various stages of rule development.

Table 4
Cumulative Scores and Frequency-Weighted Scores for the First n Thousand Words of the Brown Corpus Translated by Version 3 of the Rules

| n | No. of Words Scored | No. of Words Correct | Percent Correct | Total Frequency of Words Scored | Total Frequency of Correct Words | Percent Correct (Frequency Weighted) |
|---|---|---|---|---|---|---|
| 1 | 976 | 847 | 86.8 | 691,375 | 664,564 | 96.1 |
| 2 | 1950 | 1655 | 84.9 | 764,341 | 725,426 | 94.9 |
| 3 | 2923 | 2399 | 82.1 | 808,005 | 758,827 | 93.9 |
| 4 | 3911 | 3156 | 80.7 | 838,396 | 782,142 | 93.3 |
| 5 | 4882 | 3863 | 79.1 | 859,997 | 797,885 | 92.8 |
| ... | | | | | | |
| Entire Corpus (est.) | | | 66 to 69 | | | 89 to 90 |

Table 5
Scores and Frequency-Weighted Scores for the First 1000 Words of the Brown Corpus Translated by Various Versions of the Rules

| Version | No. of Rules* | No. of Words Scored | No. of Words Correct | Percent Correct | Total Frequency of Words Scored | Total Frequency of Correct Words | Percent Correct (Frequency Weighted) |
|---|---|---|---|---|---|---|---|
| 1 | 182 | 976 | 428 | 43.9 | 691,375 | 470,575 | 68.1 |
| 2 | 264 | 977 | 688 | 70.4 | 691,497 | 606,287 | 87.7 |
| 3 | 319 | 976 | 847 | 86.8 | 691,375 | 664,564 | 96.1 |

*These counts exclude rules for the ten digits and for all punctuation symbols except . , - ' ? and blank.

Table 6 gives version 3 of the English-to-IPA rules together with the statistics computed by STAT for the first 8000 words of the Corpus. The first column gives the number of distinct words that matched each rule. Column 2 is column 1 normalized to a total of 1. Column 3 gives the sum of the frequencies of the words matching each rule, and column 4 is column 3 normalized. Column 5 sums the number of texts in which the words occurred, and column 6 is column 5 normalized. If a rule was used more than once in translating a word, that word contributed more than once to the word count, frequency sum, and number-of-texts sum for the given rule. Table 7 is based on the 1000-word sample selected from the tail of the Corpus. Table 8 gives the rules for translation from IPA to Votrax codes, together with STAT results based on the first 8000 words as translated by version 3 of the rules.

Table 6
STAT Results for the First 8000 Words of the Brown Corpus
Translated by Version 3 of the Rules

| Rule | No. of Words Matched | | Total Frequencies of Words Matched | | Total No. of Texts for Words Matched | |
|---|---|---|---|---|---|---|
| | Abs. | Relative | Abs. | Relative | Abs. | Relative |
| *** ARULE *** | | | | | | |
| [A] =/AX/ | 94 | 0.0021493 | 26051 | 0.0090661 | 1697 | 0.0012855 |
| [ARE] =/AA R/ | 1 | 0.0000229 | 4393 | 0.0015288 | 453 | 0.0003431 |
| [AR]O=/AX R/ | 3 | 0.0000686 | 599 | 0.0002085 | 288 | 0.0002182 |
| [AR]#=/EH R/ | 151 | 0.0034526 | 6320 | 0.0021994 | 4020 | 0.0030451 |
| ^[AS]#=/EY S/ | 18 | 0.0004116 | 1334 | 0.0004642 | 766 | 0.0005802 |
| [A]WA=/AX/ | 10 | 0.0002286 | 728 | 0.0002534 | 417 | 0.0003159 |
| [AW]=/AO/ | 23 | 0.0005259 | 1256 | 0.0004371 | 719 | 0.0005446 |
| :[ANY]=/EH N IY/ | 9 | 0.0002058 | 2954 | 0.0010280 | 1201 | 0.0009097 |
| [A]^+#=/EY/ | 221 | 0.0050532 | 8369 | 0.0029125 | 4588 | 0.0034754 |
| #:[ALLY]=/AX L IY/ | 46 | 0.0010518 | 1920 | 0.0006682 | 1556 | 0.0011787 |
| [AL]#=/AX L/ | 17 | 0.0003887 | 898 | 0.0003125 | 578 | 0.0004378 |
| [AGAIN]=/AX G EH N/ | 2 | 0.0000457 | 1204 | 0.0004190 | 555 | 0.0004204 |
| #:[AG]E=/IH JH/ | 49 | 0.0011204 | 1799 | 0.0006261 | 1087 | 0.0008234 |
| [A]^+:#=/AE/ | 193 | 0.0044129 | 7458 | 0.0025955 | 4608 | 0.0034905 |
| :[A]^+ =/EY/ | 89 | 0.0020350 | 9944 | 0.0034606 | 4838 | 0.0036647 |
| [A]^%=/EY/ | 232 | 0.0053047 | 8750 | 0.0030451 | 5778 | 0.0043768 |
| [ARR]=/AX R/ | 13 | 0.0002972 | 329 | 0.0001145 | 276 | 0.0002091 |
| [ARR]=/AE R/ | 22 | 0.0005030 | 841 | 0.0002927 | 544 | 0.0004121 |
| :[AR] =/AA R/ | 7 | 0.0001601 | 849 | 0.0002955 | 408 | 0.0003091 |
| [AR] =/ER/ | 24 | 0.0005488 | 986 | 0.0003431 | 666 | 0.0005045 |
| [AR]=/AA R/ | 211 | 0.0048245 | 10137 | 0.0035278 | 5952 | 0.0045086 |
| [AIR]=/EH R/ | 27 | 0.0006174 | 1244 | 0.0004329 | 767 | 0.0005810 |
| [AI]=/EY/ | 163 | 0.0037270 | 6774 | 0.0023574 | 4490 | 0.0034011 |
| [AY]=/EY/ | 97 | 0.0022179 | 8739 | 0.0030413 | 4305 | 0.0032610 |
| [AU]=/AO/ | 59 | 0.0013490 | 2743 | 0.0009546 | 1512 | 0.0011453 |
| #:[AL] =/AX L/ | 201 | 0.0045959 | 11422 | 0.0039750 | 6166 | 0.0046707 |
| #:[ALS] =/AX L Z/ | 12 | 0.0002744 | 484 | 0.0001684 | 285 | 0.0002159 |
| [ALK]=/AO K/ | 10 | 0.0002286 | 694 | 0.0002415 | 483 | 0.0003659 |
| [AL]^=/AO L/ | 109 | 0.0024923 | 10348 | 0.0036012 | 4535 | 0.0034352 |
| :[ABLE]=/EY B AX L/ | 4 | 0.0000915 | 488 | 0.0001698 | 319 | 0.0002416 |
| [ABLE]=/AX B AX L/ | 45 | 0.0010289 | 1342 | 0.0004670 | 1005 | 0.0007613 |
| [ANG]+=/EY N JH/ | 29 | 0.0006631 | 1495 | 0.0005203 | 985 | 0.0007461 |
| [A]=/AE/ | 1482 | 0.0338859 | 118519 | 0.0412462 | 39864 | 0.0301967 |
| | 3673 | 0.0839831 | 261411 | 0.0909745 | 105711 | 0.0800752 |

Table 6 (continued)
STAT Results for the First 8000 Words of the Brown Corpus
Translated by Version 3 of the Rules

| Rule | No. of Words Matched | | Total Frequencies of Words Matched | | Total No. of Texts for Words Matched | |
|---|---|---|---|---|---|---|
| | Abs. | Relative | Abs. | Relative | Abs. | Relative |
| *** BRULE *** | | | | | | |
| [BE]^#=/B IH/ | 35 | 0.0008003 | 4727 | 0.0016451 | 2616 | 0.0019816 |
| [BEING]=/B IY IH NX/ | 2 | 0.0000457 | 748 | 0.0002603 | 361 | 0.0002735 |
| [BOTH] =/B OW TH/ | 1 | 0.0000229 | 730 | 0.0002540 | 337 | 0.0002553 |
| [BUS]#=/B IH Z/ | 4 | 0.0000915 | 484 | 0.0001684 | 237 | 0.0001795 |
| [BUIL]=/B IH L/ | 6 | 0.0001372 | 481 | 0.0001674 | 291 | 0.0002204 |
| [B]=/B/ | 729 | 0.0166686 | 50010 | 0.0174042 | 19966 | 0.0151241 |
| | ------ | -------- | ------ | -------- | ------ | -------- |
| | 777 | 0.0177661 | 57180 | 0.0198994 | 23808 | 0.0180344 |
| *** CRULE *** | | | | | | |
| [CH]^=/K/ | 9 | 0.0002058 | 392 | 0.0001364 | 138 | 0.0001045 |
| ^E[CH]=/K/ | 10 | 0.0002286 | 451 | 0.0001570 | 266 | 0.0002015 |
| [CH]=/CH/ | 215 | 0.0049160 | 16131 | 0.0056138 | 6955 | 0,0052684 |
| S[CI]#=/S AY/ | 5 | 0.0001143 | 305 | 0.0001061 | 151 | 0.0001144 |
| [CI]A=/SH/ | 35 | 0.0008003 | 1763 | 0.0006135 | 1077 | 0.0008158 |
| [CI]O=/SH/ | 10 | 0.0002286 | 230 | 0.0000800 | 173 | 0.0001310 |
| [CI]EN=/SH/ | 7 | 0.0001601 | 307 | 0.0001068 | 224 | 0.0001697 |
| [C]+=/S/ | 475 | 0.0108609 | 23550 | 0.0081957 | 14371 | 0.0108859 |
| [CK]=/K/ | 98 | 0.0022408 | 4217 | 0.0014676 | 2415 | 0.0018293 |
| [COM]%=/K AH M/ | 13 | 0.0002972 | 1706 | 0.0005937 | 1017 | 0.0007704 |
| [C]=/K/ | 1482 | 0.0338859 | 65195 | 0.0226887 | 38499 | 0.0291627 |
| | ------ | -------- | ------ | -------- | ------ | -------- |
| | 2359 | 0.0539385 | 114247 | 0.0397595 | 65286 | 0.0494536 |
| *** DRULE *** | | | | | | |
| #:[DED] =/D IH D/ | 51 | 0.0011661 | 1927 | 0.0006706 | 1540 | 0.0011665 |
| .E[D] =/D/ | 312 | 0.0071339 | 12985 | 0.0045190 | 9639 | 0.0073015 |
| #^:E[D] =/T/ | 140 | 0.0032011 | 6040 | 0.0021020 | 4502 | 0.0034102 |
| [DE]^#=/D IH/ | 124 | 0.0028353 | 4867 | 0.0016938 | 3158 | 0.0023922 |
| [DO] =/D UW/ | 1 | 0.0000229 | 1363 | 0.0004743 | 396 | 0.0003000 |
| [DOES]=/D AH Z/ | 2 | 0.0000457 | 572 | 0.0001991 | 318 | 0.0002409 |
| [DOING]=/D UW IH NX/ | 1 | 0.0000229 | 163 | 0.0000567 | 124 | 0.0000939 |
| [DOW]=/D AW/ | 4 | 0.0000915 | 964 | 0.0003355 | 340 | 0.0002575 |
| [DU]A=/JH UW/ | 12 | 0.0002744 | 503 | 0.0001750 | 330 | 0.0002500 |
| [D]=/D/ | 1301 | 0.0297473 | 102440 | 0.0356505 | 40797 | 0.0309034 |
| | ------ | -------- | ------ | -------- | ------ | -------- |
| | 1948 | 0.0445410 | 131824 | 0.0458765 | 61144 | 0.0463161 |

Table 6 (continued)
STAT Results for the First 8000 Words of the Brown Corpus
Translated by Version 3 of the Rules

| Rule | No. of Words Matched | | Total Frequencies of Words Matched | | Total No. of Texts for Words Matched | |
|---|---|---|---|---|---|---|
| | Abs. | Relative | Abs. | Relative | Abs. | Relative |
| *** ERULE *** | | | | | | |
| #:[E] =/ / | 1006 | 0.0230022 | 73857 | 0.0257032 | 38891 | 0.0294596 |
| ' ^:[E] =/ / | 7 | 0.0001601 | 519 | 0.0001806 | 315 | 0.0002386 |
| :[E] =/IY/ | 19 | 0.0004344 | 23483 | 0.0081724 | 2257 | 0.0017097 |
| #[ED] =/D/ | 14 | 0.0003201 | 641 | 0.0002231 | 495 | 0.0003750 |
| #:[E]D =/ / | 446 | 0.0101978 | 18502 | 0.0064389 | 13820 | 0.0104685 |
| [EV]ER=/EH V/ | 20 | 0.0004573 | 3258 | 0.0011338 | 1915 | 0.0014506 |
| [E]^%=/IY/ | 106 | 0.0024237 | 4302 | 0.0014972 | 3007 | 0.0022778 |
| [ERI]#=/IY R IY/ | 21 | 0.0004802 | 1508 | 0.0005248 | 873 | 0.0006613 |
| [ERI]=/EH R IH/ | 24 | 0.0005488 | 1423 | 0.0004952 | 724 | 0.0005484 |
| #:[ER]#=/ER/ | 115 | 0.0026295 | 6410 | 0.0022308 | 3657 | 0.0027701 |
| [ER]#=/EH R/ | 17 | 0.0003887 | 1110 | 0.0003863 | 532 | 0.0004030 |
| [ER]=/ER/ | 622 | 0.0142220 | 33594 | 0.0116912 | 18042 | 0.0136667 |
| [EVEN]=/IY V EH N/ | 7 | 0.0001601 | 1564 | 0.0005443 | 685 | 0.0005189 |
| #:[E]W=/ / | 10 | 0.0002286 | 173 | 0.0000602 | 127 | 0.0000962 |
| @[EW]=/UW/ | 20 | 0.0004573 | 2819 | 0.0009810 | 1019 | 0.0007719 |
| [EW]=/Y UW/ | 1 | 0.0000229 | 601 | 0.0002092 | 311 | 0.0002356 |
| [E]O=/IY/ | 27 | 0.0006174 | 792 | 0.0002756 | 426 | 0.0003227 |
| #:&[ES] =/IH Z/ | 116 | 0.0026523 | 4265 | 0.0014843 | 2599 | 0.0019687 |
| #:[E]S =/ / | 264 | 0.0060364 | 11065 | 0.0038508 | 6612 | 0.0050085 |
| #:[ELY] =/L IY/ | 45 | 0.0010289 | 1834 | 0.0006383 | 1461 | 0.0011067 |
| #:[EMENT]=/M EH N T/ | 37 | 0.0008460 | 1437 | 0.0005001 | 854 | 0.0006469 |
| [EFUL]=/F UH L/ | 7 | 0.0001601 | 281 | 0.0000978 | 233 | 0.0001765 |
| [EE]=/IY/ | 168 | 0.0038413 | 13544 | 0.0047135 | 6845 | 0.0051850 |
| [EARN]=/ER N/ | 8 | 0.0001829 | 345 | 0.0001201 | 268 | 0.0002030 |
| [EAR]^=/ER/ | 7 | 0.0001601 | 751 | 0.0002614 | 447 | 0.0003386 |
| [EAD]=/EH D/ | 29 | 0.0006631 | 2297 | 0.0007994 | 1517 | 0.0011491 |
| #:[EA] =/IY AX/ | 3 | 0.0000686 | 530 | 0.0001844 | 278 | 0.0002106 |
| [EA]SU=/EH/ | 9 | 0.0002058 | 440 | 0.0001531 | 260 | 0.0001969 |
| [EA]=/IY/ | 302 | 0.0069052 | 17378 | 0.0060478 | 10646 | 0.0080643 |
| [EIGH]=/EY/ | 16 | 0.0003658 | 534 | 0.0001858 | 358 | 0.0002712 |
| [EI]=/IY/ | 31 | 0.0007088 | 1349 | 0.0004695 | 849 | 0.0006431 |
| [EYE]=/AY/ | 3 | 0.0000686 | 533 | 0.0001855 | 238 | 0.0001803 |
| [EY]=/IY/ | 30 | 0.0006859 | 1169 | 0.0004068 | 604 | 0.0004575 |
| [EU]=/Y UW/ | 11 | 0.0002515 | 364 | 0.0001267 | 170 | 0.0001288 |
| [E]=/EH/ | 2065 | 0.0472162 | 95200 | 0.0331309 | 57157 | 0.0432960 |
| | ------- | --------- | ------- | --------- | ------- | --------- |
| | 5633 | 0.1287984 | 327872 | 0.1141039 | 178492 | 0.1352063 |

Table 6 (continued)
STAT Results for the First 8000 Words of the Brown Corpus
Translated by Version 3 of the Rules

| Rule | No. of Words Matched | | Total Frequencies of Words Matched | | Total No. of Texts for Words Matched | |
|---|---|---|---|---|---|---|
| | Abs. | Relative | Abs. | Relative | Abs. | Relative |
| *** FRULE *** | | | | | | |
| [FUL]=/F UH L/ | 29 | 0.0006631 | 1043 | 0.0003630 | 817 | 0.0006189 |
| [F]=/F/ | 736 | 0.0168286 | 58778 | 0.0204555 | 26736 | 0.0202523 |
| | 765 | 0.0174917 | 59821 | 0.0208185 | 27553 | 0.0208712 |
| *** GRULE *** | | | | | | |
| [GIV]=/G IH V/ | 6 | 0.0001372 | 1015 | 0.0003532 | 557 | 0.0004977 |
| [G]I^=/G/ | 8 | 0.0001829 | 475 | 0.0001653 | 225 | 0.0001704 |
| [GE]T=/G EH/ | 12 | 0.0002744 | 1504 | 0.0005234 | 788 | 0.0005969 |
| SU[GGES]=/G JH EH S/ | 6 | 0.0001372 | 258 | 0.0000898 | 223 | 0.0001689 |
| [GG]=/G/ | 20 | 0.0004573 | 399 | 0.0001389 | 287 | 0.0002174 |
| B#[G]=/G/ | 10 | 0.0002286 | 1102 | 0.0003835 | 694 | 0.0005257 |
| [G]+=/JH/ | 176 | 0.0040242 | 7355 | 0.0025596 | 4170 | 0.0031587 |
| [GREAT]=/G R EY T/ | 5 | 0.0001143 | 1014 | 0.0003529 | 546 | 0.0004136 |
| #[GH]=/ / | 11 | 0.0002515 | 522 | 0.0001817 | 366 | 0.0002772 |
| [G]=/G/ | 347 | 0.0079341 | 15701 | 0.0054642 | 8692 | 0.0065841 |
| | 601 | 0.0137419 | 29345 | 0.0102125 | 16648 | 0.0126107 |
| *** HRULE *** | | | | | | |
| [HAV]=/HH AE V/ | 5 | 0.0001143 | 4284 | 0.0014909 | 730 | 0.0005530 |
| [HERE]=/HH IY R/ | 2 | 0.0000457 | 761 | 0.0002648 | 325 | 0.0002462 |
| [HOUR]=/AW ER/ | 2 | 0.0000457 | 319 | 0.0001110 | 209 | 0.0001583 |
| [HOW]=/HH AW/ | 8 | 0.0001829 | 1583 | 0.0005509 | 744 | 0.0005636 |
| [H]#=/HH/ | 296 | 0.0067680 | 45711 | 0.0159080 | 11372 | 0.0086142 |
| [H]=/ / | 21 | 0.0004802 | 976 | 0.0003397 | 360 | 0.0002727 |
| | 334 | 0.0076369 | 53634 | 0.0186654 | 13740 | 0.0104079 |

Table 6 (continued)
STAT Results for the First 8000 Words of the Brown Corpus
Translated by Version 3 of the Rules

| Rule | No. of Words Matched | | Total Frequencies of Words Matched | | Total No. of Texts for Words Matched | |
|---|---|---|---|---|---|---|
| | Abs. | Relative | Abs. | Relative | Abs. | Relative |
| *** IRULE *** | | | | | | |
| [IN]=/IH N/ | 202 | 0.0046187 | 31259 | 0.0108786 | 6176 | 0.0046783 |
| [I] =/AY/ | 6 | 0.0001372 | 5894 | 0.0020512 | 692 | 0.0005242 |
| [IN]D=/AY N/ | 22 | 0.0005030 | 2022 | 0.0007037 | 1253 | 0.0009491 |
| [IER]=/IY ER/ | 11 | 0.0002515 | 419 | 0.0001458 | 282 | 0.0002136 |
| #:R[IED] =/IY D/ | 6 | 0.0001372 | 348 | 0.0001211 | 250 | 0.0001894 |
| [IED] =/AY D/ | 24 | 0.0005488 | 1009 | 0.0003511 | 769 | 0.0005825 |
| [IEN]=/IY EH N/ | 17 | 0.0003887 | 700 | 0.0002436 | 416 | 0.0003151 |
| [IE]T=/AY EH/ | 13 | 0.0002972 | 779 | 0.0002711 | 403 | 0.0003053 |
| :[I]%=/AY/ | 10 | 0.0002286 | 277 | 0.0000964 | 215 | 0.0001629 |
| [I]%=/IY/ | 88 | 0.0020121 | 2808 | 0.0009772 | 1520 | 0.0011514 |
| [IE]=/IY/ | 36 | 0.0008231 | 1811 | 0.0006303 | 1139 | 0.0008628 |
| [I]^+:#=/IH/ | 384 | 0.0087802 | 15196 | 0.0052884 | 9640 | 0.0073022 |
| [IR]#=/AY R/ | 51 | 0.0011661 | 2006 | 0.0006981 | 1378 | 0.0010438 |
| [IZ]%=/AY Z/ | 19 | 0.0004344 | 697 | 0.0002426 | 521 | 0.0003947 |
| [IS]%=/AY Z/ | 32 | 0.0007317 | 1027 | 0.0003574 | 799 | 0.0006052 |
| [I]D%=/AY/ | 40 | 0.0009146 | 2544 | 0.0008853 | 1710 | 0.0012953 |
| +^[I]^+=/IH/ | 74 | 0.0016920 | 2855 | 0.0009936 | 1737 | 0.0013158 |
| [I]T%=/AY/ | 24 | 0.0005488 | 2043 | 0.0007110 | 1119 | 0.0008476 |
| #^:[I]^+=/IH/ | 232 | 0.0053047 | 9645 | 0.0033566 | 5899 | 0.0044684 |
| [I]^+=/AY/ | 116 | 0.0026523 | 10713 | 0.0037283 | 5221 | 0.0039549 |
| [IR]=/ER/ | 42 | 0.0009603 | 3221 | 0.0011210 | 1603 | 0.0012143 |
| [IGH]=/AY/ | 55 | 0.0012576 | 4271 | 0.0014864 | 2451 | 0.0018566 |
| [ILD]=/AY L D/ | 11 | 0.0002515 | 810 | 0.0002819 | 382 | 0.0002894 |
| [IGN] =/AY N/ | 3 | 0.0000686 | 226 | 0.0000787 | 116 | 0.0000879 |
| [IGN]^=/AY N/ | 4 | 0.0000915 | 176 | 0.0000612 | 89 | 0.0000674 |
| [IGN]%=/AY N/ | 4 | 0.0000915 | 216 | 0.0000752 | 147 | 0.0001114 |
| [IQUE]=/IY K/ | 4 | 0.0000915 | 229 | 0.0000797 | 147 | 0.0001114 |
| [I]=/IH/ | 2038 | 0.0465988 | 128923 | 0.0448669 | 55411 | 0.0419734 |
| | 3568 | 0.0815823 | 232124 | 0.0807823 | 101485 | 0.0768741 |
| *** JRULE *** | | | | | | |
| [J]=/JH/ | 125 | 0.0028581 | 6066 | 0.0021110 | 3099 | 0.0023475 |
| | 125 | 0.0028581 | 6066 | 0.0021110 | 3099 | 0.0023475 |
| *** KRULE *** | | | | | | |
| [K]N=/ / | 13 | 0.0002972 | 1847 | 0.0006428 | 961 | 0.0007279 |
| [K]=/K/ | 224 | 0.0051218 | 13401 | 0.0046637 | 7403 | 0.0056077 |
| | 237 | 0.0054190 | 15248 | 0.0053065 | 8364 | 0.0063357 |

Table 6 (continued)
STAT Results for the First 8000 Words of the Brown Corpus
Translated by Version 3 of the Rules

| Rule | No. of Words Matched | | Total Frequencies of Words Matched | | Total No. of Texts for Words Matched | |
|---|---|---|---|---|---|---|
| | Abs. | Relative | Abs. | Relative | Abs. | Relative |
| *** LRULE *** | | | | | | |
| [LO]C#=/L OW/ | 9 | 0.0002058 | 514 | 0.0001789 | 265 | 0.0002007 |
| L[L]=/ / | 236 | 0.0053961 | 17526 | 0.0060993 | 8048 | 0.0060963 |
| #^:[L]%=/AX L/ | 108 | 0.0024694 | 6084 | 0.0021173 | 3428 | 0.0025967 |
| [LEAD]=/L IY D/ | 7 | 0.0001601 | 515 | 0.0001792 | 343 | 0.0002598 |
| [L]=/L/ | 1755 | 0.0401280 | 85646 | 0.0298060 | 49966 | 0.0378488 |
| | ------- | --------- | ------- | --------- | ------- | --------- |
| | 2115 | 0.0483594 | 110285 | 0.0383807 | 62050 | 0.0470024 |
| *** MRULE *** | | | | | | |
| [MOV]=/M UW V/ | 12 | 0.0002744 | 930 | 0.0003237 | 630 | 0.0004772 |
| [M]=/M/ | 1370 | 0.0313250 | 88465 | 0.0307870 | 42262 | 0.0320131 |
| | ------- | --------- | ------- | --------- | ------- | --------- |
| | 1382 | 0.0315994 | 89395 | 0.0311107 | 42892 | 0.0324903 |
| *** NRULE *** | | | | | | |
| E[NG]+=/N JH/ | 9 | 0.0002058 | 270 | 0.0000940 | 144 | 0.0001091 |
| [NG]R=/NX G/ | 9 | 0.0002058 | 353 | 0.0001228 | 164 | 0.0001242 |
| [NG]#=/NX G/ | 30 | 0.0006859 | 1036 | 0.0003605 | 704 | 0.0005333 |
| [NGL]%=/NX G AX L/ | 4 | 0.0000915 | 254 | 0.0000884 | 144 | 0.0001091 |
| [NG]=/NX/ | 526 | 0.0120270 | 23241 | 0.0080882 | 15847 | 0.0120040 |
| [NK]=/NX K/ | 38 | 0.0008689 | 1577 | 0.0005488 | 961 | 0.0007279 |
| [NOW] =/N AW/ | 1 | 0.0000229 | 1314 | 0.0004573 | 394 | 0.0002985 |
| [N]=/N/ | 2446 | 0.0559277 | 170584 | 0.0593655 | 73610 | 0.0557590 |
| | ------- | --------- | ------- | --------- | ------- | --------- |
| | 3063 | 0.0700354 | 198629 | 0.0691256 | 91968 | 0.0696650 |

Table 6 (continued)
STAT Results for the First 8000 Words of the Brown Corpus
Translated by Version 3 of the Rules

| Rule | No. of Words Matched | | Total Frequencies of Words Matched | | Total No. of Texts for Words Matched | |
|---|---|---|---|---|---|---|
| | Abs. | Relative | Abs. | Relative | Abs. | Relative |
| *** ORULE *** | | | | | | |
| [OF] =/AX V/ | 2 | 0.0000457 | 36427 | 0.0126771 | 509 | 0.0003856 |
| [OROUCH]=/ER OW/ | 2 | 0.0000457 | 61 | 0.0000212 | 57 | 0.0000432 |
| #:[OR] =/ER/ | 69 | 0.0015777 | 2711 | 0.0009435 | 1506 | 0.0011408 |
| #:[ORS] =/ER Z/ | 22 | 0.0005030 | 624 | 0.0002172 | 355 | 0.0002689 |
| [OR]=/AO R/ | 360 | 0.0082314 | 32460 | 0.0112965 | 11195 | 0.0084801 |
| [ONE]=/W AH N/ | 4 | 0.0000915 | 3487 | 0.0012135 | 637 | 0.0004825 |
| [OW]=/OW/ | 112 | 0.0025609 | 7450 | 0.0025927 | 4514 | 0.0034193 |
| [OVER]=/OW V ER/ | 9 | 0.0002058 | 1398 | 0.0004865 | 549 | 0.0004159 |
| [OV]=/AH V/ | 70 | 0.0016005 | 3713 | 0.0012922 | 2170 | 0.0016438 |
| [O]^%=/OW/ | 134 | 0.0030639 | 7003 | 0.0024371 | 4611 | 0.0034928 |
| [O]^EN=/OW/ | 32 | 0.0007317 | 1849 | 0.0006435 | 1217 | 0.0009219 |
| [O]^I#=/OW/ | 40 | 0.0009146 | 1728 | 0.0006014 | 842 | 0.0006378 |
| [OL]D=/OW L/ | 27 | 0.0006174 | 2161 | 0.0007521 | 1118 | 0.0008469 |
| [OUGHT]=/AO T/ | 9 | 0.0002058 | 1072 | 0.0003731 | 665 | 0.0005037 |
| [OUGH]=/AH F/ | 5 | 0.0001143 | 544 | 0.0001893 | 351 | 0.0002659 |
| [OU]=/AW/ | 15 | 0.0003430 | 3895 | 0.0013555 | 1104 | 0.0008363 |
| H[OU]S#=/AW/ | 8 | 0.0001829 | 932 | 0.0003243 | 425 | 0.0003219 |
| [OUS]=/AX S/ | 56 | 0.0012804 | 2031 | 0.0007068 | 1492 | 0.0011302 |
| [OUR]=/AO R/ | 28 | 0.0006402 | 1955 | 0.0006804 | 1139 | 0.0008628 |
| [OULD]=/UH D/ | 9 | 0.0002058 | 5649 | 0.0019659 | 1444 | 0.0010938 |
| ^[OU]^L=/AH/ | 10 | 0.0002286 | 443 | 0.0001542 | 330 | 0.0002500 |
| [OUP]=/UW P/ | 3 | 0.0000686 | 531 | 0.0001848 | 275 | 0.0002083 |
| [OU]=/AW/ | 107 | 0.0024466 | 8077 | 0.0028109 | 4168 | 0.0031572 |
| [OY]=/OY/ | 28 | 0.0006402 | 1137 | 0.0003957 | 685 | 0.0005189 |
| [OING]=/OW IH NX/ | 3 | 0.0000686 | 422 | 0.0001469 | 216 | 0.0001636 |
| [OI]=/OY/ | 42 | 0.0009603 | 1903 | 0.0006623 | 1230 | 0.0009317 |
| [OOR]=/AO R/ | 12 | 0.0002744 | 745 | 0.0002593 | 397 | 0.0003007 |
| [OOK]=/UH K/ | 13 | 0.0002972 | 1948 | 0.0006779 | 1097 | 0.0008310 |
| [OOD]=/UH D/ | 19 | 0.0004344 | 1847 | 0.0006428 | 901 | 0.0006825 |
| [OO]=/UW/ | 60 | 0.0013719 | 3764 | 0.0013099 | 1852 | 0.0014029 |
| [O]E=/OW/ | 20 | 0.0004573 | 772 | 0.0002687 | 360 | 0.0002727 |
| [O] =/OW/ | 49 | 0.0011204 | 7433 | 0.0025868 | 2319 | 0.0017566 |
| [OA]=/OW/ | 47 | 0.0010747 | 1964 | 0.0006835 | 1016 | 0.0007696 |
| [ONLY]=/OW N L IY/ | 1 | 0.0000229 | 1747 | 0.0006080 | 460 | 0.0003484 |
| [ONCE]=/W AH N S/ | 1 | 0.0000229 | 499 | 0.0001737 | 262 | 0.0001985 |
| [ON ' T]=/OW N T/ | 2 | 0.0000457 | 594 | 0.0002067 | 250 | 0.0001894 |
| C[O]N=/AA/ | 179 | 0.0040928 | 7030 | 0.0024465 | 4843 | 0.0036685 |
| [O]NG=/AO/ | 22 | 0.0005030 | 2475 | 0.0008613 | 1451 | 0.0010991 |
| ^:[O]N=/AH/ | 57 | 0.0013033 | 2364 | 0.0008227 | 1432 | 0.0010847 |
| I[ON]=/AX N/ | 362 | 0.0082771 | 14961 | 0.0052066 | 8533 | 0.0064637 |
| #:[ON] =/AX N/ | 70 | 0.0016005 | 2648 | 0.0009215 | 1286 | 0.0009741 |
| #^[ON]=/AX N/ | 23 | 0.0005259 | 691 | 0.0002405 | 459 | 0.0003477 |
| [O]ST =/OW/ | 8 | 0.0001829 | 2137 | 0.0007437 | 965 | 0.0007310 |
| [OF]^=/AO F/ | 17 | 0.0003887 | 2065 | 0.0007186 | 1161 | 0.0008794 |
| [OTHER]=/AH DH ER/ | 12 | 0.0002744 | 3231 | 0.0011244 | 1339 | 0.0010143 |
| [OSS] =/AO S/ | 6 | 0.0001372 | 520 | 0.0001810 | 327 | 0.0002477 |
| #^:[OM]=/AH M/ | 49 | 0.0011204 | 1627 | 0.0005662 | 931 | 0.0007052 |
| [O]=/AA/ | 850 | 0.0194352 | 51239 | 0.0178319 | 22065 | 0.0167141 |
| | 3085 | 0.0705385 | 241964 | 0.0842067 | 96510 | 0.0731055 |

Table 6 (continued)
STAT Results for the First 8000 Words of the Brown Corpus
Translated by Version 3 of the Rules

| Rule | No. of Words Matched | | Total Frequencies of Words Matched | | Total No. of Texts for Words Matched | |
|---|---|---|---|---|---|---|
| | Abs. | Relative | Abs. | Relative | Abs. | Relative |
| *** PRULE *** | | | | | | |
| [PH]=/F/ | 59 | 0.0013490 | 1717 | 0.0005975 | 1031 | 0.0007810 |
| [PEOP]=/P IY P/ | 3 | 0.0000686 | 902 | 0.0003139 | 326 | 0.0002469 |
| [POW]=/P AW/ | 6 | 0.0001372 | 535 | 0.0001862 | 277 | 0.0002098 |
| [PUT] =/P UH T/ | 3 | 0.0000686 | 492 | 0.0001712 | 271 | 0.0002053 |
| [P]=/P/ | 1556 | 0.0355779 | 69000 | 0.0240129 | 43119 | 0.0326623 |
| | 1627 | 0.0372013 | 72646 | 0.0252818 | 45024 | 0.0341053 |
| *** QRULE *** | | | | | | |
| [QUAR]=/K W AO R/ | 7 | 0.0001601 | 314 | 0.0001093 | 198 | 0.0001500 |
| [QU]=/K W/ | 76 | 0.0017377 | 3287 | 0.0011439 | 2233 | 0.0016915 |
| [Q]=/K/ | 2 | 0.0000457 | 35 | 0.0000122 | 3 | 0.0000023 |
| | 85 | 0.0019435 | 3636 | 0.0012654 | 2434 | 0.0018437 |
| *** RRULE *** | | | | | | |
| [RE]^#=/R IY/ | 186 | 0.0042529 | 8287 | 0.0028840 | 5727 | 0.0043382 |
| [R]=/R/ | 1497 | 0.0342289 | 73680 | 0.0256416 | 41537 | 0.0314639 |
| | 1683 | 0.0384818 | 81967 | 0.0285256 | 47264 | 0.0358021 |
| *** SRULE *** | | | | | | |
| [SH]=/SH/ | 177 | 0.0040471 | 10754 | 0.0037425 | 4981 | 0.0037731 |
| #[SION]=/ZH AX N/ | 23 | 0.0005259 | 972 | 0.0003383 | 643 | 0.0004871 |
| [SOME]=/S AH M/ | 12 | 0.0002744 | 2772 | 0.0009647 | 1162 | 0.0008802 |
| #[SUR]#=/ZH ER/ | 11 | 0.0002515 | 476 | 0.0001657 | 288 | 0.0002182 |
| [SUR]#=/SH ER/ | 10 | 0.0002286 | 709 | 0.0002467 | 452 | 0.0003424 |
| #[SU]#=/ZH UW/ | 5 | 0.0001143 | 416 | 0.0001448 | 291 | 0.0002204 |
| #[SSU]#=/SH UW/ | 5 | 0.0001143 | 322 | 0.0001121 | 178 | 0.0001348 |
| #[SED] =/Z D/ | 26 | 0.0005945 | 1686 | C.0005867 | 1090 | 0.0008257 |
| #[S]#=/Z/ | 271 | 0.0061964 | 13840 | 0.0048165 | 8563 | 0.0064864 |
| [SAID]=/S EH D/ | 1 | 0.0000229 | 1961 | 0.0006825 | 317 | 0.0002401 |
| ^[SION]=/SH AX N/ | 43 | 0.0009832 | 1415 | 0.0004924 | 912 | 0.0006908 |
| [S]S=/ / | 248 | 0.0056705 | 10255 | 0.0035689 | 6435 | 0.0048745 |
| .[S] =/Z/ | 512 | 0.0117069 | 21193 | 0.0073754 | 12390 | 0.0093853 |
| #:.E[S] =/Z/ | 138 | 0.0031554 | 5887 | 0.0020488 | 3662 | 0.0027739 |
| #^:##[S] =/Z/ | 107 | 0.0024466 | 4437 | 0.0015441 | 2487 | 0.0018839 |
| #^:#[S] =/S/ | 89 | 0.0020350 | 3773 | 0.0013131 | 1928 | 0.0014604 |
| U[S] =/S/ | 3 | 0.0000686 | 778 | 0.0002708 | 308 | 0.0002333 |
| :#[S] =/Z/ | 39 | 0.0008917 | 38870 | 0.0135273 | 3562 | 0.0026982 |
| [SCH]=/S K/ | 9 | 0.0002058 | 883 | 0.0003073 | 327 | 0.0002477 |
| [S]C+=/ / | 20 | 0.0004573 | 723 | 0.0002516 | 434 | 0.0003288 |
| #[SM]=/Z M/ | 26 | 0.0005945 | 514 | 0.0001789 | 289 | 0.0002189 |
| #[SN] '=/Z AX N/ | 3 | 0.0000686 | 271 | 0.0000943 | 162 | 0.0001227 |
| [S]=/S/ | 2063 | 0.0471705 | 104475 | 0.0363587 | 60294 | 0.0456722 |
| | 3841 | 0.0878244 | 227382 | 0.0791320 | 111155 | 0.0841990 |

Table 6 (continued)
STAT Results for the First 8000 Words of the Brown Corpus
Translated by Version 3 of the Rules

| Rule | No. of Words Matched | | Total Frequencies of Words Matched | | Total No. of Texts for Words Matched | |
|---|---|---|---|---|---|---|
| | Abs. | Relative | Abs. | Relative | Abs. | Relative |
| *** TRULE *** | | | | | | |
| [THE] =/DH AX/ | 1 | 0.0000229 | 69971 | 0.0243508 | 500 | 0.0003787 |
| [TO] =/T UW/ | 14 | 0.0003201 | 28177 | 0.0098060 | 1083 | 0.0008204 |
| [THAT] =/DH AE T/ | 2 | 0.0000457 | 10781 | 0.0037519 | 601 | 0.0004553 |
| [THIS] =/DH IH S/ | 1 | 0.0000229 | 5146 | 0.0017909 | 495 | 0.0003750 |
| [THEY]=/DH EY/ | 5 | 0.0001143 | 3761 | 0.0013089 | 575 | 0.0004356 |
| [THERE]=/DH EH R/ | 8 | 0.0001829 | 3142 | 0.0010935 | 741 | 0.0005613 |
| [THER]=/DH ER/ | 27 | 0.0006174 | 2408 | 0.0008380 | 1599 | 0.0012112 |
| [THEIR]=/DH EH R/ | 2 | 0.0000457 | 2691 | 0.0009365 | 484 | 0.0003666 |
| [THAN] =/DH AE N/ | 1 | 0.0000229 | 1789 | 0.0006226 | 456 | 0.0003454 |
| [THEM] =/DH EH M/ | 1 | 0.0000229 | 1789 | 0.0006226 | 429 | 0.0003250 |
| [THESE] =/DH IY Z/ | 1 | 0.0000229 | 1573 | 0.0005474 | 413 | 0.0003128 |
| [THEN]=/DH EH N/ | 1 | 0.0000229 | 1377 | 0.0004792 | 408 | 0.0003091 |
| [THROUGH]=/TH R UW/ | 2 | 0.0000457 | 1110 | 0.0003863 | 478 | 0.0003621 |
| [THOSE]=/DH OW Z/ | 1 | 0.0000229 | 850 | 0.0002958 | 367 | 0.0002780 |
| [THOUGH] =/DH OW/ | 2 | 0.0000457 | 761 | 0.0002648 | 439 | 0.0003325 |
| [THUS]=/DH AH S/ | 1 | 0.0000229 | 312 | 0.0001086 | 180 | 0.0001363 |
| [TH]=/TH/ | 191 | 0.0043672 | 19586 | 0.0068162 | 7526 | 0.0057009 |
| #:[TED] =/T IH D/ | 186 | 0.0042529 | 6418 | 0.0022335 | 4758 | 0.0036041 |
| S[TI]#N=/CH/ | 12 | 0.0002744 | 756 | 0.0002631 | 428 | 0.0003242 |
| [TI]O=/SH/ | 338 | 0.0077284 | 13438 | 0.0046766 | 7733 | 0.0058577 |
| [TI]A=/SH/ | 17 | 0.0003887 | 603 | 0.0002099 | 419 | 0.0003174 |
| [TIEN]=/SH AX N/ | 4 | 0.0000915 | 165 | 0.0000574 | 75 | 0.0000568 |
| [TUR]#=/CH ER/ | 55 | 0.0012576 | 2573 | 0.0008954 | 1519 | 0.0011506 |
| [TU]A=/CH UW/ | 15 | 0.0003430 | 858 | 0.0002986 | 579 | 0.0004386 |
| [TWO]=/T UW/ | 2 | 0.0000457 | 1424 | 0.0004956 | 440 | 0.0003333 |
| [T]=/T/ | 3064 | 0.0700583 | 183179 | 0.0637488 | 93605 | 0.0709050 |
| | ------ | --------- | ------ | --------- | ------ | --------- |
| | 3954 | 0.0904081 | 364638 | 0.1268939 | 126330 | 0.0956940 |
| *** URULE *** | | | | | | |
| [UN]I=/Y UW N/ | 15 | 0.0003430 | 1461 | 0.0005084 | 633 | 0.0004795 |
| [UN]=/AH N/ | 49 | 0.0011204 | 2462 | 0.0008568 | 1626 | 0.0012317 |
| [UPON]=/AX P AO N/ | 1 | 0.0000229 | 495 | 0.0001723 | 235 | 0.0001780 |
| @[UR]#=/UH R/ | 15 | 0.0003430 | 1084 | 0.0003772 | 555 | 0.0004204 |
| [UR]#=/Y UH R/ | 26 | 0.0005945 | 980 | 0.0003411 | 656 | 0.0004969 |
| [UR]=/ER/ | 109 | 0.0024923 | 4572 | 0.0015911 | 2832 | 0.0021452 |
| [U]^ =/AH/ | 70 | 0.0016005 | 9270 | 0.0032261 | 2461 | 0.0018642 |
| [U]^^=/AH/ | 366 | 0.0083686 | 17715 | 0.0061651 | 9963 | 0.0075469 |
| [UY]=/AY/ | 5 | 0.0001143 | 182 | 0.0000633 | 116 | 0.0000879 |
| G[U]#=/ / | 16 | 0.0003658 | 470 | 0.0001636 | 325 | 0.0002462 |
| G[U]%=/ / | 11 | 0.0002515 | 270 | 0.0000940 | 191 | 0.0001447 |
| G[U]#=/W/ | 9 | 0.0002058 | 278 | 0.0000967 | 172 | 0.0001303 |
| #N[U]=/Y UW/ | 25 | 0.0005716 | 1149 | 0.0003999 | 796 | 0.0006030 |
| @[U]=/UW/ | 198 | 0.0045273 | 7998 | 0.0027834 | 4884 | 0.0036996 |
| [U]=/Y UW/ | 149 | 0.0034069 | 7024 | 0.0024444 | 3952 | 0.0029936 |
| | ------ | --------- | ------ | --------- | ------ | --------- |
| | 1064 | 0.0243283 | 55410 | 0.0192834 | 29397 | 0.0222680 |

Table 6 (continued)
STAT Results for the First 8000 Words of the Brown Corpus
Translated by Version 3 of the Rules

| Rule | No. of Words Matched | | Total Frequencies of Words Matched | | Total No. of Texts for Words Matched | |
|---|---|---|---|---|---|---|
| | Abs. | Relative | Abs. | Relative | Abs. | Relative |
| *** VRULE *** | | | | | | |
| [VIEW]=/V Y UW/ | 9 | 0.0002058 | 411 | 0.0001430 | 278 | 0.0002106 |
| [V]=/V/ | 550 | 0.0125757 | 22264 | 0.0077482 | 14356 | 0.0108746 |
| | 559 | 0.0127815 | 22675 | 0.0078912 | 14634 | 0.0110851 |
| *** WRULE *** | | | | | | |
| [WERE]=/W ER/ | 2 | 0.0000457 | 3:06 | 0.0011505 | 473 | 0.0003583 |
| [WA]S=/W AA/ | 8 | 0.0001829 | 10343 | 0.0035995 | 736 | 0.0005575 |
| [WA]T=/W AA/ | 8 | 0.0001829 | 794 | 0.0002763 | 374 | 0.0002833 |
| [WHERE]=/WH EH R/ | 8 | 0.0001829 | 1216 | 0.0004232 | 614 | 0.0004651 |
| [WHAT]=/WH AA T/ | 4 | 0.0000915 | 2200 | 0.0007656 | 640 | 0.0004848 |
| [WHOL]=/HH OW L/ | 3 | 0.0000686 | 344 | 0.0001197 | 222 | 0.0001682 |
| [WHO]=/HH UW/ | 5 | 0.0001143 | 2681 | 0.0009330 | 713 | 0.0005401 |
| [WH]=/WH/ | 18 | 0.0004116 | 7925 | 0.0027580 | 2014 | 0.0015256 |
| [WAR]=/W AO R/ | 31 | 0.0007088 | 1372 | 0.0004775 | 784 | 0.0005939 |
| [WOR]^=/W ER/ | 25 | 0.0005716 | 3136 | 0.0010914 | 1621 | 0.0012279 |
| [WR]=/R/ | 12 | 0.0002744 | 961 | 0.0003344 | 558 | 0.0004227 |
| [W]=/W/ | 222 | 0.0050760 | 29047 | 0.0101087 | 9957 | 0.0075423 |
| | 346 | 0.0079113 | 63325 | 0.0220380 | 18706 | 0.0141696 |
| *** XRULE *** | | | | | | |
| [X]=/K S/ | 179 | 0.0040928 | 7242 | 0.0025203 | 4631 | 0.0035079 |
| | 1/9 | 0.0040928 | 7242 | 0.0025203 | 4631 | 0.0035079 |
| *** YRULE *** | | | | | | |
| [YOUNG]=/Y AH NX/ | 4 | 0.0000915 | 461 | 0.0001604 | 252 | 0.0001909 |
| [YOU]=/Y UW/ | 11 | 0.0002515 | 4749 | 0.0016527 | 832 | 0.0006302 |
| [YES]=/Y EH S/ | 2 | 0.0000457 | 227 | 0.0000790 | 127 | 0.0000962 |
| [Y]=/Y/ | 22 | 0.0005030 | 2764 | 0.0009619 | 1191 | 0.0009022 |
| #^:[Y] =/IY/ | 514 | 0.0117526 | 24405 | 0.0084933 | 15225 | 0.0115328 |
| #^:[Y]I=/IY/ | 8 | 0.0001829 | 245 | 0.0000853 | 211 | 0.0001598 |
| :[Y] =/AY/ | 10 | 0.0002286 | 7400 | 0.0025753 | 1218 | 0.0009226 |
| :[Y]#=/AY/ | 8 | 0.0001829 | 347 | 0.0001208 | 242 | 0.0001833 |
| :[Y]^+:#=/IH/ | 8 | 0.0001829 | 304 | 0.0001058 | 194 | 0.0001470 |
| :[Y]^#=/AY/ | 24 | 0.0005488 | 930 | 0.0003237 | 419 | 0.0003174 |
| [Y]=/IH/ | 63 | 0.0014405 | 2235 | 0.0007778 | 1145 | 0.0008673 |
| | 674 | 0.0154110 | 44067 | 0.0153359 | 21056 | 0.0159497 |
| *** ZRULE *** | | | | | | |
| [Z]=/Z/ | 58 | 0.0013262 | 1419 | 0.0004938 | 765 | 0.0005795 |
| | 58 | 0.0013262 | 1419 | 0.0004938 | 765 | 0.0005795 |
| | 43735 | 1.0 | 2873452 | 1.0 | 1320146 | 1.0 |

Table 7
STAT Results for the 1000-Word Sample from the Low-Frequency End of the Brown Corpus Translated by Version 3 of the Rules

| Rule | No. of Words Matched | | Total Frequencies of Words Matched | | Total No. of Texts for Words Matched | |
|---|---|---|---|---|---|---|
| | Abs. | Relative | Abs. | Relative | Abs. | Relative |
| *** ARULE *** | | | | | | |
| [A] =/AX/ | 30 | 0.0046069 | 41 | 0.0045480 | 34 | 0.0041474 |
| [ARE] =/AA R/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [AR]O=/AX R/ | 1 | 0.0001536 | 2 | 0.0002219 | 1 | 0.0001220 |
| [AR]#=/EH R/ | 10 | 0.0015356 | 15 | 0.0016639 | 14 | 0.0017077 |
| ^[AS]#=/EY S/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [A]WA=/AX/ | 2 | 0.0003071 | 3 | 0.0003328 | 3 | 0.0003659 |
| [AW]=/AO/ | 6 | 0.0009214 | 10 | 0.0011093 | 9 | 0.0010978 |
| :[ANY]=/EH N IY/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [A]^+#=/EY/ | 28 | 0.0042998 | 39 | 0.0043261 | 38 | 0.0046353 |
| #:[ALLY]=/AX L IY/ | 4 | 0.0006142 | 4 | 0.0004437 | 4 | 0.0004879 |
| [AL]#=/AX L/ | 1 | 0.0001536 | 2 | 0.0002219 | 1 | 0.0001220 |
| [AGAIN]=/AX G EH N/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| #:[AG]E=/IH JH/ | 3 | 0.0004607 | 4 | 0.0004437 | 4 | 0.0004879 |
| [A]^+:#=/AE/ | 39 | 0.0059889 | 56 | 0.0062119 | 52 | 0.0063430 |
| :[A]^+ =/EY/ | 6 | 0.0009214 | 8 | 0.0008874 | 7 | 0.0008539 |
| [A]^%=/EY/ | 39 | 0.0059889 | 56 | 0.0062119 | 51 | 0.0062210 |
| [ARR]=/AX R/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [ARR]=/AE R/ | 4 | 0.0006142 | 4 | 0.0004437 | 4 | 0.0004879 |
| :[AR] =/AA R/ | 0 | 0.0000000. | 0 | 0.0000000 | 0 | 0.0000000 |
| [AR] =/ER/ | 4 | 0.0006142 | 6 | 0.0006656 | 5 | 0.0006099 |
| [AR]=/AA R/ | 33 | 0.0050676 | 44 | 0.0048808 | 41 | 0.0050012 |
| [AIR]=/EH R/ | 5 | 0.0007678 | 7 | 0.0007765 | 7 | 0.0008539 |
| [AI]=/EY/ | 12 | 0.0018428 | 19 | 0.0021076 | 17 | 0.0020737 |
| [AY]=/EY/ | 12 | 0.0018428 | 18 | 0.0019967 | 17 | 0.0020737 |
| [AU]=/AO/ | 18 | 0.0027641 | 25 | 0.0027732 | 22 | 0.0026836 |
| #:[AL] =/AX L/ | 22 | 0.0033784 | 27 | 0.0029950 | 26 | 0.0031715 |
| #:[ALS] =/AX L Z/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [ALK]=/AO K/ | 1 | 0.0001536 | 1 | 0.0001109 | 1 | 0.0001220 |
| [AL]^=/AO L/ | 24 | 0.0036855 | 32 | 0.0035496 | 31 | 0.0037814 |
| :[ABLE]=/EY B AX L/ | 2 | 0.0003071 | 4 | 0.0004437 | 4 | 0.0004879 |
| [ABLE]=/AX B AX L/ | 4 | 0.0006142 | 5 | 0.0005546 | 5 | 0.0006099 |
| [ANG]+=/EY N JH/ | 1 | 0.0001536 | 2 | 0.0002219 | 1 | 0.0001220 |
| [A]=/AE/ | 263 | 0.0403870 | 366 | 0.0405990 | 325 | 0.0396438 |
| | -------- | --------- | -------- | --------- | -------- | --------- |
| | 574 | 0.0881450 | 800 | 0.0887410 | 724 | 0.0883142 |

Table 7 (continued)
STAT Results for the 1000-Word Sample from the Low-Frequency End of the Brown Corpus Translated by Version 3 of the Rules

| Rule | No. of Words Matched | | Total Frequencies of Words Matched | | Total No. of Texts for Words Matched | |
|---|---|---|---|---|---|---|
| | Abs. | Relative | Abs. | Relative | Abs. | Relative |
| *** BRULE *** | | | | | | |
| [BE]^#=/B IH/ | 5 | 0.0007678 | 7 | 0.0007765 | 5 | 0.0006099 |
| [BEING]=/B IY IH NX/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [BOTH] =/B OW TH/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [BUS]#=/B IH Z/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [BUIL]=/B IH L/ | 1 | 0.0001536 | 1 | 0.0001109 | 1 | 0.0001220 |
| [B]=/B/ | 146 | 0.0224201 | 207 | 0.0229617 | 189 | 0.0230544 |
| | ------- | --------- | ------- | --------- | ------- | --------- |
| | 152 | 0.0233415 | 215 | 0.0238491 | 195 | 0.0237863 |
| *** CRULE *** | | | | | | |
| [CH]^=/K/ | 1 | 0.0001536 | 2 | 0.0002219 | 1 | 0.0001220 |
| ^E[CH]=/K/ | 2 | 0.0003071 | 3 | 0.0003328 | 3 | 0.0003659 |
| [CH]=/CH/ | 38 | 0.0058354 | 57 | 0.0063228 | 45 | 0.0054891 |
| S[CI]#=/S AY/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [CI]A=/SH/ | 5 | 0.0007678 | 7 | 0.0007765 | 7 | 0.0008539 |
| [CI]O=/SH/ | 2 | 0.0003071 | 4 | 0.0004437 | 4 | 0.0004879 |
| [CI]EN=/SH/ | 1 | 0.0001536 | 1 | 0.0001109 | 1 | 0.0001220 |
| [C]+=/S/ | 47 | 0.0072174 | 66 | 0.0073211 | 60 | 0.0073189 |
| [CK]=/K/ | 33 | 0.0050676 | 46 | 0.0051026 | 45 | 0.0054891 |
| [COM]%=/K AH M/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [C]=/K/ | 174 | 0.0267199 | 251 | 0.0278425 | 225 | 0.0274457 |
| | ------- | --------- | ------- | --------- | ------- | --------- |
| | 303 | 0.0465295 | 437 | 0.0484748 | 391 | 0.0476946 |
| *** DRULE *** | | | | | | |
| #:[DED] =/D IH D/ | 8 | 0.0012285 | 12 | 0.0013311 | 11 | 0.0013418 |
| .E[D] =/D/ | 37 | 0.0056818 | 55 | 0.0061009 | 52 | 0.0063430 |
| #^:E[D] =/T/ | 12 | 0.0018428 | 18 | 0.0019967 | 18 | 0.0021957 |
| [DE]^#=/D IH/ | 13 | 0.0019963 | 19 | 0.0021076 | 17 | 0.0020737 |
| [DO] =/D UW/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [DOES]=/D AH Z/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [DOING]=/D UW IH NX/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [DOW]=/D AW/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [DU]A=/JH UW/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [D]=/D/ | 201 | 0.0308661 | 275 | 0.0305047 | 251 | 0.0306172 |
| | ------- | --------- | ------- | --------- | ------- | --------- |
| | 271 | 0.0416155 | 379 | 0.0420410 | 349 | 0.0425714 |

Table 7 (continued)
STAT Results for the 1000-Word Sample from the Low-Frequency End of the Brown Cc:pus Translated by Version 3 of the Rules

| Rule | No. of Words Matched | | Total Frequencies of Words Matched | | Total No. of Texts for Words Matched | |
|---|---|---|---|---|---|---|
| | Abs. | Relative | Abs. | Relative | Abs. | Relative |
| *** ERULE *** | | | | | | |
| #:[E] =/ / | 108 | 0.0165848 | 149 | 0.0165280 | 136 | 0.0165894 |
| ' ^:[E] =/ / | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| :[E] =/IY/ | 4 | 0.0006142 | 6 | 0.0006656 | 6 | 0.0007319 |
| #[ED] =/D/ | 2 | 0.0003071 | 3 | 0.0003328 | 3 | 0.0003659 |
| #:[E]D =/ / | 45 | 0.0069103 | 68 | 0.0075430 | 65 | 0.0079288 |
| [EV]ER=/EH V/ | 2 | 0.0003071 | 3 | 0.0003328 | 3 | 0.0003659 |
| [E]^%=/IY/ | 17 | 0.0026106 | 30 | 0.0033278 | 28 | 0.0034155 |
| [ERI]#=/IY R IY/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [ERI]=/EH R IH/ | 4 | 0.0006142 | 4 | 0.0004437 | 4 | 0.0004879 |
| #:[ER]#=/ER/ | 11 | 0.0016892 | 15 | 0.0016639 | 15 | 0.0018297 |
| [ER]#=/EH R/ | 2 | 0.0003071 | 4 | 0.0004437 | 4 | 0.0004879 |
| [ER]=/ER/ | 105 | 0.0161241 | 153 | 0.0169717 | 135 | 0.0164674 |
| [EVEN]=/IY V EH N/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| #:[E]W=/ / | 1 | 0.0001536 | 1 | 0.0001109 | 1 | 0.0001220 |
| @[EW]=/UW/ | 6 | 0.0009214 | 10 | 0.0011093 | 8 | 0.0009758 |
| [EW]=/Y UW/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [E]O=/IY/ | 6 | 0.0009214 | 9 | 0.0009983 | 8 | 0.0009758 |
| #:&[ES] =/IH Z/ | 11 | 0.0016892 | 15 | 0.0016639 | 12 | 0.0014638 |
| #:[E]S =/ / | 38 | 0.0058354 | 53 | 0.0058791 | 52 | 0.0063430 |
| #:[ELY] =/L IY/ | 5 | 0.0007678 | 8 | 0.0008874 | 8 | 0.0009758 |
| #:[EMENT]=/M EH N T/ | 2 | 0.0003071 | 3 | 0.0003328 | 3 | 0.0003659 |
| [EFUL]=/F UH L/ | 1 | 0.0001536 | 2 | 0.0002219 | 2 | 0.0002440 |
| [EE]=/IY/ | 22 | 0.0033784 | 29 | 0.0032169 | 25 | 0.0030495 |
| [EARN]=/ER N/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [EAR]^=/ER/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [EAD]=/EH D/ | 4 | 0.0006142 | 7 | 0.0007765 | 7 | 0.0008539 |
| #:[EA] =/IY AX/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [EA]SU=/EH/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [EA]=/IY/ | 36 | 0.0055283 | 46 | 0.0051026 | 44 | 0.0053672 |
| [EIGH]=/EY/ | 1 | 0.0001536 | 2 | 0.0002219 | 2 | 0.0002440 |
| [EI]=/IY/ | 5 | 0.0007678 | 7 | 0.0007765 | 6 | 0.0007319 |
| [EYE]=/AY/ | 1 | 0.0001536 | 1 | 0.0001109 | 1 | 0.0001220 |
| [EY]=/IY/ | 15 | 0.0023034 | 18 | 0.0019967 | 17 | 0.0020737 |
| [EU]=/Y UW/ | 7 | 0.0010749 | 10 | 0.0011093 | 8 | 0.0009758 |
| [E]=/EH/ | 301 | 0.0462224 | 403 | 0.0447033 | 369 | 0.0450110 |
| | 762 | 0.1170147 | 1059 | 0.1174709 | 972 | 0.1185655 |

Table 7 (continued)
STAT Results for the 1000-Word Sample from the Low-Frequency End of the Brown Corpus Translated by Version 3 of the Rules

| Rule | No. of Words Matched | | Total Frequencies of Words Matched | | Total No. of Texts for Words Matched | |
|---|---|---|---|---|---|---|
| | Abs. | Relative | Abs. | Relative | Abs. | Relative |
| *** FRULE *** | | | | | | |
| [FUL]=/F UH L/ | 4 | 0.0006142 | 7 | 0.0007765 | 7 | 0.0008539 |
| [F]=/F/ | 115 | 0.0176597 | 159 | 0.0176373 | 145 | 0.0176872 |
| | 119 | 0.0182740 | 166 | 0.0184138 | 152 | 0.0185411 |
| ***. GRULE *** | | | | | | |
| [GIV]=/G IH V/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [G]I^=/G/ | 3 | 0.0004607 | 4 | 0.0004437 | 4 | 0.0004879 |
| [GE]T=/G EH/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| SU[GGES]=/G JH EH S/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [GG]=/G/ | 4 | 0.0006142 | 5 | 0.0005546 | 4 | 0.0004879 |
| B#[G]=/G/ | 1 | 0.0001536 | 1 | 0.0001109 | 1 | 0.0001220 |
| [G]+=/JH/ | 34 | 0.0052211 | 49 | 0.0054354 | 43 | 0.0052452 |
| [GREAT]=/G R EY T/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| #[GH]=/ / | 1 | 0.0001536 | 1 | 0.0001109 | 1 | 0.0001220 |
| [G]=/G/ | 73 | 0.0112101 | 101 | 0.0112035 | 91 | 0.0111003 |
| | 116 | 0.0178133 | 161 | 0.0178591 | 144 | 0.0175653 |
| *** HRULE *** | | | | | | |
| [HAV]=/HH AE V/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [HERE]=/HH IY R/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [HOUR]=/AW ER/ | 1 | 0.0001536 | 2 | 0.0002219 | 1 | 0.0001220 |
| [HOW]=/HH AW/ | 1 | 0.0001536 | 2 | 0.0002219 | 1 | 0.0001220 |
| [H]#=/HH/ | 58 | 0.0089066 | 76 | 0.0084304 | 70 | 0.0085387 |
| [H]=/ / | 10 | 0.0015356 | 12 | 0.0013311 | 11 | 0.0013418 |
| | 70 | 0.0107494 | 92 | 0.0102052 | 83 | 0.0101244 |

Table 7 (continued)
STAT Results for the 1000-Word Sample from the Low-Frequency End of the Brown Corpus Translated by Version 3 of the Rules

| Rule | No. of Words Matched | | Total Frequencies of Words Matched | | Total No. of Texts for Words Matched | |
|---|---|---|---|---|---|---|
| | Abs. | Relative | Abs. | Relative | Abs. | Relative |
| *** IRULE *** | | | | | | |
| [IN]=/IH N/ | 27 | 0.0041462 | 36 | 0.0039933 | 35 | 0.0042693 |
| [I] =/AY/ | 2 | 0.0003071 | 3 | 0.0003328 | 2 | 0.0002440 |
| [IN]D=/AY N/ | 4 | 0.0006142 | 6 | 0.0006656 | 4 | 0.0004879 |
| [IER]=/IY ER/ | 6 | 0.0009214 | 9 | 0.0009983 | 8 | 0.0009758 |
| #:R[IED] =/IY D/ | 1 | 0.0001536 | 1 | 0.0001109 | 1 | 0.0001220 |
| [IED] =/AY D/ | 7 | 0.0010749 | 7 | 0.0007765 | 7 | 0.0008539 |
| [IEN]=/IY EH N/ | 1 | 0.0001536 | 1 | 0.0001109 | 1 | 0.0001220 |
| [IE]T=/AY EH/ | 2 | 0.0003071 | 3 | 0.0003328 | 3 | 0.0003659 |
| :[I]%=/AY/ | 1 | 0.0001536 | 1 | 0.0001109 | 1 | 0.0001220 |
| [I]%=/IY/ | 17 | 0.0026106 | 27 | 0.0029950 | 27 | 0.0032935 |
| [IE]=/IY/ | 11 | 0.0016892 | 16 | 0.0017748 | 14 | 0.0017077 |
| [I]^+:#=/IH/ | 56 | 0.0085995 | 71 | 0.0078758 | 67 | 0.0081727 |
| [IR]#=/AY R/ | 4 | 0.0006142 | 6 | 0.0006656 | 6 | 0.0007319 |
| [IZ]%=/AY Z/ | 7 | 0.0010749 | 9 | 0.0009983 | 8 | 0.0009758 |
| [IS]%=/AY Z/ | 4 | 0.0006142 | 5 | 0.0005546 | 5 | 0.0006099 |
| [I]D%=/AY/ | 4 | 0.0006142 | 7 | 0.0007765 | 6 | 0.0007319 |
| +^[I]^+=/IH/ | 11 | 0.0016892 | 16 | 0.0017748 | 14 | 0.0017077 |
| [I]T%=/AY/ | 6 | 0.0009214 | 7 | 0.0007765 | 7 | 0.0008539 |
| #^:[I]^+=/IH/ | 20 | 0.0030713 | 29 | 0.0032169 | 25 | 0.0030495 |
| [I]^+=/AY/ | 16 | 0.0024570 | 25 | 0.0027732 | 20 | 0.0024396 |
| [IR]=/ER/ | 12 | 0.0018428 | 18 | 0.0019967 | 16 | 0.0019517 |
| [IGH]=/AY/ | 6 | 0.0009214 | 8 | 0.0008874 | 7 | 0.0008539 |
| [ILD]=/AY L D/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [IGN] =/AY N/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [IGN]^=/AY N/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [IGN]%=/AY N/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [IQUE]=/IY K/ | 3 | 0.0004607 | 6 | 0.0006656 | 5 | 0.0006099 |
| [I]=/IH/ | 356 | 0.0546683 | 493 | 0.0546866 | 445 | 0.0542815 |
| | 584 | 0.0896806 | 810 | 0.0898502 | 734 | 0.0895340 |
| *** JRULE *** | | | | | | |
| [J]=/JH/ | 20 | 0.0030713 | 28 | 0.0031059 | 24 | 0.0029275 |
| | 20 | 0.0030713 | 28 | 0.0031059 | 24 | 0.0029275 |
| *** KRULE *** | | | | | | |
| [K]N=/ / | 2 | 0.0003071 | 3 | 0.0003328 | 3 | 0.0003659 |
| [K]=/K/ | 62 | 0.0095209 | 81 | 0.0089850 | 71 | 0.0086606 |
| | 64 | 0.0098280 | 84 | 0.0093178 | 74 | 0.0090266 |

Table 7 (continued)
STAT Results for the 100C-Word Sample from the Low-Frequency End of the Brown Corpus Translated by Version 3 of the Rules

| Rule | No. of Words Matched | | Total Frᵔquencies of Words Matched | | Total No. of Texts for Words Matched | |
|---|---|---|---|---|---|---|
| | Abs. | Relative | Abs. | Relative | Abs. | Relative |
| *** LRULE *** | | | | | | |
| [LO]C#=/L OW/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| L[L]=/ / | 38 | 0.0058354 | 51 | 0.0056572 | 50 | 0.0060990 |
| #^:[L]%=/AX L/ | 21 | 0.0032248 | 26 | 0.0028841 | 23 | 0.0028056 |
| [LEAD]=/L IY D/ | 2 | 0.0003071 | 2 | 0.0002219 | 2 | 0.0002440 |
| [L]=/L/ | 297 | 0.0456081 | 422 | 0.0468109 | 387 | 0.0472066 |
| | ------- | --------- | ------- | --------- | ------- | --------- |
| | 358 | 0.0549754 | 501 | 0.0555740 | 462 | 0.0563552 |
| *** MRULE *** | | | | | | |
| [MOV]=/M UW V/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [M]=/M/ | 231 | 0.0354730 | 317 | 0.0351636 | 287 | 0.0350085 |
| | ------- | --------- | ------- | --------- | ------- | --------- |
| | 231 | 0.0354730 | 317 | 0.0351636 | 287 | 0.0350085 |
| *** NRULE *** | | | | | | |
| E[NG]+=/N JH/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [NG]R=/NX G/ | 2 | 0.0003071 | 3 | 0.0003328 | 2 | 0.0002440 |
| [NG]#=/NX G/ | 6 | 0.0009214 | 8 | 0.0008874 | 6 | 0.0007319 |
| [NGL]%=/NX G AX L/ | 2 | 0.0003071 | 3 | 0.0003328 | 3 | 0.0003659 |
| [NG]=/NX/ | 84 | 0.0128993 | 113 | 0.0125347 | 110 | 0.0134179 |
| [NK]=/NX K/ | 8 | 0.0012285 | 11 | 0.0012202 | 10 | 0.0012198 |
| [NOW] =/N AW/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [N]=/N/ | 359 | 0.0551290 | 490 | 0.0543539 | 443 | 0.0540376 |
| | ------- | --------- | ------- | --------- | ------- | --------- |
| | 461 | 0.0707924 | 628 | 0.0696617 | 574 | 0.0700171 |

Table 7 (continued)
STAT Results for the 1000-Word Sample from the Low-Frequency End of the Brown Corpus Translated by Version 3 of the Rules

| Rule | No. of Words Matched | | Total Frequencies of Words Matched | | Total No. of Texts for Words Matched | |
|---|---|---|---|---|---|---|
| | Abs. | Relative | Abs. | Relative | Abs. | Relative |
| *** ORULE *** | | | | | | |
| [OF] =/AX V/ | 2 | 0.0003071 | 3 | 0.0003328 | 3 | 0.0003659 |
| [OROUGH]=/ER OW/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| #:[OR] =/ER/ | 4 | 0.0006142 | 5 | 0.0005546 | 4 | 0.0004879 |
| #:[ORS] =/ER Z/ | 3 | 0.0004607 | 5 | 0.0005546 | 5 | 0.0006099 |
| [OR]=/AO R/ | 55 | 0.0084459 | 87 | 0.0096506 | 72 | 0.0087826 |
| [ONE]=/W AH N/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [OW]=/OW/ | 25 | 0.0038391 | 31 | 0.0034387 | 29 | 0.0035374 |
| [OVER]=/OW V ER/ | 4 | 0.0006142 | 5 | 0.0005546 | 5 | 0.0006099 |
| [OV]=/AH V/ | 11 | 0.0016892 | 17 | 0.0018857 | 13 | 0.0015858 |
| [O]^%=/OW/ | 11 | 0.0016892 | 13 | 0.0014420 | 13 | 0.0015858 |
| [O]^EN=/OW/ | 5 | 0.0007678 | 7 | 0.0007765 | 7 | 0.0008539 |
| [O]^I#=/OW/ | 12 | 0.0018428 | 19 | 0.0021076 | 16 | 0.0019517 |
| [OL]D=/OW L/ | 2 | 0.0003071 | 2 | 0.0002219 | 2 | 0.0002440 |
| [OUGHT]=/AO T/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [OUGH]=/AH F/ | 1 | 0.0001536 | 2 | 0.0002219 | 2 | 0.0002440 |
| [OU]=/AW/ | 3 | 0.0004607 | 5 | 0.0005546 | 4 | 0.0004879 |
| H[OU]S#=/AW/ | 4 | 0.0006142 | 4 | 0.0004437 | 4 | 0.0004879 |
| [OUS]=/AX S/ | 8 | 0.0012285 | 11 | 0.0012202 | 11 | 0.0013418 |
| [OUR]=/AO R/ | 3 | 0.0004607 | 5 | 0.0005546 | 5 | 0.0006099 |
| [OULD]=/UH D/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| ^[OU]^L=/AH/ | 1 | 0.0001536 | 1 | 0.0001109 | 1 | 0.0001220 |
| [OUP]=/UW P/ | 1 | 0.0001536 | 1 | 0.0001109 | 1 | 0.0001220 |
| [OU]=/AW/ | 16 | 0.0024570 | 23 | 0.0025513 | 22 | 0.0026836 |
| [OY]=/OY/ | 3 | 0.0004607 | 4 | 0.0004437 | 4 | 0.0004879 |
| [OING]=/OW IH NX/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [OI]=/OY/ | 9 | 0.0013821 | 10 | 0.0011093 | 9 | 0.0010978 |
| [OOR]=/AO R/ | 2 | 0.0003071 | 3 | 0.0003328 | 3 | 0.0003659 |
| [OOK]=/UH K/ | 3 | 0.0004607 | 4 | 0.0004437 | 4 | 0.0004879 |
| [OOD]=/UH D/ | 3 | 0.0004607 | 5 | 0.0005546 | 5 | 0.0006099 |
| [OO]=/UW/ | 16 | 0.0024570 | 18 | 0.0019967 | 18 | 0.0021957 |
| [O]E=/OW/ | 1 | 0.0001536 | 1 | 0.0001109 | 1 | 0.0001220 |
| [O] =/OW/ | 32 | 0.0049140 | 44 | 0.0048808 | 36 | 0.0043913 |
| [OA]=/OW/ | 8 | 0.0012285 | 9 | 0.0009983 | 9 | 0.0010978 |
| [ONLY]=/OW N L IY/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [ONCE]=/W AH N S/ | 1 | 0.0001536 | 1 | 0.0001109 | 1 | 0.0001220 |
| [ON ' T]=/OW N T/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| C[O]N=/AA/ | 17 | 0.0026106 | 24 | 0.0026622 | 21 | 0.0025616 |
| [O]NG=/AO/ | 3 | 0.0004607 | 4 | 0.0004437 | 4 | 0.0004879 |
| ^:[O]N=/AH/ | 14 | 0.0021499 | 21 | 0.0023294 | 20 | 0.0024396 |
| I[ON]=/AX N/ | 25 | 0.0038391 | 33 | 0.0036606 | 32 | 0.0039034 |
| #:[ON] =/AX N/ | 19 | 0.0029177 | 26 | 0.0028841 | 25 | 0.0030495 |
| #^[ON]=/AX N/ | 10 | 0.0015356 | 14 | 0.0015530 | 12 | 0.0014638 |
| [O]ST =/OW/ | 1 | 0.0001536 | 1 | 0.0001109 | 1 | 0.0001220 |
| [OF]^=/AO F/ | 2 | 0.0003071 | 2 | 0.0002219 | 2 | 0.0002440 |
| [OTHER]=/AH DH ER/ | 1 | 0.0001536 | 1 | 0.0001109 | 1 | 0.0001220 |
| [OSS] =/AO S/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| #^:[OM]=/AH M/ | 8 | 0.0012285 | 13 | 0.0014420 | 10 | 0.0012198 |
| [O]=/AA/ | 122 | 0.0187346 | 165 | 0.0183028 | 151 | 0.0184191 |
| | 471 | 0.0723280 | 649 | 0.0719911 | 588 | 0.0717248 |

Table 7 (continued)
STAT Results for the 1000-Word Sample from the Low-Frequency End of the Brown Corpus Translated by Version 3 of the Rules

| Rule | No. of Words Matched | | Total Frequencies of Words Matched | | Total No. of Texts for Words Matched | |
|---|---|---|---|---|---|---|
| | Abs. | Relative | Abs. | Relative | Abs. | Relative |
| *** PRULE *** | | | | | | |
| [PH]=/F/ | 21 | 0.0032248 | 29 | 0.0032169 | 27 | 0.0032935 |
| [PEOP]=/P IY P/ | 1 | 0.0001536 | 1 | 0.0001109 | 1 | 0.0001220 |
| [POW]=/P AW/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [PUT] =/P UH T/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [P]=/P/ | 194 | 0.0297912 | 269 | 0.0298392 | 241 | 0.0293974 |
| | 216 | 0.0331695 | 299 | 0.0331669 | 269 | 0.0328129 |
| *** QRULE *** | | | | | | |
| [QUAR]=/K W AO R/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [QU]=/K W/ | 10 | 0.0015356 | 14 | 0.0015530 | 13 | 0.0015858 |
| [Q]=/K/ | 1 | 0.0001536 | 2 | 0.0002219 | 2 | 0.0002440 |
| | 11 | 0.0016892 | 16 | 0.0017748 | 15 | 0.0018297 |
| *** RRULE *** | | | | | | |
| [RE]^#=/R IY/ | 14 | 0.0021499 | 19 | 0.0021076 | 18 | 0.0021957 |
| [R]=/R/ | 228 | 0.0350123 | 315 | 0.0349418 | 280 | 0.0341547 |
| | 242 | 0.0371622 | 334 | 0.0370494 | 298 | 0.0363503 |
| *** SRULE *** | | | | | | |
| [SH]=/SH/ | 25 | 0.0038391 | 31 | 0.0034387 | 30 | 0.0036594 |
| #[SION]=/ZH AX N/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [SOME]=/S AH M/ | 2 | 0.0003071 | 4 | 0.0004437 | 3 | 0.0003659 |
| #[SUR]#=/ZH ER/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [SUR]#=/SH ER/ | 1 | 0.0001536 | 2 | 0.0002219 | 2 | 0.0002440 |
| #[SU]#=/ZH UW/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| #[SSU]#=/SH UW/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| #[SED] =/Z D/ | 3 | 0.0004607 | 4 | 0.0004437 | 4 | 0.0004879 |
| #[S]#=/Z/ | 43 | 0.0066032 | 59 | 0.0065446 | 56 | 0.0068309 |
| [SAID]=/S EH D/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| ^[SION]=/SH AX N/ | 5 | 0.0007678 | 7 | 0.0007765 | 6 | 0.0007319 |
| [S]S=/ / | 33 | 0.0050676 | 42 | 0.0046589 | 39 | 0.0047573 |
| .[S] =/Z/ | 63 | 0.0096744 | 85 | 0.0094287 | 77 | 0.0093925 |
| #:.E[S] =/Z/ | 16 | 0.0024570 | 22 | 0.0024404 | 21 | 0.0025616 |
| #^:##[S] =/Z/ | 20 | 0.0030713 | 33 | 0.0036606 | 31 | 0.0037814 |
| #^:#[S] =/S/ | 19 | 0.0029177 | 25 | 0.0027732 | 21 | 0.0025616 |
| U[S] =/S/ | 1 | 0.0001536 | 2 | 0.0002219 | 2 | 0.0002440 |
| :#[S] =/Z/ | 8 | 0.0012285 | 10 | 0.0011093 | 9 | 0.0010978 |
| [SCH]=/S K/ | 2 | 0.0003071 | 3 | 0.0003328 | 2 | 0.0002440 |
| [S]C+=/ / | 4 | 0.0006142 | 6 | 0.0006656 | 5 | 0.0006099 |
| #[SM]=/Z M/ | 7 | 0.0010749 | 11 | 0.0012202 | 10 | 0.0012198 |
| #[SN] '=/Z AX N/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [S]=/S/ | 307 | 0.0471437 | 418 | 0.0463672 | 378 | 0.0461088 |
| | 559 | 0.0858415 | 764 | 0.0847476 | 696 | 0.0848988 |

Table 7 (continued)
STAT Results for the 1000-Word Sample from the Low-Frequency End of the Brown Corpus Translated by Version 3 of the Rules

| Rule | No. of Words Matched | | Total Frequencies of Words Matched | | Total No. of Texts for Words Matched | |
|---|---|---|---|---|---|---|
| | Abs. | Relative | Abs. | Relative | Abs. | Relative |
| *** TRULE *** | | | | | | |
| [THE] =/DH AX/ | 2 | 0.0003071 | 3 | 0.0003328 | 3 | 0.0003659 |
| [TO] =/T UW/ | 2 | 0.0003071 | 3 | 0.0003328 | 2 | 0.0002440 |
| [THAT] =/DH AE T/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [THIS] =/DH IH S/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [THEY]=/DH EY/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [THERE]=/DH EH R/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [THER]=/DH ER/ | 3 | 0.0004607 | 5 | 0.0005546 | 5 | 0.0006099 |
| [THEIR]=/DH EH R/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [THAN] =/DH AE N/ | 1 | 0.0001536 | 2 | 0.0002219 | 2 | 0.0002440 |
| [THEM] =/DH EH M/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [THESE] =/DH IY Z/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [THEN]=/DH EH N/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [THROUGH]=/TH R UW/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [THOSE]=/DH OW Z/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [THOUGH] =/DH OW/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [THUS]=/DH AH S/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [TH]=/TH/ | 28 | 0.0042998 | 37 | 0.0041043 | 35 | 0.0042693 |
| #:[TED] =/T IH D/ | 11 | 0.0016892 | 17 | 0.0018857 | 15 | 0.0018297 |
| S[TI]#N=/CH/ | 1 | 0.0001536 | 1 | 0.0001109 | 1 | 0.0001220 |
| [TI]O=/SH/ | 20 | 0.0030713 | 28 | 0.0031059 | 27 | 0.0032935 |
| [TI]A=/SH/ | 2 | 0.0003071 | 2 | 0.0002219 | 2 | 0.0002440 |
| [TIEN]=/SH AX N/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [TUR]#=/CH ER/ | 3 | 0.0004607 | 6 | 0.0006656 | 4 | 0.0004879 |
| [TU]A=/CH UW/ | 4 | 0.0006142 | 6 | 0.0006656 | 6 | 0.0007319 |
| [TWO]=/T UW/ | 2 | 0.0003071 | 2 | 0.0002219 | 2 | 0.0002440 |
| [T]=/T/ | 406 | 0.0623464 | 555 | 0.0615641 | 506 | 0.0617224 |
| | ------- | --------- | ------- | --------- | ------- | --------- |
| | 485 | 0.0744779 | 667 | 0.0739878 | 610 | 0.0744084 |
| *** URULE *** | | | | | | |
| [UN]I=/Y UW N/ | 2 | 0.0003071 | 3 | 0.0003328 | 3 | 0.0003659 |
| [UN]=/AH N/ | 17 | 0.0026106 | 23 | 0.0025513 | 21 | 0.0025616 |
| [UPON]=/AX P AO N/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| @[UR]#=/UH R/ | 4 | 0.0006142 | 5 | 0.0005546 | 4 | 0.0004879 |
| [UR]#=/Y UH R/ | 2 | 0.0003071 | 3 | 0.0003328 | 3 | 0.0003659 |
| [UR]=/ER/ | 17 | 0.0026106 | 23 | 0.0025513 | 20 | 0.0024396 |
| [U]^ =/AH/ | 13 | 0.0019963 | 16 | 0.0017748 | 15 | 0.0018297 |
| [U]^^=/AH/ | 59 | 0.0090602 | 84 | 0.0093178 | 76 | 0.0092706 |
| [UY]=/AY/ | 2 | 0.0003071 | 3 | 0.0003328 | 3 | 0.0003659 |
| G[U]#=/ / | 1 | 0.0001536 | 1 | 0.0001109 | 1 | 0.0001220 |
| G[U]%=/ / | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| G[U]#=/W/ | 2 | 0.0003071 | 2 | 0.0002219 | 2 | 0.0002440 |
| #N[U]=/Y UW/ | 3 | 0.0004607 | 3 | 0.0003328 | 3 | 0.0003659 |
| @[U]=/UW/ | 23 | 0.0035319 | 31 | 0.0034387 | 27 | 0.0032935 |
| [U]=/Y UW/ | 19 | 0.0029177 | 27 | 0.0029950 | 25 | 0.0030495 |
| | ------- | --------- | ------- | --------- | ------- | --------- |
| | 164 | 0.0251843 | 224 | 0.0248475 | 203 | 0.0247621 |

Table 7 (continued)
STAT Results for the 1000-Word Sample from the Low-Frequency End of the Brown Corpus Translated by Version 3 of the Rules

| Rule | No. of Words Matched | | Total Frequencies of Words Matched | | Total No. of Texts for Words Matched | |
|---|---|---|---|---|---|---|
| | Abs. | Relative | Abs. | Relative | Abs. | Relative |
| *** VRULE *** | | | | | | |
| [VIEW]=/V Y UW/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [V]=/V/ | 66 | 0.0101351 | 91 | 0.0100943 | 85 | 0.0103684 |
| | ------- | --------- | ------- | --------- | ------- | --------- |
| | 66 | 0.0101351 | 91 | 0.0100943 | 85 | 0.0103684 |
| *** WRULE *** | | | | | | |
| [WERE]=/W ER/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [WA]S=/K AA/ | 1 | 0.0001536 | 1 | 0.0001109 | 1 | 0.0001220 |
| [WA]T=/W AA/ | 2 | 0.0003071 | 3 | 0.0003328 | 3 | 0.0003659 |
| [WHERE]=/WH EH R/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [WHAT]=/WH AA T/ | 1 | 0.0001536 | 1 | 0.0001109 | 1 | 0.0001220 |
| [WHOL]=/HH OW L/ | 1 | 0.0001536 | 1 | 0.0001109 | 1 | 0.0001220 |
| [WHO]=/HH UW/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [WH]=/WH/ | 5 | 0.0007678 | 7 | 0.0007765 | 6 | 0.0007319 |
| [WAR]=/W AO R/ | 2 | 0.0003071 | 3 | 0.0003328 | 2 | 0.0002440 |
| [WOR]^=/W ER/ | 7 | 0.0010749 | 8 | 0.0008874 | 8 | 0.0009758 |
| [WR]=/R/ | 1 | 0.0001536 | 1 | 0.0001109 | 1 | 0.0001220 |
| [W]=/W/ | 41 | 0.0062961 | 56 | 0.0062119 | 55 | 0.0067090 |
| | ------- | --------- | ------- | --------- | ------- | --------- |
| | 61 | 0.0093673 | 81 | 0.0089850 | 78 | 0.0095145 |
| *** XRULE *** | | | | | | |
| [X]=/K S/ | 19 | 0.0029177 | 26 | 0.0028841 | 23 | 0.0028056 |
| | ------- | --------- | ------- | --------- | ------- | --------- |
| | 19 | 0.0029177 | 26 | 0.0028841 | 23 | 0.0028056 |
| *** YRULE *** | | | | | | |
| [YOUNG]=/Y AH NX/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [YOU]=/Y UW/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [YES]=/Y EH S/ | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [Y]=/Y/ | 4 | 0.0006142 | 6 | 0.0006656 | 6 | 0.0007319 |
| #^:[Y] =/IY/ | 67 | 0.0102887 | 97 | 0.0107598 | 91 | 0.0111003 |
| #^:[Y]I=/IY/ | 3 | 0.0004607 | 3 | 0.0003328 | 3 | 0.0003659 |
| :[Y] =/AY/ | 2 | 0.0003071 | 2 | 0.0002219 | 2 | 0.0002440 |
| :[Y]#=/AY/ | 2 | 0.0003071 | 3 | 0.0003328 | 2 | 0.0002440 |
| :[Y]^+:#=/IH/ | 4 | 0.0006142 | 6 | 0.0006656 | 5 | 0.0006099 |
| :[Y]^#=/AY/ | 5 | 0.0007678 | 6 | 0.0006656 | 5 | 0.0006099 |
| [Y]=/IH/ | 21 | 0.0032248 | 30 | 0.0033278 | 25 | 0.0030495 |
| | ------- | --------- | ------- | --------- | ------- | --------- |
| | 108 | 0.0165848 | 153 | 0.0169717 | 139 | 0.0169554 |
| *** ZRULE *** | | | | | | |
| [Z]=/Z/ | 25 | 0.0038391 | 34 | 0.0037715 | 29 | 0.0035374 |
| | ------- | --------- | ------- | --------- | ------- | --------- |
| | 25 | 0.0038391 | 34 | 0.0037715 | 29 | 0.0035374 |
| | ======= | ========= | ======= | ========= | ======= | ========= |
| | 6512 | 1.0 | 9015 | 1.0 | 8198 | 1.0 |

Table 8
STAT Results for the First 8000 Words of the Brown Corpus and the IPA-to-Votrax Phase of the Translation by Version 3 of the Rules

| Rule | No. of Words Matched | | Total Frequencies of Words Matched | | Total No. of Texts for Words Matched | |
|---|---|---|---|---|---|---|
| | Abs. | Relative | Abs. | Relative | Abs. | Relative |
| *** IYRULE *** | | | | | | |
| [IY]=[E] | 1721 | 0.0363433 | 119333 | 0.0353984 | 58706 | 0.0400941 |
| | 1721 | 0.0363433 | 119333 | 0.0353984 | 58706 | 0.0400941 |
| *** IHRULE *** | | | | | | |
| [IH]=[I] | 3609 | 0.0762132 | 224302 | 0.0665360 | 99065 | 0.0676579 |
| | 3609 | 0.0762132 | 224302 | 0.0665360 | 99065 | 0.0676579 |
| *** EYRULE *** | | | | | | |
| L [EY] R=[UH3 AI I3] | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| L [EY]=[UH3 AI AY] | 98 | 0.0020695 | 4764 | 0.0014132 | 2947 | 0.0020127 |
| [EY] R=[A I3] | 2 | 0.0000422 | 81 | 0.0000240 | 45 | 0.0000307 |
| [EY]=[A AY] | 779 | 0.0164506 | 46357 | 0.0137511 | 24556 | 0.0167709 |
| | 879 | 0.0185623 | 51202 | 0.0151883 | 27548 | 0.0188143 |
| *** EHRULE *** | | | | | | |
| L [EH]=[UH3 EH] | 178 | 0.0037589 | 7349 | 0.0021800 | 4476 | 0.0030569 |
| [EH]=[EH] | 2290 | 0.0483592 | 126746 | 0.0375974 | 70661 | 0.0482590 |
| | 2468 | 0.0521181 | 134095 | 0.0397774 | 75137 | 0.0513159 |
| *** AERULE *** | | | | | | |
| L [AE] R=[UH3 AE EH3] | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| L [AE]=[UH3 AE] | 129 | 0.0027242 | 5700 | 0.0016908 | 2961 | 0.0020223 |
| [AE] R=[AE1 EH3] | 22 | 0.0004646 | 841 | 0.0002495 | 544 | 0.0003715 |
| [AE]=[AE] | 1554 | 0.0328167 | 137131 | 0.0406780 | 43298 | 0.0295710 |
| | 1705 | 0.0360054 | 143672 | 0.0426182 | 46803 | 0.0319648 |
| *** AARULE *** | | | | | | |
| [AA]=[AH] | 1268 | 0.0267770 | 86985 | 0.0258029 | 35471 | 0.0242254 |
| | 1268 | 0.0267770 | 86985 | 0.0258029 | 35471 | 0.0242254 |

Table 8 (continued)
STAT Results for the First 8000 Words of the Brown Corpus and the IPA-to-Votrax Phase of the Translation by Version 3 of the Rules

| Rule | No. of Words Matched | | Total Frequencies of Words Matched | | Total No. of Texts for Words Matched | |
|---|---|---|---|---|---|---|
| | Abs. | Relative | Abs. | Relative | Abs. | Relative |
| *** AO RULE *** | | | | | | |
| L [AO] R=[UH3 O] | 21 | 0.0004435 | 633 | 0.0001878 | 317 | 0.0002165 |
| L [AO] ER=[UH3 AW O2] | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| L [AO]=[UH3 AW] | 24 | 0.0005068 | 2222 | 0.0006591 | 1235 | 0.0008435 |
| [AO] R=[O] | 422 | 0.0089116 | 36342 | 0.0107803 | 13460 | 0.0091927 |
| [AO] ER=[AW O2] | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [AO]=[AW] | 227 | 0.0047937 | 19317 | 0.0057301 | 9789 | 0.0066855 |
| | 694 | 0.0146556 | 58514 | 0.0173573 | 24801 | 0.0169382 |
| *** OWRULE *** | | | | | | |
| L [OW]=[UH3 O1 U1] | 78 | 0.0016472 | 4100 | 0.0012162 | 2766 | 0.0018891 |
| [OW]=[O1 U1] | 424 | 0.0089538 | 35818 | 0.0106249 | 17358 | 0.0118549 |
| | 502 | 0.0106010 | 39918 | 0.0118411 | 20124 | 0.0137440 |
| *** UH RULE *** | | | | | | |
| L [UH]=[UH3 OO] | 8 | 0.0001689 | 1282 | 0.0003803 | 729 | 0.0004979 |
| [UH]=[OO] | 113 | 0.0023863 | 12042 | 0.0035721 | 5245 | 0.0035822 |
| | 121 | 0.0025552 | 13324 | 0.0039524 | 5974 | 0.0040800 |
| *** UW RULE *** | | | | | | |
| [UW]=[IU U] | 576 | 0.0121637 | 68818 | 0.0204139 | 20244 | 0.0138259 |
| | 576 | 0.0121637 | 68818 | 0.0204139 | 20244 | 0.0138259 |
| *** ER RULE *** | | | | | | |
| IY [ER]=[I3 ER] | 14 | 0.0002956 | 551 | 0.0001634 | 342 | 0.0002336 |
| ER [ER]=[IU R] | 5 | 0.0001056 | 115 | 0.0000341 | 56 | 0.0000382 |
| L [ER]=[UH3 ER] | 45 | 0.0009503 | 1730 | 0.0005132 | 1141 | 0.0007793 |
| [ER] L=[UH3 ER] | 21 | 0.0004435 | 2081 | 0.0006173 | 1065 | 0.0007274 |
| R [ER]=[UH3 R] | 9 | 0.0001901 | 210 | 0.0000623 | 152 | 0.0001038 |
| [ER]=[ER] | 1090 | 0.0230181 | 66563 | 0.0197450 | 35008 | 0.0239092 |
| | 1184 | 0.0250032 | 71250 | 0.0211353 | 37764 | 0.0257915 |
| *** AXRULE *** | | | | | | |
| [AX]=[UH2] | 1195 | 0.0252355 | 183722 | 0.0544985 | 32803 | 0.0224033 |
| | 1195 | 0.0252355 | 183722 | 0.0544985 | 32803 | 0.0224033 |

Table 8 (continued)
STAT Results for the First 8000 Words of the Brown Corpus and the IPA-to-Votrax Phase of the Translation by Version 3 of the Rules

| Rule | No. of Words Matched | | Total Frequencies of Words Matched | | Total No. of Texts for Words Matched | |
|---|---|---|---|---|---|---|
| | Abs. | Relative | Abs. | Relative | Abs. | Relative |
| *** AHRULE *** | | | | | | |
| [AH]=[UH] | 725 | 0.0153102 | 51178 | 0.0151812 | 24431 | 0.0166855 |
| | 725 | 0.0153102 | 51178 | 0.0151812 | 24431 | 0.0166855 |
| *** AYRULE *** | | | | | | |
| [AY] L=[AH AY] | 32 | 0.0006758 | 2564 | 0.0007606 | 1345 | 0.0009186 |
| [AY] R=[AH I3] | 60 | 0.0012671 | 2188 | 0.0006490 | 1417 | 0.0009678 |
| [AY] ER=[AH AY] | 1 | 0.0000211 | 160 | 0.0000475 | 88 | 0.0000601 |
| [AY]=[AH EI] | 396 | 0.0083625 | 39495 | 0.0117156 | 16799 | 0.0114731 |
| | 489 | 0.0103265 | 44407 | 0.0131727 | 19649 | 0.0134196 |
| *** AWRULE *** | | | | | | |
| [AW]=[AH OI] | 151 | 0.0031887 | 17619 | 0.0052264 | 7661 | 0.0052322 |
| | 151 | 0.0031887 | 17619 | 0.0052264 | 7661 | 0.0052322 |
| *** OYRULE *** | | | | | | |
| L [OY] ER=[UH3 OI AY] | 2 | 0.0000422 | 32 | 0.0000095 | 23 | 0.0000157 |
| L [OY] L=[UH3 OI AY] | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| L [OY] R=[UH3 OI EH2] | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [OY] ER=[OI AY] | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [OY] L=[OI AY] | 9 | 0.0001901 | 244 | 0.0000724 | 123 | 0.0000840 |
| [OY] R=[OI EH2] | 0 | 0.0000000 | 0 | 0.0000000 | 0 | 0.0000000 |
| [OY]=[OI EI] | 59 | 0.0012459 | 2764 | 0.0008199 | 1769 | 0.0012082 |
| | 70 | 0.0014782 | 3040 | 0.0009018 | 1915 | 0.0013079 |
| *** YRULE *** | | | | | | |
| [Y]=[YI] | 275 | 0.0058073 | 20191 | 0.0059894 | 9198 | 0.0062819 |
| | 275 | 0.0058073 | 20191 | 0.0059894 | 9198 | 0.0062819 |
| *** PRULE *** | | | | | | |
| [P]=[P] | 1575 | 0.0332601 | 72857 | 0.0216120 | 44829 | 0.0306166 |
| | 1575 | 0.0332601 | 72857 | 0.0216120 | 44829 | 0.0306166 |
| *** BRULE *** | | | | | | |
| [B]=[B] | 826 | 0.0174431 | 59010 | 0.0175045 | 25132 | 0.0171643 |
| | 826 | 0.0174431 | 59010 | 0.0175045 | 25132 | 0.0171643 |

Table 8 (continued)
STAT Results for the First 8000 Words of the Brown Corpus and the IPA-to-Votrax Phase of the Translation by Version 3 of the Rules

| Rule | No. of Words Matched | | Total Frequencies of Words Matched | | Total No. of Texts for Words Matched | |
|---|---|---|---|---|---|---|
| | Abs. | Relative | Abs. | Relative | Abs. | Relative |
| *** TRULE *** | | | | | | |
| [T]=[T] | 3468 | 0.0732356 | 242828 | 0.0720315 | 108215 | 0.0739070 |
| | 3468 | 0.0732356 | 242828 | 0.0720315 | 108215 | 0.0739070 |
| *** DRULE *** | | | | | | |
| [D]=[D] | 2179 | 0.0460151 | 150389 | 0.0446108 | 70118 | 0.0478881 |
| | 2179 | 0.0460151 | 150389 | 0.0446108 | 70118 | 0,0478881 |
| *** KRULE *** | | | | | | |
| [K]=[K] | 2174 | 0.0459095 | 101571 | 0.0301296 | 59818 | 0.0408536 |
| | 2174 | 0.0459095 | 101571 | 0.0301296 | 59818 | 0.0408536 |
| *** GRULE *** | | | | | | |
| [G]=[G] | 459 | 0.0096929 | 24315 | 0.0072127 | 13679 | 0.0093423 |
| | 459 | 0.0096929 | 24315 | 0.0072127 | 13679 | 0.0093423 |
| *** FRULE *** | | | | | | |
| [F]=[F] | 853 | 0.0180133 | 64428 | 0.0191116 | 30329 | 0.0207136 |
| | 853 | 0.0180133 | 64428 | 0.0191116 | 30329 | 0.0207136 |
| *** VRULE *** | | | | | | |
| [V]=[V] | 690 | 0.0145711 | 75264 | 0.0223260 | 22479 | 0.0153524 |
| | 690 | 0.0145711 | 75264 | 0.0223260 | 22479 | 0.0153524 |
| *** THRULE *** | | | | | | |
| [TH]=[TH] | 194 | 0.0040968 | 21426 | 0.0063557 | 8341 | 0.0056966 |
| | 194 | 0.0040968 | 21426 | 0.0063557 | 8341 | 0.0056966 |
| *** DHRULE *** | | | | | | |
| [DH]=[THV] | 66 | 0.0013938 | 109582 | 0.0325059 | 9026 | 0.0061644 |
| | 66 | 0.0013938 | 109582 | 0.0325059 | 9026 | 0.0061644 |

Table 8 (continued)
STAT Results for the First 8000 Words of the Brown Corpus and the IPA-to-Votrax Phase of the Translation by Version 3 of the Rules

| Rule | No. of Words Matched | | Total Frequencies of Words Matched | | Total No. of Texts for Words Matched | |
|---|---|---|---|---|---|---|
| | Abs. | Relative | Abs. | Relative | Abs. | Relative |
| *** SRULE *** | | | | | | |
| [S]=[S] | 2927 | 0.0618110 | 156066 | 0.0462948 | 87361 | 0.0596645 |
| | 2927 | 0.0618110 | 156066 | 0.0462948 | 87361 | 0.0596645 |
| *** ZRULE *** | | | | | | |
| [Z]=[Z] | 1446 | 0.0305360 | 100188 | 0.0297193 | 39797 | 0.0271800 |
| | 1446 | 0.0305360 | 100188 | 0.0297193 | 39797 | 0.0271800 |
| *** SHRULE *** | | | | | | |
| [SH]=[SH] | 646 | 0.0136419 | 29706 | 0.0088119 | 16224 | 0.0110804 |
| | 646 | 0.0136419 | 29706 | 0.0088119 | 16224 | 0.0110804 |
| *** ZHRULE *** | | | | | | |
| [ZH]=[ZH] | 39 | 0.0008236 | 1864 | 0.0005529 | 1222 | 0.0008346 |
| | 39 | 0.0008236 | 1864 | 0.0005529 | 1222 | 0.0008346 |
| *** HHRULE *** | | | | | | |
| [HH]=[H] | 319 | 0.0067365 | 55364 | 0.0164229 | 14106 | 0.0096339 |
| | 319 | 0.0067365 | 55364 | 0.0164229 | 14106 | 0.0096339 |
| *** CHRULE *** | | | | | | |
| [CH]=[T CH] | 297 | 0.0062719 | 20318 | 0.0060270 | 9481 | 0.0064752 |
| | 297 | 0.0062719 | 20318 | 0.0060270 | 9481 | 0.0064752 |
| *** JHRULE *** | | | | | | |
| [JH]=[D J] | 406 | 0.0085737 | 17746 | 0.0052641 | 10038 | 0.0068556 |
| | 406 | 0.0085737 | 17746 | 0.0052641 | 10038 | 0.0068556 |
| *** MRULE *** | | | | | | |
| [M]=[M] | 1520 | 0.0320987 | 99240 | 0.0294381 | 47574 | 0.0324914 |
| | 1520 | 0.0320987 | 99240 | 0.0294381 | 47574 | 0.0324914 |

Table 8 (continued)
STAT Results for the First 8000 Words of the Brown Corpus and the IPA-to-Votrax Phase of the Translation by Version 3 of the Rules

| Rule | No. of Words Matched | | Total Frequencies of Words Matched | | Total No. of Texts for Words Matched | |
|---|---|---|---|---|---|---|
| | Abs. | Relative | Abs. | Relative | Abs. | Relative |
| *** NRULE *** | | | | | | |
| [N]=[N] | 3403 | 0.0718630 | 250800 | 0.0743962 | 103930 | 0.0709805 |
| | 3403 | 0.0718630 | 250800 | 0.0743962 | 103930 | 0.0709805 |
| *** NXRULE *** | | | | | | |
| [NX]=[NG] | 617 | 0.0130295 | 28255 | 0.0083814 | 18773 | 0.0128213 |
| | 617 | 0.0130295 | 28255 | 0.0083814 | 18773 | 0.0128213 |
| *** LRULE *** | | | | | | |
| IY [L]=[I3 L] | 87 | 0.0018372 | 4272 | 0.0012672 | 2757 | 0.0018829 |
| EY [L]=[I3 L] | 50 | 0.0010559 | 1864 | 0.0005529 | 1118 | 0.0007636 |
| AY [L]=[I3 L] | 32 | 0.0006758 | 2564 | 0.0007606 | 1345 | 0.0009186 |
| OY [L]=[I3 L] | 9 | 0.0001901 | 244 | 0.0000724 | 123 | 0.0000840 |
| AE [L]=[UH3 L] | 65 | 0.0013726 | 2009 | 0.0005959 | 1129 | 0.0007711 |
| AO [L]=[UH3 L] | 115 | 0.0024285 | 10474 | 0.0031070 | 4607 | 0.0031464 |
| OW [L]=[UH3 L] | 55 | 0.0011615 | 3448 | 0.0010228 | 1977 | 0.0013502 |
| [L]=[L] | 2033 | 0.0429320 | 103741 | 0.0307733 | 60518 | 0.0413317 |
| | 2446 | 0.0516535 | 128616 | 0.0381521 | 73574 | 0.0502485 |
| *** WRULE *** | | | | | | |
| [W]=[W] | 393 | 0.0082992 | 55863 | 0.0165710 | 17447 | 0.0119157 |
| | 393 | 0.0082992 | 55863 | 0.0165710 | 17447 | 0.0119157 |
| *** WHRULE *** | | | | | | |
| [WH]=[H W] | 30 | 0.0006335 | 11341 | 0.0033641 | 3268 | 0.0022319 |
| | 30 | 0.0006335 | 11341 | 0.0033641 | 3268 | 0.0022319 |

Table 8 (continued)
STAT Results for the First 8000 Words of the Brown Corpus and the IPA-to-Votrax Phase of the Translation by Version 3 of the Rules

| Rule | No. of Words Matched | | Total Frequencies of Words Matched | | Total No. of Texts for Words Matched | |
|---|---|---|---|---|---|---|
| | Abs. | Relative | Abs. | Relative | Abs. | Relative |
| *** RRULE *** | | | | | | |
| [R] L=[UH3 R] | 26 | 0.0005491 | 1183 | 0.0003509 | 828 | 0.0005655 |
| [R]=[R] | 2723 | 0.0575031 | 161348 | 0.0478616 | 81321 | 0.0555394 |
| | 2749 | 0.0580521 | 162531 | 0.0482125 | 82149 | 0.0561049 |
| | 47354 | 1.0 | 3371138 | 1.0 | 1464204 | 1.0 |

## DISCUSSION AND CONCLUSIONS

Our results demonstrate that a simple algorithm driven by a small set of letter-to-sound rules (fewer than 350) can produce correct IPA transcriptions of the great majority of English words without using a large pronouncing dictionary; with the same algorithm, driven by a smaller set of rules, the IPA transcription can be translated into a form acceptable to a commercial speech synthesizer. Of the thousand most frequent words in English, the process mispronounces fewer than 4% if words are counted according to their frequency of occurrence. The error rate rises with decreasing word frequency. However, since over 2/3 of the words in a typical sample are among the most frequent thousand, the program's relatively poor performance on rare words does not drive the overall error rate higher than about 10%. Thus on the average the program mispronounces fewer than two words per sentence of ordinary written English. Most of the mispronounciations are single-phoneme errors and are easily correctable from context.

It has proved to be quite easy to modify the rules and experiment with different versions. As a result we have been able in passing from version 1 to version 3 to reduce the error rate for the 1000 most frequent words from an initial 32% to the present 4% while increasing the number of rules by three quarters.

We were at first slightly disappointed and more than slightly puzzled by the discrepancy between our performance score of 68% for version 1 of the rules (frequency-weighted score from Table 5) and Ainsworth's reported scores of 89% to 92% for his set of rules [10]. Since our version 1 is derived from and quite similar to Ainsworth's set, we had expected similar performance figures.

Three possible explanations suggest themselves. First, the difference between British and American pronunciation is more than a simple matter of dropping or retaining r's and replacement of one sound by another. Ainsworth's rules, being adapted to British English, might therefore in various subtle ways be unamenable to Americanization by such straightforward changes as we made while setting up version 1. Second, the question of what pronunciations of a word are acceptable is by no means cut and dried, even when one has a pronunciation dictionary at hand. Thus, although we had definite criteria in mind while scoring translations, we were not able to avoid subjectivity entirely. It is a dubious business at best to compare judgments of correctness arrived at independently under different circumstances by different judges having different expectations and different temperaments. Finally, the samples translated were different. The performance of a set of rules is sensitive to the vocabulary level of the material it is applied to; Table 3 illustrates this clearly. The Brown Corpus includes selections in a comprehensive range of styles, and Ainsworth's descriptions, namely, "textbook on phonetics," "modern novel," and "newspaper article on a political theme" [10], do not pin down where in that range the sources of his samples fall; they may be written plainly, or their authors may have salted their language with rare words. The actual reason for the discrepancy in scores is probably some combination of these three explanations.

Further additions and refinements to the rules could reduce the error rate still further. At this point however it appears that any improvement in inflection (pitch, stress, and timing) would be more beneficial than reducing the error rate by a few more percent. The present system produces a flat monotone that is fatiguing to follow for long. We are testing some simple heuristics for inflection. The listening-preference tests that we have done so far have led us to three conclusions: Not only stress and pitch but rhythm and timing are important to producing acceptable inflection. Not just any stress pattern will do; for instance, neither random stress nor strictly alternating stress were significantly preferred to the present monotone. One scheme for adding inflection was significantly preferred to all others tested except those involving hand-inserted "correct" English stress.

This preferred scheme bases stress assignment on two notions beyond the obvious one of falling inflection before a period and rising inflection before a question mark. The first is that a number of letter-to-sound rules, even in their present form, are good predictors for stressing and destressing. This is especially true of the rules with a schwa (/ə/) on the right-hand side, those for common function words, and those for common endings like ES and ED. The second is the tendency in English speech to stress approximately alternating syllables. Stress is assigned by a simple algorithm that plays these two notions off against one another. Unstressed syllables are given lower pitch and shorter duration than they would have if stressed. The timing of the stressed syllables is adjusted by further reduction of the duration of adjacent unstressed syllables and by lengthening any adjacent stressed syllables.

Our next step will be to test the inflection schemes using comprehension measures instead of listening preference. Present results indicate that simple intonation rules can make the output of a text-to-speech program easier to listen to. This is important in delaying fatigue and boredom in listening to machine speech for long periods. The comprehension tests are necessary, since naturalness and intelligibility do not always go together, and one is sometimes attained at the expense of the other. If the inflection scheme described results in increased intelligibility as well as increased naturalness, this will be an important advance.

Another task already in progress involves tailoring a version of the rules for a special application. We are putting together a data base of words pertinent to subjects of interest to the Navy. We intend to test the rules by translating these and, if it is necessary, retune the rules. It remains to be seen whether the statistics of the ordinary words in Navy-oriented English will require much reworking of the rules; however acronyms must certainly be dealt with. The pronounciations people give to unpronounceable combinations like WWMCCS (/wImIks/) are too arbitrary for any systematic procedure to have much hope of duplicating them. A more reasonable goal is to pronounce pronounceable combinations plausibly and spell out unpronounceable ones. One simple expedient is to pronounce each consonant as its name when the context is an isolated cluster consisting entirely of consonants. This already catches a good number of important acronyms and abbreviations (e.g., NRL), and the idea could be pushed further.

## ACNKOWLEDGMENTS

The authors are grateful to David Stryker for writing the software for the TI 960A that controls the speech-synthesis laboratory system. We thank David Woods for advice in Americanizing Ainsworth's rules and Ida Stockman for help in setting up the IPA-to-Votrax correspondence. The program that tabulates scores for translated words was contributed by Kay Lee Babington; we are indebted to her for help with the scoring.

## REFERENCES

1. H. Dudley, R.R. Riesz, and S.S.A. Watkins, "A Synthetic Speaker," J. Franklin Institute, 227, 739-764 (June 1939).

2. G.S. Kang, "Application of Linear Prediction Encoding to a Narrowband Voice Digitizer," NRL Report 7774, Oct. 1974.

3. H. Kučera and W.N. Francis, *Computational Analysis of Present-Day American English*, Brown University Press, Providence, 1967.

4. F.F. Lee, "Machine to Man Communication by Speech, Part I: Generation of Segmental Phonemes from Text," Spring Joint Computer Conference, 1968, pp. 333-338.

5. J. Allen, "Machine to Man Communication by Speech, Part II: Synthesis of Prosodic Features of Speech by Rule," Spring Joint Computer Conference, 1968, pp. 339-344.

6. F.F. Lee, "Reading Machine: From Text to Speech," IEEE Trans. Audio and Electroacoustics AU-17 (No. 4), 275-282 (Dec. 1969).

7. J. Allen, "Speech Synthesis from Unrestricted Text," IEEE International Convention Digest, 1971, pp. 108-109.

8. J. Allen, "Reading Machines for the Blind: The Technical Problems and the Methods Adopted for Their Solution," IEEE Trans. Audio and Electroacoustics AU-21 (No. 3), 259-264 (June 1973).

9. J. Allen, private communication, 1973.

10. W.A. Ainsworth, "A System for Converting English Text into Speech," IEEE Trans. Audio and Electroacoustics AU-21 (No. 3), 288-290 (June 1973).

11. W.A. Ainsworth, private communication, Apr. 1974.

12. M.D. McIlroy, "Synthetic English Speech by Rule," Bell Telephone Laboratories, Mar. 1974.

13. J.S. Kenyon and T.A. Knott, *A Pronouncing Dictionary of American English*, G&C Merriam Company, Springfield, Mass., 1951.

## Appendix A
## PROGRAM DOCUMENTATION FOR TRANS

The translation program was designed to make experimentation with the letter-to-sound rules simple. This resulted in a program that, once written and debugged, required a minimum of changing when the rules were altered.

The program starts by asking the user for input and output names: either the name of a text file or 'TTY', meaning the terminal; for output 'CAS', meaning the cassette unit, is also allowed. It then asks about various output options, including whether a stat file is wanted and what translation is wanted. Table A1 indicates the possible translations. English is arbitrary English text, IPA is text in the representation of the International Phonetic Alphabet given in Table 1, Votrax text consists of the mnemonic names of Votrax synthesizer codes, and ASCII is a representation of Votrax codes by pairs of ASCII characters that is sometimes used in transmitting to a Votrax over serial ASCII communication channels.

Table A1
Legal Translations

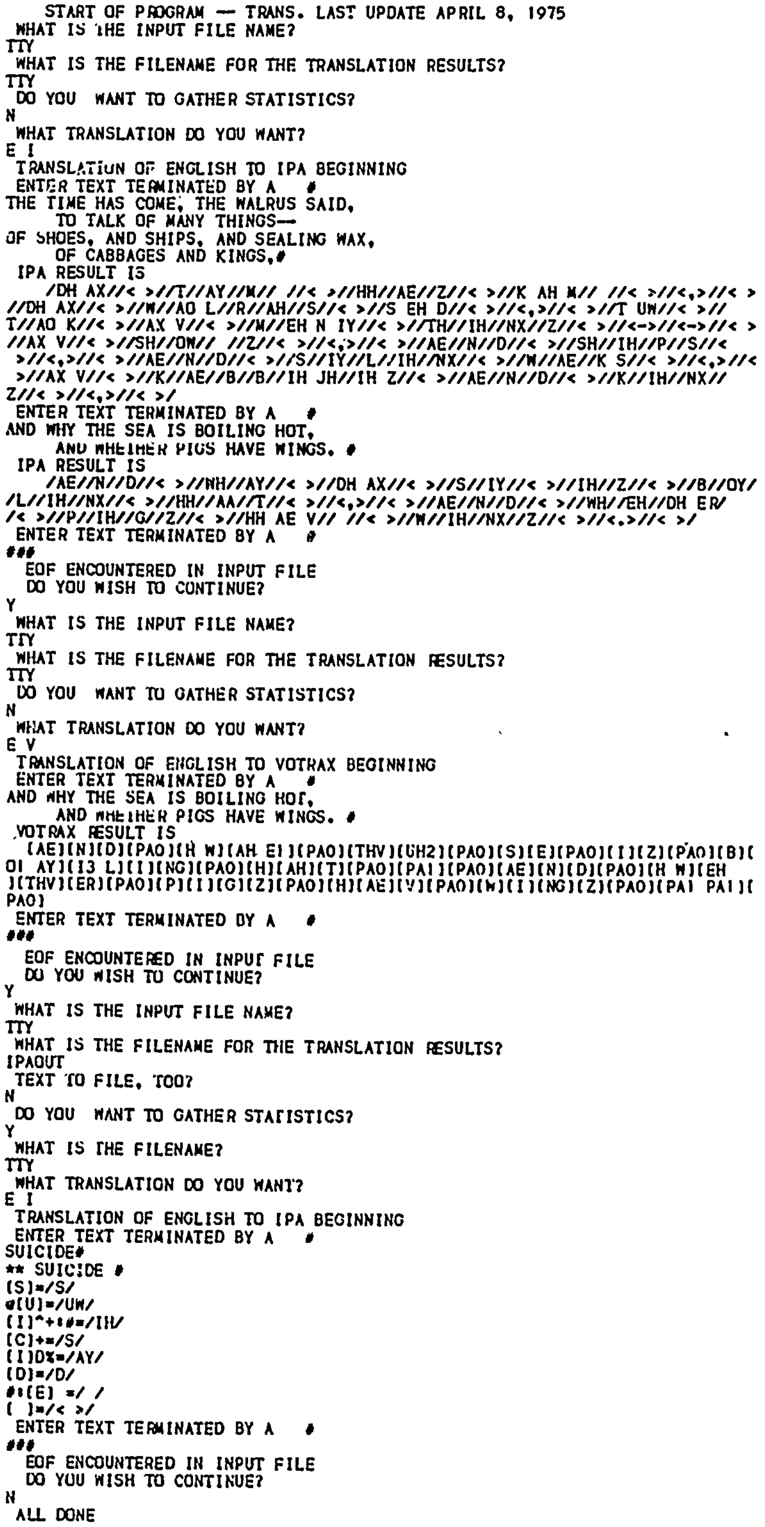
| Input String | Possible Output Strings |
|---|---|
| English | English, IPA, Votrax, or ASCII |
| IPA | IPA, Votrax, or ASCII |
| Votrax | Votrax or ASCII |
| ASCII | ASCII |

After questioning the user, the program expects a string of symbols terminated by an end-of-text marker '#'. If it receives one, it translates the string, produces the requested outputs, and looks for another such string. It keeps translating strings until it comes to the end of the input file or encounters '###' in an input string; at that point the used may choose whether to quit or to start over, respecifying file names. The translation consists of any of the following three passes that may be needed, applied in order: English to IPA, IPA to Votrax, and Votrax to ASCII. Figure A1 shows a sample dialog.

The program consists of four major sections:

- the rules,
- the function-name declarations and initialization,
- the translation routines, and
- the service routines.

These will now be described in detail.



```
    START OF PROGRAM -- TRANS. LAST UPDATE APRIL 8, 1975
 WHAT IS THE INPUT FILE NAME?
TTY
 WHAT IS THE FILENAME FOR THE TRANSLATION RESULTS?
TTY
 DO YOU  WANT TO GATHER STATISTICS?
N
 WHAT TRANSLATION DO YOU WANT?
E I
 TRANSLATION OF ENGLISH TO IPA BEGINNING
 ENTER TEXT TERMINATED BY A   #
THE TIME HAS COME, THE WALRUS SAID,
     TO TALK OF MANY THINGS--
OF SHOES, AND SHIPS, AND SEALING WAX,
     OF CABBAGES AND KINGS,#
 IPA RESULT IS
    /DH AX//< >//T//AY//M// //< >//HH//AE//Z//< >//K AH M// //< >//<,>//< >
//DH AX//< >//W//AO L//R//AH//S//< >//S EH D//< >//<,>//< >//T UW//< >//
T//AO K//< >//AX V//< >//M//EH N IY//< >//TH//IH//NX//Z//< >//<->//<->//< >
//AX V//< >//SH//OW// //Z//< >//<,>//< >//AE//N//D//< >//SH//IH//P//S//<
 >//<,>//< >//AE//N//D//< >//S//IY//L//IH//NX//< >//W//AE//K S//< >//<,>//<
 >//AX V//< >//K//AE//B//B//IH JH//IH Z//< >//AE//N//D//< >//K//IH//NX//
Z//< >//<,>//< >/
 ENTER TEXT TERMINATED BY A   #
AND WHY THE SEA IS BOILING HOT,
     AND WHETHER PIGS HAVE WINGS. #
 IPA RESULT IS
    /AE//N//D//< >//WH//AY//< >//DH AX//< >//S//IY//< >//IH//Z//< >//B//OY/
/L//IH//NX//< >//HH//AA//T//< >//<,>//< >//AE//N//D//< >//WH//EH//DH ER/
/< >//P//IH//G//Z//< >//HH AE V// //< >//W//IH//NX//Z//< >//<.>//< >/
 ENTER TEXT TERMINATED BY A   #
###
  EOF ENCOUNTERED IN INPUT FILE
  DO YOU WISH TO CONTINUE?
Y
 WHAT IS THE INPUT FILE NAME?
TTY
 WHAT IS THE FILENAME FOR THE TRANSLATION RESULTS?
TTY
 DO YOU  WANT TO GATHER STATISTICS?
N
 WHAT TRANSLATION DO YOU WANT?
E V
 TRANSLATION OF ENGLISH TO VOTRAX BEGINNING
 ENTER TEXT TERMINATED BY A   #
AND WHY THE SEA IS BOILING HOT,
     AND WHETHER PIGS HAVE WINGS. #
 VOTRAX RESULT IS
  [AE][N][D][PAO][H W][AH EI][PAO][THV][UH2][PAO][S][E][PAO][I][Z][PAO][B][
OI AY][I3 L][I][NG][PAO][H][AH][T][PAO][PA1][PAO][AE][N][D][PAO][H W][EH
][THV][ER][PAO][P][I][G][Z][PAO][H][AE][V][PAO][W][I][NG][Z][PAO][PA1 PA1][
PAO]
 ENTER TEXT TERMINATED BY A   #
###
  EOF ENCOUNTERED IN INPUT FILE
  DO YOU WISH TO CONTINUE?
Y
 WHAT IS THE INPUT FILE NAME?
TTY
 WHAT IS THE FILENAME FOR THE TRANSLATION RESULTS?
IPAOUT
 TEXT TO FILE, TOO?
N
 DO YOU  WANT TO GATHER STATISTICS?
Y
 WHAT IS THE FILENAME?
TTY
 WHAT TRANSLATION DO YOU WANT?
E I
 TRANSLATION OF ENGLISH TO IPA BEGINNING
 ENTER TEXT TERMINATED BY A   #
SUICIDE#
** SUICIDE #
[S]=/S/
@[U]=/UW/
[I]^+:#=/IH/
[C]+=/S/
[I]D%=/AY/
[D]=/D/
#:[E] =/ /
[ ]=/< >/
 ENTER TEXT TERMINATED BY A   #
###
  EOF ENCOUNTERED IN INPUT FILE
  DO YOU WISH TO CONTINUE?
N
 ALL DONE
```

Fig. A1 — Sample dialog with TRANS

### Rules

The rules section contains three groups of translation rules: English to IPA, IPA to Votrax, and Votrax to ASCII. The English-to-IPA and IPA-to-Votrax rules have the form

A[B]C=D,

where A, B, C, and D are character strings and B is nonempty. The interpretation is that in the left and right contexts specified by A and C, the string B is to be translated to D. The English-to-IPA part of the rules section initializes variables ARULE.ENG, BRULE.ENG, . . . , ZRULE.ENG, assigning to each a string of the form

rule\rule\. . .\rule\.

The string assigned to ARULE.ENG contains, in their proper order, all the A rules: the rules where A is the first character in brackets. The other letters of the alphabet are handled similarly, and there are in addition variables NUMBERRULE.ENG and PUNCTRULE.ENG that contain the rules for translating digits and punctuation marks. The IPA-to-Votrax part initializes variables IYRULE.IPA, . . . , RRULE.IPA, and PUNCTRULE.IPA in the same way.

Changing the translation rules requires no program changes except revising the rule text as it appears explicitly in the rules section. Since the English-to-IPA and IPA-to-Votrax rules are needed by programs other than TRANS, these parts of the rules section are kept in a separate file and combined with the rest of TRANS (or another program) for compilation.

The Votrax-to-ASCII rules vary from the format described. There is a one-to-one correspondence between the Votrax codes and their ASCII representations; consequently the rules are not context sensitive as are the other rules. Therefore each Votrax code names a rule consisting only of the ASCII pair corresponding to the code. For instance the variable AE1.CODE becomes 'OJ', since OJ is the ASCII representation for the Votrax code AE1.

### Function-name Declarations and Initialization

After the rules section, the program listing has function declarations and initialization of some often-used patterns and other variables. The function declarations define function names and formal parameters. The code for each function is in the program body, with the function name labeling the first statement.

### Translation Routines

The translation routines for English to IPA and IPA to Votrax are TRANSLATETEXT and VOTRAXTRANSLATE, both of which call on the routine TRANSLATE to do most of the work.

TRANSLATETEXT starts with a pointer I at position 1 of the input string and places the character pointed to in a variable CHAR. If the character is not the end-of-text marker, TRANSLATE is invoked with three parameters: the input text, CHAR, and an indication 'ENG' of which set of rules to use. TRANSLATE determines the rule that matches and returns the translation result as a value to be concatenated with any previous results. TRANSLATE also sets a variable INCVALUE to the length of the bracketed substring in the rule. TRANSLATETEXT uses this information to set the pointer to the next character that must be translated. The character is placed in CHAR, and the process repeats. On encountering the end-of-text marker, the routine returns to the calling program.

VOTRAXTRANSLATE works much like TRANSLATETEXT. Minor differences in the details of the scan are due to differences in the format of the input: the IPA symbols are represented by one- and two-character combinations delimited by blanks or slashes, and the symbols of English text are represented as single characters without delimiters. TRANSLATE is given 'IPA', rather than 'ENG', as the indication of which rules to use.

TRANSLATE takes three arguments: the text being translated (BUF), the character or symbol currently being scanned (GRAPHEME), and an indicator of the set of rules to use (QUAL). The routine replaces GRAPHEME by 'PUNCT' or 'NUMBER' if it is a punctuation symbol or a digit, then builds from GRAPHEME and QUAL the name of a string of rules to search for a match. For instance, if the last two arguments were 'A' and 'ENG', it would use ARULE.ENG, which contains the A rules for English-to-IPA translation. The routine then sequentially breaks off rules (substrings delimited by '\') from the rule string until it either runs out of rules or finds a rule whose left-hand side matches the text at the current position. In the first case it gives an error message; in the second case it returns the right-hand side of the rule as a function value and puts the length of the bracketed part of the left-hand side in INCVALUE to indicate how many spaces the pointer should be moved before resuming the scan.

As each rule is broken from the rule string, it is in turn broken into four pieces called BACKCHAR, CHARDEF, FORCHAR, and PHONEME. These pieces correspond to A, B, C, and D in the notation where A[B]C=D is the form of a rule. From the first three is built a SNOBOL pattern that tests whether the left-hand side of the rule matches the text at the appropriate position. Both BACKCHAR and FORCHAR are examined for special symbols. If none are found, BACKCHAR and FORCHAR, as they stand, are used in building the pattern. If any special symbols do occur, the code starting at SPECIALCASEPROC is executed. This code builds the necessary pattern by applying the function SPECIALBREAK to BACKCHAR and to FORCHAR.

SPECIALBREAK breaks its argument into (a) strings free of special symbols and (b) special symbols. These are concatenated back together in the same order with each special symbol replaced by its corresponding pattern. The routine works on the input string from left to right: it breaks off (a) everything up to the first special symbol and (b) the first special symbol; then it concatenates the initial string and the pattern corresponding to the symbol onto the end of the partial result (originally null); this continues until no special symbols remain, at which point what is left of the input string is added to the result and the function returns.

The patterns corresponding to the special symbols '*', '#', . . . are in variables whose names are 'PATTERN*', 'PATTERN#', . . . and must be referred to by writing $'PATTERN*', $'PATTERN#', . . . , since the names are not legal identifiers. For the introduction of a new special symbol, say '?', only two steps are necessary:

1. Add '?' to the string of special symbols in the variable SPECIALCASE.
2. Write the desired SNOBOL pattern and assign it to the appropriate variable with an assignment statement of the form

$'PATTERN?' = SNOBOL pattern.

A third step is desirable:

3. Update the comments to reflect the addition.

The same changes should probably be made in DICT at the same time.

ASCII translates a Votrax code string into the ASCII representation.

### Service Routines

The remainder of the SNOBOL program contains service routines to decide the type of translation, to set up file name definitions, to input the text from a specified file, to output translation results to a specified file, to gather statistics on the rules used in a translation, and to make the correct sequence of function calls to perform the translation requested.

Initially the program invokes a routine called FILEDEFINE. This routine asks for the input and output file names and sets up the correct correspondence for the computer system. It also sets flags to indicate whether the input text should also be written to the output file and whether statistics should be recorded to a named file. This is all at the user's option. Finally FILEDEFINE makes the logical file-name correspondences of INPUTTEXT for the input file, STATISTICS for the statistics file, and TRANSTEXT for the translation results file.

After file definitions, the user must indicate the type of translation wanted. The routine CLI reads this information, which may be in abbreviated form, and expands the input type and output type to their full spellings, placing these results in the variables IN and OUT. Then the program branches to the statement labeled by the contents of IN concatenated with OUT. The code at each of these points invokes TRANSLATETEXT, VOTRAXTRANSLATE, or ASCII with the appropriate parameters to produce the translation requested.

FILEOUT outputs Votrax code to a file in a format compatible for a TI 733 cassette.

## TRANS Program Listing

```
****************************************************************
*
*
*               *****  TRANS  *****
*
*
*  THIS IS THE TRANSLATION PROGRAM WHICH INPUTS
*  ENGLISH TEXT AND TRANSLATES TO PHONEMES.
*  IT IS WRITTEN IN SNOBOL FOR THE PDP10.
*  IT WILL REDEFINE FILES ON ENCOUNTERING AN EOF
*  OR ON SEEING A ### STARTING IN POSITION 1 OF THE INPUT STRING.
*  IT ALSO PROVIDES THE FACILITY TO TRANSFER INTERMEDIATE AND FINAL
*  OUTPUT RESULTS TO A PREDEFINED FILE OR TO THE TTY.
*  OUTPUT TO A FILE IS IN A FORM COMPATIBLE WITH THE SPEECH LAB.
*  IF THE CASSETTE IS SPECIFIED IT WILL OUTPUT IN A FORM COMPATIBLE
*  TO THE SPEECH LAB.
*
****************************************************************
*
*
****************************************************************
*
*
*       *****  ENGLISH TO IPA TRANSLATION RULES  *****
*
*
*  IN THESE RULES SOME SPECIAL SYMBOLS SERVE AS KEYWORDS.
*  THIS SPECIAL CONNOTATION HOLDS UNLESS THE SYMBOL
*  APPEARS BETWEEN BRACKETS; THEN IT DENOTES ITSELF.
*
*       # = 1 OR MORE VOWELS
*       * = 1 OR MORE CONSONANTS
*       . = A VOICED CONSONANT
*       $ = SINGLE CONSONANT FOLLOWED BY AN 'I' OR 'E'
*       % = SUFFIX SUCH AS 'E','ES','ED','ER','ING','ELY'
*       & = A SIBILANT
*       @ = A CONSONANT AFTER WHICH LONG 'U' IS PRONOUNCED
*               AS IN 'RULE', NOT 'MULE'
*       ^ = A SINGLE CONSONANT
*       + = A FRONT VOWEL: 'E','I','Y'
*       : = 0 OR MORE CONSONANTS
*
*
****************************************************************
*
*
```

```
          PUNCTRULE.ENG =
+                   "[ ]'=/ /\"
+                   '[ - ]=/ /\'
+                   '[ ]=/< >/\'
+                   '[-]=/<->/\'
+                   ", [' S]=/Z/\"
+                   "#:.E [' S]=/Z/\"
+                   "# [' S]=/Z/\"
+                   "[' ]=/ /\"
+                   '[,]=/<,>/\'
+                   '[.]=/<.>/\'
+                   '[?]=/<?>/\'
*
          ARULE.ENG =
+                   '[A] =/AX/\'
+                   ' [ARE] =/AA R/\'
+                   ' [AR]O=/AX R/\'
+                   '[AR]#=/EH R/\'
+                   ' ^[AS]#=/EY S/\'
+                   '[A]WA=/AX/\'
+                   '[AW]=/AO/\'
+                   ' :[ANY]=/EH N IY/\'
+                   '[A]^+#=/EY/\'
+                   '#:[ALLY]=/AX L IY/\'
+                   ' [AL]#=/AX L/\'
+                   '[AGAIN]=/AX G EH N/\'
+                   '#:[AG]E=/IH JH/\'
+                   '[A]^+:#=/AE/\'
+                   ' :[A]^+ =/EY/\'
+                   '[A]^%=/EY/\'
+                   ' [ARR]=/AX R/\'
+                   '[ARR]=/AE R/\'
+                   ' :[AR] =/AA R/\'
+                   '[AR] =/ER/\'
+                   '[AR]=/AA R/\'
+                   '[AIR]=/EH R/\'
+                   '[AI]=/EY/\'
+                   '[AY]=/EY/\'
+                   '[AU]=/AO/\'
+                   '#:[AL] =/AX L/\'
+                   '#:[ALS] =/AX L Z/\'
+                   '[ALK]=/AO K/\'
+                   '[AL]^=/AO L/\'
+                   ' :[ABLE]=/EY B AX L/\'
+                   '[ABLE]=/AX B AX L/\'
+                   '[ANG]+=/EY N JH/\'
+                   '[A]=/AE/\'
*
```

```
          BRULE.ENG =
+                   ' [BE]^#=/B IH/\'
+                   '[BEING]=/B IY IH NX/\'
+                   ' [BOTH] =/B OW TH/\'
+                   ' [BUS]#=/B IH Z/\'
+                   '[BUIL]=/B IH L/\'
+                   '[B]=/B/\'
*
          CRULE.ENG =
+                   ' [CH]^=/K/\'
+                   '^E[CH]=/K/\'
+                   '[CH]=/CH/\'
+                   ' S[CI]#=/S AY/\'
+                   '[CI]A=/SH/\'
+                   '[CI]O=/SH/\'
+                   '[CI]EN=/SH/\'
+                   '[C]+=/S/\'
+                   '[CK]=/K/\'
+                   '[COM]%=/K AH M/\'
+                   '[C]=/K/\'
*
          DRULE.ENG =
+                   '#:[DED] =/D IH D/\'
+                   '.E[D] =/D/\'
+                   '#^:E[D] =/T/\'
+                   ' [DE]^#=/D IH/\'
+                   ' [DO] =/D UW/\'
+                   ' [DOES]=/D AH Z/\'
+                   ' [DOING]=/D UW IH NX/\'
+                   ' [DOW]=/D AW/\'
+                   '[DU]A=/JH UW/\'
+                   '[D]=/D/\'
*
```

```
          ERULE.ENG =
+                    '#:[E] =/ /\'
+                    "' ^:[E] =/ /\"
+                    ' :[E] =/IY/\'
+                    '#[ED] =/D/\'
+                    '#:[E]D =/ /\'
+                    '[EV]ER=/EH V/\'
+                    '[E]^%=/IY/\'
+                    '[ERI]#=/IY R IY/\'
+                    '[ERI]=/EH R IH/\'
+                    '#:[ER]#=/ER/\'
+                    '[ER]#=/EH R/\'
+                    '[ER]=/ER/\'
+                    ' [EVEN]=/IY V EH N/\'
+                    '#:[E]W=/ /\'
+                    '@[EW]=/UW/\'
+                    '[EW]=/Y UW/\'
+                    '[E]O=/IY/\'
+                    '#:&[ES] =/IH Z/\'
+                    '#:[E]S =/ /\'
+                    '#:[ELY] =/L IY/\'
+                    '#:[EMENT]=/M EH N T/\'
+                    '[EFUL]=/F UH L/\'
+                    '[EE]=/IY/\'
+                    '[EARN]=/ER N/\'
+                    ' [EAR]^=/ER/\'
+                    '[EAD]=/EH D/\'
+                    '#:[EA] =/IY AX/\'
+                    '[EA]SU=/EH/\'
+                    '[EA]=/IY/\'
+                    '[EIGH]=/EY/\'
+                    '[EI]=/IY/\'
+                    ' [EYE]=/AY/\'
+                    '[EY]=/IY/\'
+                    '[EU]=/Y UW/\'
+                    '[E]=/EH/\'
*

          FRULE.ENG =
+                    '[FUL]=/F UH L/\'
+                    '[F]=/F/\'
*
```

```
             GRULE.ENG =
+                    '[GIV]=/G IH V/\'
+                    ' [G]I^=/G/\'
+                    '[GE]T=/G EH/\'
+                    'SU[GGES]=/G JH EH S/\'
+                    '[GG]=/G/\'
+                    ' B#[G]=/G/\'
+                    '[G]+=/JH/\'
+                    '[GREAT]=/G R EY T/\'
+                    '#[GH]=/ /\'
+                    '[G]=/G/\'
*

             HRULE.ENG =
+                    ' [HAV]=/HH AE V/\'
+                    ' [HERE]=/HH IY R/\'
+                    ' [HOUR]=/AW ER/\'
+                    '[HOW]=/HH AW/\'
+                    '[H]#=/HH/\'
+                    '[H]=/ /\'
',

             IRULE.ENG =
+                    ' [IN]=/IH N/\'
+                    ' [I] =/AY/\'
+                    '[IN]D=/AY N/\'
+                    '[IER]=/IY ER/\'
+                    '#:R[IED] =/IY D/\'
+                    '[IED] =/AY D/\'
+                    '[IEN]=/IY EH N/\'
+                    '[IE]T=/AY EH/\'
+                    ' :[I]%=/AY/\'
+                    '[I]%=/IY/\'
+                    '[IE]=/IY/\'
+                    '[I]^+:#=/IH/\'
+                    '[IR]#=/AY R/\'
+                    '[IZ]%=/AY Z/\'
+                    '[IS]%=/AY Z/\'
+                    '[I]D%=/AY/\'
+                    '+^[I]^+=/IH/\'
+                    '[I]T%=/AY/\'
+                    '#^:[I]^+=/IH/\'
+                    '[I]^+=/AY/\'
+                    '[IR]=/ER/\'
+                    '[IGH]=/AY/\'
+                    '[ILD]=/AY L D/\'
+                    '[IGN] =/AY N/\'
+                    '[IGN]^=/AY N/\'
+                    '[IGN]%=/AY N/\'
+                    '[IQUE]=/IY K/\'
+                    '[I]=/IH/\'
*
```

```
        JRULE.ENG =
+               '[J]=/JH/\'
*
        KRULE.ENG =
+               ' [K]N=/ /\'
+               '[K]=/K/\'
*
        LRULE.ENG =
+               '[LO]C#=/L OW/\'
+               'L[L]=/ /\'
+               '#^:[L]%=/AX L/\'
+               '[LEAD]=/L IY D/\'
+               '[L]=/L/\'
*
        MRULE.ENG =
+               '[MOV]=/M UW V/\'
+               '[M]=/M/\'
*
        NRULE.ENG =
+               'E[NG]+=/N JH/\'
+               '[NG]R=/NX G/\'
+               '[NG]#=/NX G/\'
+               '[NGL]%=/NX G AX L/\'
+               '[NG]=/NX/\'
+               '[NK]=/NX K/\'
+               ' [NOW] =/N AW/\'
+               '[N]=/N/\'
*
```

```
              ORULE.ENG =
+                         '[OF] =/AX V/\'
+                         '[OROUGH]=/ER OW/\'
+                         '#:[OR] =/ER/\'
+                         '#:[ORS] =/ER Z/\'
+                         '[OR]=/AO R/\'
+                         ' [ONE]=/W AH N/\'
+                         '[OW]=/OW/\'
+                         ' [OVER]=/OW V ER/\'
+                         '[OV]=/AH V/\'
+                         '[O]^%=/OW/\'
+                         '[O]^EN=/OW/\'
+                         '[O]^I#=/OW/\'
+                         '[OL]D=/OW L/\'
+                         '[OUGHT]=/AO T/\'
+                         '[OUGH]=/AH F/\'
+                         ' [OU]=/AW/\'
+                         'H[OU]S#=/AW/\'
+                         '[OUS]=/AX S/\'
+                         '[OUR]=/AO R/\'
+                         '[OULD]=/UH D/\'
+                         '^[OU]^L=/AH/\'
+                         '[OUP]=/UW P/\'
+                         '[OU]=/AW/\'
+                         '[OY]=/OY/\'
+                         '[OING]=/OW IH NX/\'
+                         '[OI]=/OY/\'
+                         '[OOR]=/AO R/\'
+                         '[OOK]=/UH K/\'
+                         '[OOD]=/UH D/\'
+                         '[OO]=/UW/\'
+                         '[O]E=/OW/\'
+                         '[O] =/OW/\'
+                         '[OA]=/OW/\'
+                         ' [ONLY]=/OW N L IY/\'
+                         ' [ONCE]=/W AH N S/\'
+                         *[ON ' T]=/OW N T/\*
+                         'C[O]N=/AA/\'
+                         '[O]NG=/AO/\'
+                         ' ^:[O]N=/AH/\'
+                         'I[ON]=/AX N/\'
+                         '#:[ON] =/AX N/\'
+                         '#^[ON]=/AX N/\'
+                         '[O]ST =/OW/\'
+                         '[OF]^=/AO F/\'
+                         '[OTHER]=/AH DH ER/\'
+                         '[OSS] =/AO S/\'
+                         '#^:[OM]=/AH M/\'
+                         '[O]=/AA/\'
*
```

```
          PRULE.ENG =
+                 '[PH]=/F/\'
+                 '[PEOP]=/P IY P/\'
+                 '[POW]=/P AW/\'
+                 '[PUT] =/P UH T/\'
+                 '[P]=/P/\'
*
          QRULE.ENG =
+                 '[QUAR]=/K W AO R/\'
+                 '[QU]=/K W/\'
+                 '[Q]=/K/\'
*
          RRULE.ENG =
+                 ' [RE]^#=/R IY/\'
+                 '[R]=/R/\'
*
          SRULE.ENG =
+                 '[SH]=/SH/\'
+                 '#[SION]=/ZH AX N/\'
+                 '[SOME]=/S AH M/\'
+                 '#[SUR]#=/ZH ER/\'
+                 '[SUR]#=/SH ER/\'
+                 '#[SU]#=/ZH UW/\'
+                 '#[SSU]#=/SH UW/\'
+                 '#[SED] =/Z D/\'
+                 '#[S]#=/Z/\'
+                 '[SAID]=/S EH D/\'
+                 '^[SION]=/SH AX N/\'
+                 '[S]S=/ /\'
+                 '.[S] =/Z/\'
+                 '#:.E[S] =/Z/\'
+                 '#^:##[S] =/Z/\'
+                 '#^:#[S] =/S/\'
+                 'U[S] =/S/\'
+                 ' :#[S] =/Z/\'
+                 ' [SCH]=/S K/\'
+                 '[S]C+=/ /\'
+                 '#[SM]=/Z M/\'
+                 "#[SN] '=/Z AX N/\"
+                 '[S]=/S/\'
*
```

```
            TRULE.ENG =
+                   ' [THE] =/DH AX/\'
+                   '[TO] =/T UW/\'
+                   '[THAT] =/DH AE T/\'
+                   ' [THIS] =/DH IH S/\'
+                   ' [THEY]=/DH EY/\'
+                   ' [THERE]=/DH EH R/\'
+                   '[THER]=/DH ER/\'
+                   '[THEIR]=/DH EH R/\'
+                   ' [THAN] =/DH AE N/\'
+                   ' [THEM] =/DH EH M/\'
+                   '[THESE] =/DH IY Z/\'
+                   ' [THEN]=/DH EH N/\'
+                   '[THROUGH]=/TH R UW/\'
+                   '[THOSE]=/DH OW Z/\'
+                   '[THOUGH] =/DH OW/\'
+                   ' [THUS]=/DH AH S/\'
+                   '[TH]=/TH/\'
+                   '#:[TED] =/T IH D/\'
+                   'S[TI]#N=/CH/\'
+                   '[TI]O=/SH/\'
+                   '[TI]A=/SH/\'
+                   '[TIEN]=/SH AX N/\'
+                   '[TUR]#=/CH ER/\'
+                   '[TU]A=/CH UW/\'
+                   ' [TWO]=/T UW/\'
+                   '[T]=/T/\'
*
            URULE.ENG =
+                   ' [UN]I=/Y UW N/\'
+                   ' [UN]=/AH N/\'
+                   ' [UPON]=/AX P AO N/\'
+                   '@[UR]#=/UH R/\'
+                   '[UR]#=/Y UH R/\'
+                   '[UR]=/ER/\'
+                   '[U]^ =/AH/\'
+                   '[U]^^=/AH/\'
+                   '[UY]=/AY/\'
+                   ' G[U]#=/ /\'
+                   'G[U]%=/ /\'
+                   'G[U]#=/W/\'
+                   '#N[U]=/Y UW/\'
+                   '@[U]=/UW/\'
+                   '[U]=/Y UW/\'
*
            VRULE.ENG =
+                   '[VIEW]=/V Y UW/\'
+                   '[V]=/V/\'
*
```

```
          WRULE.ENG =
+                   ' [WERE]=/W ER/\'
+                   '[WA]S=/W AA/\'
+                   '[WA]T=/W AA/\'
+                   '[WHERE]=/WH EH R/\'
+                   '[WHAT]=/WH AA T/\'
+                   '[WHOL]=/HH OW L/\'
+                   '[WHO]=/HH UW/\'
+                   '[WH]=/WH/\'
+                   '[WAR]=/W AO R/\'
+                   '[WOR]^=/W ER/\'
+                   '[WR]=/ R/\'
+                   '[W]=/W/\'
*
          XRULE.ENG =
+                   '[X]=/K S/\'
*
          YRULE.ENG =
+                   '[YOUNG]=/Y AH NX/\'
+                   ' [YOU]=/Y UW/\'
+                   ' [YES]=/Y EH S/\'
+                   ' [Y]=/Y/\'
+                   '#^:[Y] =/IY/\'
+                   '#^:[Y]I=/IY/\'
+                   ' :[Y] =/AY/\'
+                   ' :[Y]#=/AY/\'
+                   ' :[Y]^+:#=/IH/\'
+                   ' :[Y]^#=/AY/\'
+                   '[Y]=/IH/\'
*
          ZRULE.ENG =
+                   '[Z]=/Z/\'
*
          NUMBERRULE.ENG =
+                   '[0]=/Z IH R OW/\'
+                   '[1]=/W AH N/\'
+                   '[2]=/T UW/\'
+                   '[3]=/TH R IY/\'
+                   '[4]=/F OW R/\'
+                   '[5]=/F AY V/\'
+                   '[6]=/S IH K S/\'
+                   '[7]=/S EH V AX N/\'
+                   '[8]=/EY T/\'
+                   '[9]=/N AY N/\'
*
*
```

```
************************************************************
*
*
*       *****  IPA TO VOTRAX TRANSLATIONS RULES  *****
*
*
************************************************************
*
*
          IYRULE.IPA = '[IY]=[E]\'
          IHRULE.IPA = '[IH]=[I]\'
          EYRULE.IPA = 'L [EY] R=[UH3 A1 I3]\'
+                   'L [EY]=[UH3 A1 AY]\'
+                   '[EY] R=[A I3]\'
+                   '[EY]=[A AY]\'
          EHRULE.IPA = 'L [EH]=[UH3 EH]\'
+                   '[EH]=[EH]\'
          AERULE.IPA = 'L [AE] R=[UH3 AE EH3]\'
+                   'L [AE]=[UH3 AE]\'
+                   '[AE] R=[AE1 EH3]\'
+                   '[AE]=[AE]\'
          AARULE.IPA = '[AA]=[AH]\'
          AORULE.IPA = 'L [AO] R=[UH3 O]\'
+                   'L [AO] ER=[UH3 AW O2]\'
+                   'L [AO]=[UH3 AW]\'
+                   '[AO] R=[O]\'
+                   '[AO] ER=[AW O2]\'
+                   '[AO]=[AW]\'
          OWRULE.IPA = 'L [OW]=[UH3 O1 U1]\'
+                   '[OW]=[O1 U1]\'
          UHRULE.IPA = 'L [UH]=[UH3 OO]\'
+                   '[UH]=[OO]\'
          UWRULE.IPA = '[UW]=[IU U]\'
          ERRULE.IPA = 'IY [ER]=[I3 ER]\'
+                   'ER [ER]=[IU R]\'
+                   'L [ER]=[UH3 ER]\'
+                   '[ER] L=[UH3 ER]\'
+                   'R [ER]=[UH3 R]\'
+                   '[ER]=[ER]\'
          AXRULE.IPA = '[AX]=[UH2]\'
          AHRULE.IPA = '[AH]=[UH]\'
          AYRULE.IPA = '[AY] L=[AH AY]\'
+                   '[AY] R=[AH I3]\'
+                   '[AY] ER=[AH AY]\'
+                   '[AY]=[AH E1]\'
          AWRULE.IPA = '[AW]=[AH O1]\'
          OYRULE.IPA = 'L [OY] ER=[UH3 O1 AY]\'
+                 'L [OY] L=[UH3 O1 AY]\'
+                 'L [OY] R=[UH3 O1 EH2]\'
+                 '[OY] ER=[O1 AY]\'
+                 '[OY] L=[O1 AY]\'
+                 '[OY] R=[O1 EH2]\'
+                 '[OY]=[O1 E1 ]\'
          YRULE.IPA = '[Y]=[Y1 ]\'
          PRULE.IPA = '[P]=[P]\'
          BRULE.IPA = '[B]=[B]\'
          TRULE.IPA = '[T]=[T]\'
          DRULE.IPA = '[D]=[D]\'
          KRULE.IPA = '[K]=[K]\'
          GRULE.IPA = '[G]=[G]\'
          FRULE.IPA = '[F]=[F]\'
          VRULE.IPA = '[V]=[V]\'
          THRULE.IPA = '[TH]=[TH]\'
          DHRULE.IPA = '[DH]=[THV]\'
          SRULE.IPA = '[S]=[S]\'
          ZRULE.IPA = '[Z]=[Z]\'
          SHRULE.IPA = '[SH]=[SH]\'
          ZHRULE.IPA = '[ZH]=[ZH]\'
          HHRULE.IPA = '[HH]=[H]\'
          CHRULE.IPA = '[CH]=[T CH]\'
          JHRULE.IPA = '[JH]=[D J]\'
          MRULE.IPA = '[M]=[M]\'
          NRULE.IPA = '[N]=[N]\'
          NXRULE.IPA = '[NX]=[NG]\'
          LRULE.IPA = 'IY [L]=[I3 L]\'
+                 'EY [L]=[I3 L]\'
+                 'AY [L]=[I3 L]\'
+                 'OY [L]=[I3 L]\'
+                 'AE [L]=[UH3 L]\'
+                 'AO [L]=[UH3 L]\'
+                 'OW [L]=[UH3 L]\'
+                 '[L]=[L]\'
          WRULE.IPA = '[W]=[W]\'
          WHRULE.IPA = '[WH]=[H W]\'
          RRULE.IPA = '[R] L=[UH3 R]\'
+                 '[R]=[R]\'
          PUNCTRULE.IPA = '[< >]=[PA0]\'
+                 '[<,>]=[PA1 ]\'
+                 '[<.>]=[PA1 PA1 ]\'
+                 '[<?>]=[PA1 PA1 ]\'
+                 '[<->]=[PA1 ]\'
*
*
```

```
******************************************************************
*
*
*       *****  VOTRAX TO ASCII TRANSLATION RULES   *****
*
*
******************************************************************
*
*
        BLANK.CODE =
        PA0.CODE = 'CH'
        PA1.CODE = 'NK'
*
        A.CODE = '@J'
        A1.CODE = 'FH'
        A2.CODE = 'EH'
        AE.CODE = 'NJ'
        AE1.CODE = 'OJ'
        AH.CODE = 'DJ'
        AH1.CODE = 'EI'
        AH2.CODE = 'HH'
        AW.CODE = 'MK'
        AW1.CODE = 'CI'
        AW2.CODE = '@K'
        AY.CODE = 'AJ'
*
        B.CODE = 'NH'
*
        CH.CODE = '@I'
*
        D.CODE = 'NI'
        DT.CODE = 'DH'
*
        E.CODE = 'LJ'
        E1.CODE = 'LK'
        EH.CODE = 'KK'
        EH1.CODE = 'BH'
        EH2.CODE = 'AH'
        EH3.CODE = '@H'
        ER.CODE = 'JK'
*
        F.CODE = 'MI'
*
        G.CODE = 'LI'
*
        H.CODE = 'KI'
*
        I.CODE = 'GJ'
        I1.CODE = 'KH'
        I2.CODE = 'JH'
```

```
         I3.CODE = 'IH'
         IU.CODE = 'FK'
*
         J.CODE = 'JI'
*
         K.CODE = 'II'
*
         L.CODE = 'HI'
*
         M.CODE = 'LH'
*
         N.CODE = 'MH'
         NG.CODE = 'DI'
*
         O.CODE = 'FJ'
         O1.CODE = 'EK'
         O2.CODE = 'DK'
         OO.CODE = 'GI'
         OO1.CODE = 'FI'
*
         P.CODE = 'EJ'
*
         R.CODE = 'KJ'
*
         S.CODE = 'OI'
         SH.CODE = 'AI'
*
         T.CODE = 'JJ'
         TH.CODE = 'IK'
         THV.CODE  = 'HK'
*
         U.CODE = 'HJ'
         U1.CODE = 'GK'
         UH.CODE = 'CK'
         UH1.CODE = 'BK'
         UH2.CODE = 'AK'
         UH3.CODE = 'CJ'
*
         V.CODE = 'OH'
*
         W.CODE = 'MJ'
*
         Y.CODE = 'IJ'
         Y1.CODE = 'BJ'
*
         Z.CODE = 'BI'
         ZH.CODE = 'GH'
*
*
```

```
**************************************************************
*
*
*  DEFINE FUNCTIONS TO BE USED BY THE PROGRAM.
*
*
*  SPECIALBREAK:  BREAKS APART SEGMENTS OF RULES WHICH
*  CONTAIN SPECIAL CASE SYMBOLS
*  VOWEL OR CONSONANT CLASSES, ETC.
*
        DEFINE('SPECIALBREAK(STR)')
*
*  TRANSLATETEXT:  CALLS TRANSLATE TO TRANSLATE THE TEXT.
*  PARAMETER IS THE ENGLISH TEXT.
*
        DEFINE('TRANSLATETEXT(TEXT)')
*
*  TRANSLATE:  BREAKS OFF SEGMENTS OF A SET
*  OF TRANSLATION RULES AND DETERMINES
*  WHETHER THEY APPLY TO TEXT.
*
        DEFINE('TRANSLATE(BUF,GRAPHEME,QUAL)')
*
*  VOTRAXTRANSLATE:  TRANSLATES A STRING OF IPA SYMBOLS
*  TO VOTRAX PHONETICS ACCORDING
*  A SET OF PREDEFINED RULES.
*
        DEFINE('VOTRAXTRANSLATE(IPAPHONEMES)')
*
*  READTEXT:  INPUT THE COMPLETE TEXT TO BE TRANSLATED
*
        DEFINE('READTEXT()')
*
*  ASCII:  TRANSLATES THE VOTRAX MNEMONIC TO ASCII.
*
        DEFINE('ASCII(STRING)')
*
*  FILEDEFINE:  THIS IS THE MODULE WHICH ASKS THE USER
*  THE NAMES OF THE INPUT FILE AND RESULT FILE
*  AND THE STATISTICS FILE AND MAKES
*  VARIABLE ASSIGNMENTS.
*
        DEFINE('FILEDEFINE()')
*
*  FILEOUT:  THIS ROUTINE OUTPUTS THE MNEMONIC VOTRAX
*  CODE TO A FILE ASSOCIATED WITH TRANSTEXT.
*
        DEFINE('FILEOUT(BUF)')
*
```

```
*  CLI:  INPUTS THE TRANSLATION TO BE DONE
*  BUILDS THE VARIABLE BRANCH IN IN AND OUT
*
        DEFINE('CLI()')
*
*
*  MAIN PROGRAM CODE STARTS HERE
*
*
*  SET TRIM VARIABLE SO TRAILING BLANKS ARE AUTOMATICALLY DELETED.
*
        &TRIM = 1
*
*  INIT SOME VARIABLES.
*
        INPUT('INPUT',2,80)
        BLANK = ' '
        DOUBLEBLANK = '  '
        NULL =
        ENDTEXT = '#'
        ESCAPECODE = '###'
        QUOTE = '"'
        SINGLEQUOTE = "'"
        SPECIALCASE = '#*.$%&@^+:'
        ILLEGALPUNCT = '[]/\'
        PUNCTSYMBOL = ' ,.?;:+*"$%&-<>!()='  SINGLEQUOTE
        PUNCTSYMNOBLANK = ',.?;:+*"$%&=-<>!()' SINGLEQUOTE
        NUMBER = '1234567890'
        SET = 'ON'
        UNSET = 'OFF'
*
*  DEFINE THE DELETE CHARACTER BY USING THE MACHINES ALPHABET.
*  ALSO DEFINE RECORDON AND REORDOFF AND ENDOFMSG.
*
        &ALPHABET
+                TAB(18) LEN(1) . RECORDON
+                TAB(20) LEN(1) . RECORDOFF
+                TAB(94) LEN(1) . ENDOFMESSAGE
+                TAB(127) LEN(1) . DELETE
*
*  DEFINE SOME PATTERNS USED IN THE PROGRAM.
*  THIS WAY SAVE THE BUILDING TIME DURING PROGRAM EXECUTION.
*
        ENDTEXTTEST = ENDTEXT RPOS(0)
        RULEBREAKPATTERN = BREAK('\') . RULE  '\'
        RULECHARSEP =  BREAK('[') . BACKCHAR  '['
+                BREAK(']') . CHARDEF  ']'
+                BREAK('=') . FORCHAR  '='
+                REM . PHONEME
```

```
        VOWEL = 'AEIOUY'
        CONSONANT = 'BCDFGHJKLMNPQRSTVWXZ'
        VOICED = 'BDVGJLMNRWZ'
        FRUNT = 'EIY'
        SUFFIX = 'ER ' ! 'E ' ! 'ES ' ! 'ED ' ! 'ING '
        SIBILANT = ANY('SCGZXJ') ! 'CH' ! 'SH'
        NONPAL = ANY('TSRDLZNJ') ! 'TH' ! 'CH' ! 'SH'
        $'PATTERN#' = ANY(VOWEL) ARBNO(ANY(VOWEL))
        $'PATTERN*' = ANY(CONSONANT) ARBNO(ANY(CONSONANT))
        $'PATTERN.' = ANY(VOICED)
        $'PATTERN$' = ANY(CONSONANT) ANY('EI')
        $'PATTERN%' = SUFFIX
        $'PATTERN&' = SIBILANT
        $'PATTERN@' = NONPAL
        $'PATTERN^' = ANY(CONSONANT)
        $'PATTERN+' = ANY(FRONT)
        $'PATTERN:' = ARBNO(ANY(CONSONANT))
*
        TTY = POS(0) 'TTY' RPOS(0)
        CAS = POS(0) 'CAS' RPOS(0)
        NOANS = POS(0)  ('N' ! 'NO')  RPOS(0)
        YESANS = POS(0) ('Y' ! 'YE' ! 'YES')  RPOS(0)
        ENGLISH = 'ENGLISH' ! 'ENGLIS' ! 'ENGLI' !
+                'ENGL' ! 'ENG' ! 'EN' ! 'E'
        IPA = 'IPA' ! 'IP' ! 'I'
        VOTRA = 'VOTRAX' ! 'VOTRA' ! 'VOTR' ! 'VOT' ! 'VO' ! 'V'
        ASCI = 'ASCII' ! 'ASCI' ! 'ASC' ! 'AS' ! 'A'
*
*
        OUTPUT = '    START OF PROGRAM — TRANS.'
+                '  LAST UPDATE APRIL 8, 1975'
*
*
*  DECLARE MAX LENGTH OF STRINGS AND NUMBER OF STATEMENT
*  EXECUTIONS SO SNOBOL DOESN'T BOMB.
*
        &STLIMIT = 100000000
        &MAXLNGTH = 50000
*
```

```
*  DEFINE FILE INPUT AND OUTPUT VARIABLES.
*
BEG     FILEDEFINE()
*
*  CALL ROUTINE TO DETERMINE WHAT TRANSLATIONS ARE TO BE DONE.
*
RECOMMAND  CLI()
*
*  READ IN THE TEXT TO BE TRANSLATED.
*  AFTER INPUTTING TEXT BRANCH TO CODE BASED ON TRANSLATION.
*
        INDIRECT = IN OUT
        OUTPUT = ' TRANSLATION OF ' IN ' TO ' OUT ' BEGINNING'
*
*  INSERT A BLANK BEFORE INPUT STRING TO DELIMIT.FIRST WORD.
*
REREED  ALLTEXT =    READTEXT()                    :F(EOF)S($INDIRECT)
*
*
*  DEFINE THE TRANSLATIONS WHICH ARE REFLEXIVE AND DO NOT
*  ACTUALLY REQUIRE TRANSLATION--
*  THEY JUST OUTPUT THE INPUT TEXT.
*
ENGLISHENGLISH
**      OUTPUT = ' TRANS FROM ENG TO ENG '
                                                   :(MSG)
IPAIPA
**      OUTPUT = ' TRANS FROM IPA TO IPA '
                                                   :(MSG)
VOTRAXVOTRAX
**      OUTPUT = ' TRANS FROM VOTRAX TO VOTRAX '
                                                   :(MSG)
ASCIIASCII
**      OUTPUT = ' TRANS FROM ASCII TO ASCII '
                                                   :(MSG)
*
MSG     TTYFLAG  SET                               :S(TTYOUTO)
        FILEOUT(ALLTEXT ENDOFMESSAGE)              :(REREED)
TTYOUTO  TRANSTEXT = ' RESULT IS ' ALLTEXT         :(REREED)
*
*
```

```
*  DEFINE THE ROUTINES WHICH ARE NOT IMPLEMENTED.
*  THESE ARE ASCII TO VOTRAX
*  ASCII TO IPA
*  ASCII TO ENGLISH
*  VOTRAX TO IPA
*  VOTRAX TO ENGLISH
*  AND IPA TO ENGLISH.
*
*
IPAENGLISH  OUTPUT = ' TRANSLATION OF IPA TO ENG NOT IMPLEMENTED'
+                                                :(RECOMMAND)
ASCIIVOTRAX  OUTPUT = ' TRANS OF ASCII TO VOTRAX NOT IMPLEMENTED'
+                                                :(RECOMMAND)
ASCIIIPA  OUTPUT = ' TRANS OF ASCII TO IPA NOT IMPLEMENTED'
+                                                :(RECOMMAND)
ASCIIENGLISH  OUTPUT = ' TRANS OF ASCII TO ENG NOT IMPLEMENTED'
+                                                :(RECOMMAND)
VOTRAXIPA  OUTPUT = ' TRANS OF VOTRAX TO IPA NOT IMPLEMENTED '
+                                                :(RECOMMAND)
VOTRAXENGLISH  OUTPUT = ' TRANS OF VOTRAX TO ENG NOT IMPLEMENTED'
+                                                :(RECOMMAND)
*
*
*  TO TRANSLATE FROM IPA TO VOTRAX AND ASCII.
*  TO TRANSLATE FROM VOTRAX TO ASCII.
*
*
IPAASCII
**      OUTPUT = ' TRANS OF IPA TO ASCII'
*
*  REMOVE END OF TEXT MARKER.
*
        ALLTEXT  ENDTEXTTEST  = BLANK
*
*  CALL ROUTINE TO TRANS FROM IPA TO VOTRAX CODES.
*
        VOTRAXSYMBOLS = VOTRAXTRANSLATE(ALLTEXT)
*
*  CALL ROUTINE TO TRANS TO ASCII.
*
        ASCIIRESULT = ASCII(VOTRAXSYMBOLS)
*
*  SEE IF SHOULD OUTPUT IN FORMAT FOR SPEECH LAB.
*
        TTYFLAG  SET                             :S(TTYOUT)
        FILEOUT(ASCIIRESULT ENDOFMESSAGE)        :(REREED)
TTYOUT  TRANSTEXT = ' ASCII RESULT IS ' ASCIIRESULT    :(REREED)
*
*
```

```
IPAVOTRAX
**       OUTPUT = ' TRANS OF IPA TO VOTRAX '
*
*   REMOVE END TEXT MARKER.
*
         ALLTEXT ENDTEXTTEST       = BLANK
*
*   TRANSLATE THE STRING.
*
         VOTRAXSYMBOLS = VOTRAXTRANSLATE(ALLTEXT)
*
*   SEE IF SHOULD OUTPUT TO CASSETTE.
*
         TTYFLAG  SET                                  :F(FILE2)
         TRANSTEXT = ' THE VOTRAX RESULT IS '
         TRANSTEXT = '    ' VOTRAXSYMBOLS              :(REREED)
FILE2    FILEOUT(VOTRAXSYMBOLS ENDOFMESSAGE)           :(REREED)
*
*
ENGLISHVOTRAX
**       OUTPUT = ' TRANS OF ENG TO VOTRAX '
*
*   SET FLAG TO SAY TRANS TO VOTRAX ALSO.
*
         VOTRAXFLAG = SET                              :(ENGVOTRAX)
*
*   WILL BRANCH TO HERE AT CONCLUSION OF TRANS TO VOTRAX.
*   SEE IF SHUULD OUTPUT TO CAS.
*
ENDENGVOTRAX     TTYFLAG     SET                       :F(FILE3)
         TRANSTEXT = ' VOTRAX RESULT IS '
         TRANSTEXT = '   ' VOTRAXSYMBOLS               :(REREED)
FILE3    VOTRAXSYMBOLS = REPLACE(VOTRAXSYMBOLS,'[]',' ')
         FILEOUT(VOTRAXSYMBOLS ENDOFMESSAGE)           :(REREED)
*
*
```

```
ENGLISHASCII
**      OUTPUT = ' TRANS OF ENG TO ASCII '
*
*  SET FLAGS FOR ASCII TRANS AND FOR VOTRAX FLAGS.
*
        ASCIIFLAG = SET
        VOTRAXFLAG = SET                            :(ENGVOTRAX)
*
*  RETURN HERE AT COMPLETION OF TRANSLATION.
*  SEE IF SHOULD OUTPUT TO CAS.
*
ENDENGASCII  TTYFLAG SET                            :F(FILE4)
        TRANSTEXT = ' ASCII RESULT IS '
        TRANSTEXT = '   ' ASCIIRESULT               :(REREED)
FILE4   FILEOUT(ASCIIRESULT ENDOFMESSAGE)           :(REREED)
*
*
VOTRAXASCII
**      OUTPUT = ' TRANS OF VOTRAX TO ASCII'
*
*  REMOVE END TEXT MARKER.
*
        ALLTEXT ENDTEXTTEST     = NULL
*
*  CALL ROUTINE TO TRANSLATE.
*
        ASCIIRESULT = ASCII(ALLTEXT)                :F(REREED)
*
*  SEE IF SHOULD OUTPUT TO CAS.
*
        TTYFLAG  SET                                :F(FILE5)
        TRANSTEXT =  '    ' ASCIIRESULT             :(REREED)
FILE5    FILEOUT(ASCIIRESULT ENDOFMESSAGE)          :(REREED)
*
*
```

```
ENGLISHIPA
**      OUTPUT = ' TRANS OF ENG TO IPA '
*
*  BRANCH HERE IF TRANS OF ENG TO VOTRAX OR ASCII.
*
ENGVOTRAX  IPARESULT = NULL
        IPARESULT = TRANSLATETEXT(ALLTEXT)
*
*  SEE IF WE ARE TO TRANS TO VOTRAX.
*
VOTRAXCALL   VOTRAXFLAG SET = UNSET              :F(ENDENGIPA)
*
*  FLAG WAS SET SO TRANS TO VOTRAX.
*  TRANS THE STRING.
*
        VOTRAXSYMBOLS = VOTRAXTRANSLATE(IPARESULT)
*
*  IS ASCII FLAG SET?
*
        ASCIIFLAG SET = UNSET                    :F(ENDENGVOTRAX)
*
*  YES--CALL ASCII ROUTINE.
*
        ASCIIRESULT = ASCII(VOTRAXSYMBOLS)       :(ENDENGASCII)
*
*  COME HERE IF NOT TRANS TO VOTRAX OR ASCII..
*  SEE IF SHOULD OUTPUT TO CAS.
*
ENDENGIPATTYFLAG  SET                            :F(FILE6)
        TRANSTEXT = ' IPA RESULT IS '
        TRANSTEXT = '     ' IPARESULT            :(REREED)
FILE6   FILEOUT(IPARESULT ENDOFMESSAGE)          :(REREED)
*
*
```

```
****************************************************************
*
*       TRANSLATETEXT
*
****************************************************************
*
*
*  START THE SCAN AT FIRST CHARACTER OF INPUT.
*  POSITION 0 IS A BLANK INSERTED BY THE PROGRAM TO DELIMIT
*  THE FIRST WORD.
*
TRANSLATETEXT      I = 1
        TRANSLATETEXT = NULL
*
*  PICK OFF ONE CHARACTER OF 'TEXT' AT ITH POSITION.
*
NEXTCHAR        TEXT   POS(I)   LEN(1) . CHAR
*
*  TEST FOR END TEXT MARKER  -- IF SO RETURN.
*
        CHAR    ENDTEXT                         :S(RETURN)
*
*  CONCATENATE THE PHONEME WHICH IS RETURNED BY 'TRANSLATE'.
*
        TRANSLATETEXT = TRANSLATETEXT TRANSLATE(TEXT,CHAR,'ENG')
*
*  INCREMENT THE POINTER TO THE NEXT CHARACTER IN 'TEXT' TO BE
*  TRANSLATED.
*  'INCVALUE' SET BY ROUTINE 'TRANSLATE'.
*
        I = I + INCVALUE                        :(NEXTCHAR)
*
*
```

```
****************************************************************
*
*        TRANSLATE
*
*  THIS ROUTINE DOES THE ACTUAL TRANSLATION OF THE LETTER
*  PASSED BY THE MAIN PROGRAM IN 'CHAR'  BY CHOOSING THE RULE
*  WHICH APPLIES TO THE CONTENTS OF 'TEXT' AND PASSING BACK THE
*  PHONEME.
*  ADDITIONALLY, TRANSLATE SETS A VARIABLE 'INCVALUE' TO THE
*  NUMBER OF SYMBOLS REPLACED SO THAT THE
*  MAIN ROUTINE MAY INCREMENT THE POINTER INTO 'TEXT'.
*
****************************************************************
*
*
*  SET OF SPECIAL CASE SYMBOLS.
*        # = 1 OR MORE VOWELS
*        * = 1 OR MORE CONSONANTS
*        . = A VOICED CONSONANT
*        $ = SINGLE CONSONANT FOLLOWED BY AN 'I' OR 'E'
*        % = SUFFIX SUCH AS 'E','ES','ED','ER','ING','ELY'
*        & = A SIBILANT
*        @ = A CONSONANT AFTER WHICH LONG 'U' IS PRONOUNCED
*                AS IN 'RULE', NOT 'MULE'
*        ^ = A SINGLE CONSONANT
*        + = A FRONT VOWEL: 'E','I','Y'
*        : = 0 OR MORE CONSONANTS
*
*        SPECIALCASE = '#*.$%&@^+:'
*        PUNCTSYMBOL = ' ,.?;:+*"$%&-<>!(=' SINGLEQUOTE
*
*
TRANSLATE  GRAPHEME ANY(PUNCTSYMBOL)  REM  = 'PUNCT'
        GRAPHEME ANY(NUMBER) REM = 'NUMBER'
*
*  COPY THE SET OF POSSIBLE RULES FOR THE CHARACTER PASSED.
*
        GRRULE = $(GRAPHEME 'RULE.' QUAL)
*
*  BREAK OFF ONE OF THE RULES.
*       RULEBREAKPATTERN = BREAK('\') . RULE  '\'
*
NEXTRULE  GRRULE  RULEBREAKPATTERN = NULL       :F(NORULEAPPLIES)
*
```

```
*  BREAK THE RULE INTO ITS COMPONENT PIECES OF THE PATTERN
*  TO MATCH AND THE PHONEMES TO REPLACE IF MATCH IS A SUCCESS.
*  BREAK RULE INTO PIECES OF STRING BEFORE THE SYMBOL TRANSLATING,
*  STRING OF LETTERS TO BE REPLACED BY 'PHONEME' IF MATCH OCCURS,
*  AND STRING OF LETTERS AFTER THE REPLACEMENT LETTERS.
*
*       RULECHARSEP =  BREAK('[') . BACKCHAR  '['
*+                BREAK(']') . CHARDEF  ']'
*+                BREAK('=') . FORCHAR  '='
*+                REM . PHONEME
*
        RULE    RULECHARSEP
*
*  A CHECK OF THE RULE MUST BE MADE IN CASE A CHARACTER OTHER THAN
*  AN ALPHABETIC OR BRACKET APPEARS.
*  WHEN THIS OCCURS THE RULE HAS A SPECIAL CASE SUCH AS A VOWEL OR
*  CONSONANT SEQUENCE  (#,*) OR A VOICED CONSONANT (.).
*  WHEN ONE OF THESE SPECIAL CHARACTERS APPEARS IN THE RULE, THE
*  ROUTINE  'SPECIALCASEPROC'  BUILDS A PATTERN TO MATCH 'TEXT'.
*  OTHERWISE THE PIECES OF THE RULE ARE USED EXPLICITLY AS BELOW.
*
*
*  IF A SPECIAL CHAR IS FOUND IN THE RULE GO TO THE SPECIAL PROC.
*
        (FORCHAR BACKCHAR)  ANY(SPECIALCASE)     :S(SPECIALCASEPROC)
*
*  NO SPECIAL SYMBOLS APPEARED IN THE RULE.
*  MATCH ON THE PIECES.  IF FAIL GET NEXT RULE.
*  DETERMINE WHERE TO BEGIN MATCH BY BACKING UP IN BUF
*  THE NUMBER OF CHARS IN BACKCHAR.
*
        BACK = GE(I,SIZE(BACKCHAR)) SIZE(BACKCHAR)
+                                                  :F(NEXTRULE)
        BUF     POS(I - BACK) BACKCHAR CHARDEF FORCHAR
+                                                  :F(NEXTRULE)
*
```

```
*  MATCH WAS MADE.
*  RETURN THE PHONEME SEQUENCE AS SPECIFIED BY THE RULE.
*  DETERMINE THE AMOUNT TO INCREMENT THE POINTER.
*  THIS VALUE COMPUTED BASED ON NO. CHARS IN CHARDEF.
*
INCSET  INCVALUE = SIZE(CHARDEF)
*
**       OUTPUT = ' RULE USED WAS <' RULE '>'
**       OUTPUT = ' PHONEME IS <' PHONEME '>'
*  GATHER STATISTIC AT THIS POINT.
*  SEE IF STATFLAG IS SET.  IF SO OUTPUT RESULTS.
*
         STATFLAG  SET                            :F(TRANSRET)
         STATISTICS = RULE
*
TRANSRET  TRANSLATE = PHONEME                     :(RETURN)
*
*
*  SPECIALCASEPROC:
*
*  THIS IS THE SECTION WHICH TAKES CARE OF THE SPECIAL CASE RULES.
*  IT CREATES PATTERNS FOR THE SPECIAL CASES BY CALLING THE
*  FUNCTION  'SPECIALBREAK'   WHICH BUILDS A PATTERN BASED ON
*  THE SPECIAL CHARACTERS IN THE STRING PASSED AS THE PARAMETER..
*  ON FAILURE TO MATCH THE PATTERN ANOTHER RULE WILL BE TRIED.
*
*  RULES MUST NOT HAVE SPECIAL CASES INTERNAL TO THE
*  BRACKETS,  I.E. IN 'CHARDEF'.
*  IF THEY DO THEN THE PROGRAM MUST BE REVISED TO HANDLE
*  THE CASE BY USING 'SPECIALBREAK' ON 'CHARDEF' ALSO.
*  A RULE IS OF THE FORM :
*       A[B]C=/PHONEMES/
*  WHERE A AND C ARE STRINGS OF ALPHABETICS OR
*  SPECIAL SYMBOLS
*  AND B IS A STRING OF ALPHABETIC ONLY.
*
*  CREATE A PATTERN FOR SPECIALCASES BY CALLING 'SPECIALBREAK'
*  POSITION POINTER AND CHARACTERS TRYING TO MATCH ('CHARDEF'),
*  CALL SPECIALBREAK WITH FORCHAR.
*  ON FAILURE TO MATCH GET ANOTHER RULE.
*
SPECIALCASEPROC
+        BUF     SPECIALBREAK(BACKCHAR) POS(I) CHARDEF
+                SPECIALBREAK(FORCHAR)   :S(INCSET)F(NEXTRULE)
*
*  ON SUCCESS  RETURN PHONEMES TO MAIN PROGRAM.
*
*
```

```
*****************************************************************
*
*       SPECIALBREAK
*
*  THIS FUNCTION BUILDS A PATTERN MATCH BASED ON THE PIECES OF THE
*  RULE PASSED TO IT AS A PARAMETER.
*
*****************************************************************
*
*
*  DEFINE PIECES OF THE PATTERN BASED ON THE SPECIAL CHARACTER
*  ENCOUNTERED IN PARAMETER.
*
SPECIALBREAK  PATTERN =
*
*
*       VOWEL = 'AEIOUY'
*       CONSONANT = 'BCDFGHJKLMNPQRSTVWXZ'
*       VOICED = 'BDVGJLMNRWZ'
*       FRONT = 'EIY'
*       SUFFIX = 'ER ' ! 'E ' ! 'ES ' ! 'ED ' ! 'ING ' ! 'ELY '
*       SIBILANT = ANY('SCGZXJ') ! 'CH' ! 'SH'
*       NONPAL = ANY('TSRDLZNJ') ! 'TH' ! 'CH' ! 'SH'
*       $'PATTERN#' = ANY(VOWEL) ARBNO(ANY(VOWEL))
*       $'PATTERN*' = ANY(CONSONANT) ARBNO(ANY(CONSONANT))
*       $'PATTERN.' = ANY(VOICED)
*       $'PATTERN$' = ANY(CONSONANT) ANY('EI')
*       $'PATTERN%' = SUFFIX
*       $'PATTERN&' = SIBILANT
        $'PATTERN@' = NONPAL
*       $'PATTERN^' = ANY(CONSONANT)
*       $'PATTERN+' = ANY(FRONT)
*       $'PATTERN:' = ARBNO(ANY(CONSONANT))
*
*  REPLACE EVERYTHING UP TO SPECIAL CHARACTER BY NULL AND ASSIGN
*  WHAT MATCHED TO 'PATTERN1'.
*
REMATCH  STR   BREAK(SPECIALCASE) . PATTERN1 =        :F(ALLDONE)
*
*  BREAK OFF THE SPECIAL CASE CHAR INTO SYM.
*  REPLACE IT BY THE NULL STRING.
*
        STR   LEN(1) . SYM =
*
*  BUILD PATTERN TO PASS BACK TO CALLER BASED ON PREVIOUSLY
*  BUILT PARTIAL PATTERN AND PATTERN BASED ON THE SPECIAL SYMBOL
*  STORED IN 'SYM' LOOP TO REMATCH UNTIL NOTHING LEFT IN STR OR
*  NO MORE SPECIAL CHARACTERS.
*
        PATTERN = PATTERN  PATTERN1  $('PATTERN' SYM)   :(REMATCH)
*
*  RETURN WITH PATTERN THAT WAS BUILT.
*  THE REMAINDER OF 'STR' HAS NO SPECIAL CHARACTERS IN IT.
*
ALLDONE  SPECIALBREAK = PATTERN   STR          :(RETURN)
*
*
```

```
***************************************************************
*
*       VOTRAXTRANSLATE
*
*  TRANSLATES FROM IPA NOTATION TO VOTRAX SYMBOLS.
*  PARAMETERS ARE THE STRING TO BE TRANSLATED. EACH PHONEME
*  MAY BE DELIMITED BY SLASHES.
*
***************************************************************
*
*
VOTRAXTRANSLATE   VOTRAXSTR = NULL
        I = 1
        ENDIPASTR  = '#'
        IPASTR = IPAPHONEMES ENDIPASTR
        IPASTR = REPLACE(IPASTR,'/',BLANK)
*
*  REMOVE DOUBLE BLANKS.
*
REMOVEBLANKS  IPASTR DOUBLEBLANK = BLANK        :S(REMOVEBLANKS)
TRY     IPASTR POS(I)  ENDIPASTR                :S(DONEVOTRAX)
        IPASTR POS(I) BREAK(BLANK) . IPASYM
DIFFERENT  VOTRAXSTR = VOTRAXSTR  TRANSLATE(IPASTR,IPASYM,'IPA')
        I = I + INCVALUE + 1                    :(TRY)
DONEVOTRAX   VOTRAXTRANSLATE = VOTRAXSTR        :(RETURN)
*
*
```

```
******************************************************************
*
*        READTEXT
*
*  READ A SERIES OF TEXT TO BE TRANSLATED.
*  TERMINATE IT BY A # .
*
******************************************************************
*
*
READTEXT   TOTALTEXT =
*
*
*        ILLEGALPUNCT = '[]/\'
*        PUNCTSYMNOBLANK = './,;+<>?*=-:)(&%$"!' SINGLEQUOTE
*        QUOTE = '"'
*
*
*  SKIP MESSAGE IF INPUT IS A FILE.
*
         INFILE  TTY                                :F(REREAD)
         OUTPUT = ' ENTER TEXT TERMINATED BY A    ' ENDTEXT
*
REREAD   TOTALTEXT = TOTALTEXT INPUTTEXT BLANK      :F(FRETURN)
         TOTALTEXT  ENDTEXT                         :F(REREAD)
*
*  SEE IF USER WISHES TO REDEFINE INPUT FILES AND OTHERS.
*  TEST FOR INPUT FROM TTY OR INPUT FILE TO BE END MARKS.
*  THE END OF FILE MARK IS ### STARTING IN FIRST CHAR POSITION.
*
         TOTALTEXT  ESCAPECODE                      :S(FRETURN)
         TOTALTEXT ENDTEXT REM = BLANK ENDTEXT
*
*  REMOVE ILLEGAL PUNCTUATION FROM STRING.
*
TEST     TOTALTEXT ANY(ILLEGALPUNCT) = BLANK        :S(TEST)
*
*  INSERT BLANKS ON EITHER SIDE OF ANY PUNCTUATION APPEARING
*  IN THE INPUT TEXT SO EACH WORD IS DELIMITED.
*
         T = 0
HERE     TOTALTEXT  POS(T) BREAK(PUNCTSYMNOBLANK) $ T1
+                SPAN(PUNCTSYMBOL) $ T2
+                = T1 BLANK T2 BLANK                :F(TEST2)
         T = SIZE(T1 T2) + T + 1                    :(HERE)
TEST2    TOTALTEXT = BLANK TOTALTEXT
*
```

```
*  REMOVE MULTIPLE BLANKS AND REPLACE BY SINGLE BLANK.
*
TEST3   TOTALTEXT  DOUBLEBLANK = BLANK            :S(TEST3)
*
*  SEE IF FLAG THAT SAYS TO OUTPUT THE INPUT TEXT TO CASSETTE ON.
*
        TEXTFLAG   SET                            :F(STATTEST)
*
*  REMOVE END OF TEXT MARKER BEFORE WRITING TO CASSETTE.
*
        TEMPTEXT = TOTALTEXT
        TEMPTEXT ENDTEXT = NULL
*
*  INSERT A QUOTE MARK BEFORE AND AFTER TEXT TO BE WRITTEN TO CAS.
*
        FILEOUT(QUOTE TEMPTEXT QUOTE)
*
*  SEE IF STATFLAG SET.  IF SO OUTPUT THE TEXT TO STAT FILE.
*
STATTEST  STATFLAG  SET                           :F(RET)
*
*  INSERT THE ACTUAL TEXT TO BE TRANSLATED TO THE STAT FILE.
*
        STATISTICS  =  '**' TOTALTEXT
*
RET     READTEXT = TOTALTEXT                      :(RETURN)
*
*
```

```
*****************************************************************
*
*       ASCII
*
*  THIS TRANSLATES TO ASCII.
*
*****************************************************************
*
*
ASCII   ASCII = NULL
*
*  REMOVE LEFT AND RIGHT BRACKETS.
*
        STRING = REPLACE(STRING,'[]','  ')
*
*  INSERT AN END BLANK SO FOLLOWING BREAK WILL WORK ON LAST WORD.
*
        STRING = STRING  BLANK
*
*  GET RID OF DOUBLE BLANKS SO THAT BREAK WILL NOT GET NULL SYMBOL.
*
AGAIN   STRING  DOUBLEBLANK = BLANK             :S(AGAIN)
*
*  REMOVE INITIAL BLANK IF ANY SO BREAK WON'T BREAK BEFORE IT.
*
        STRING POS(0) BLANK = NULL
LOOP    STRING  BREAK(BLANK) . ASCIISYM BLANK =          :F(RETURN)
        ASCIISYM  BLANK  = 'BLANK'
        ASCII = ASCII DIFFER(NULL,$(ASCIISYM '.CODE'))
+               $(ASCIISYM '.CODE')             :S(LOOP)
*
*
```

```
*********************************************************************
*
*       CLI
*
*  THIS INPUTS THE KIND OF TRANSLATION WANTED
*  THEN BUILDS VARIABLES IN 'IN' AND 'OUT'
*  TO TRANSFER INDIRECT TO THE CODE.
*
*********************************************************************
*
*
*       ENGLISH = 'ENGLISH' ! 'ENGLIS' ! 'ENGLI' ! 'ENGL' ! 'ENG' !
*+              'EN' ! 'E'
*       IPA = 'IPA' ! 'IP' ! 'I'
*       VOTRA = 'VOTRAX' ! 'VOTRA' ! 'VOTR' ! 'VOT' ! 'VO' ! 'V'
*       ASCI = 'ASCII' ! 'ASCI' ! 'ASC' ! 'AS' ! 'A'
*
*
CLI          RESPONSE = NULL
CLIRETRY            OUTPUT = ' WHAT TRANSLATION DO YOU WANT?'
        RESPONSE = INPUT BLANK
        RESPONSE  BREAK(PUNCTSYMBOL) . IN  ANY(PUNCTSYMBOL)
+               BREAK(PUNCTSYMBOL) . OUT
        IN      POS(0) ENGLISH RPOS(0) = 'ENGLISH'      :S(OUTTEST)
        IN      POS(0) IPA RPOS(0)   = 'IPA'            :S(OUTTEST)
        IN      POS(0) VOTRA RPOS(0)  = 'VOTRAX'        :S(OUTTEST)
        IN      POS(0) ASCI RPOS(0)   = 'ASCII'         :S(OUTTEST)
+                                                       F(ERRORIN)
*
OUTTEST OUT     POS(0) ENGLISH RPOS(0)   = 'ENGLISH'    :S(RETURN)
        OUT     POS(0) IPA RPOS(0)   = 'IPA'            :S(RETURN)
        OUT     POS(0) VOTRA RPOS(0)   = 'VOTRAX'       :S(RETURN)
        OUT     POS(0) ASCI RPOS(0)   = 'ASCII'         :S(RETURN)
+                                                       F(ERROROUT)
*
ERRORIN  OUTPUT = ' INITIAL TRANSLATION PARAMETER ILLEGAL'
        OUTPUT = ' PARAMETER IS ' IN            :(CLIRETRY)
ERROROUT  OUTPUT = ' FINAL TRANSLATION PARAMETER ERROR'
        OUTPUT = ' PARAMETER IS ' OUT           :(CLIRETRY)
*
```

```
*****************************************************************
*
*       FILEDEFINE
*
*****************************************************************
*
*
*  DEFINE FILENUMBERS FOR THE INPUT FILE THE STAT FILE AND THE
*  TRANS FILE.  THESE ARE USED IN THE VARIABLE ASSIGNMENTS TO
*  INDICATE THE I/O.
*
FILEDEFINE  INNO = 22
        STATNO = 23
        TRANSNO = 24
        STATFLAG = UNSET
        TTYFLAG = UNSET
        TEXTFLAG = UNSET
*
        OUTPUT = ' WHAT IS THE INPUT FILE NAME?'
        INFILE = INPUT
*
*  SEE IF IT IS THE TTY (INPUT DEVICE USING).
*
        INFILE  TTY                                :F(OKAY1)
*
*  YES IT IS SO REDEFINE INPUT FILE TO TTY = 2 ON THIS SYSTEM.
*
        INNO = 2                                   :(SOK1)
*
*  THE DEVICE IS NOT THE TTY SO MAKE CORRESPONDENCE WITH FILE
*  NAME AND DEVICE NUMBER.
*
OKAY1        IFILE(INNO,INFILE)
*
*  GET TRANSFILE NAME.
*
SOK1    OUTPUT = ' WHAT IS THE FILENAME FOR THE '
+                'TRANSLATION RESULTS?'
        TFILE = INPUT
        IDENT(TFILE,NULL)                          :F(NEXT)
        ENDFILE(TRANSNO)                           :(NEXTQ)
NEXT    TFILE    TTY                               :F(CASTEST)
*
*  REDEFINE FILENO TO TTY.
*
        TRANSNO = 2
*
```

```
*  SET FLAG TO SAY TTY OUTPUT.
*
        TTYFLAG = SET                           :(OKAY2)
*
*  SEE IF DEVICE IS THE CASSETTE.
*  SEE IF USER WISHES ORIGINAL TEXT TO BE WRITTEN.
*
CASTEST TFILE   CAS                             :F(ASKAGAIN)
*
*  YES IT IS CASSETTE SO DEFINE NO. TO TTY AND SET FLAG.
*
        TRANSNO = 2
ASKAGAIN  OUTPUT = ' TEXT TO FILE, TOO?'
        CASANS = INPUT
*
*  SEE IF ANSWER YES.
*
        CASANS   NOANS                          :S(OKAY2)
*
*  MAKE SURE ANSWER IS YES AND NOTHING ELSE.
*
        CASANS  YESANS                          :F(ASKAGAIN)
*
*  ALL IS OKAY AND ANSWER WAS YES.  SET FLAG.
*
        TEXTFLAG = SET
*
OKAY2   TFILE   POS(0) OLDTFILE RPOS(0)         :S(NEXTQ)
*
*  THIS IS A NEW FILE SO SAVE ITS NAME.
*
        OLDTFILE = TFILE
*
*  CLOSE THE OLD FILE.
*
        ENDFILE(TRANSNO)
*
*  MAKE NEW ASSIGNMENT.
*
        OFILE(TRANSNO,TFILE)
*
```

```
NEXTQ   OUTPUT = ' DO YOU  WANT TO GATHER STATISTICS?'
        ANS = INPUT
        ANS     NOANS                           :S(DEF)
*
*  STATISTICS ARE WANTED.
*
        OUTPUT = ' WHAT IS THE FILENAME?'
        STFILE = INPUT
        IDENT(STFILE,NULL)                      :S(NEXTQ)
*
*  SET FLAG TO INDICATE STAT GATHER.
*
        STATFLAG = SET
*
*  SEE IF STATS ARE TO BE SENT TO TTY.
*
        STFILE  TTY                             :F(OKAY3)
        STATNO = 2
OKAY3   STFILE  POS(0) OLDSTFILE RPOS(0)        :S(DEF)
*
*  NOT THE SAME STAT FILE SO SAVE THE NAME.
*
        OLDSTFILE = STFILE
*
*  CLOSE THE OLD STAT FILE.
*
        ENDFILE(STATNO)
*
*  REDEFINE THE STAT FILE NAME.
*
        OFILE(STATNO,STFILE)
*
*  SET UP VARIABLE ASSOCIATIONS.
*
DEF     INPUT('INPUTTEXT',INNO,80)
        OUTPUT('STATISTICS',STATNO,'(1X,15A5)')
        OUTPUT('TRANSTEXT',TRANSNO,'(1X,15A5)') :(RETURN)
*
*
```

```
****************************************************************
*
*       FILEOUT
*
*  THIS ROUTINE OUTPUTS VOTRAX MNEMONIC CODES TO A FILE.
*  EACH CODE IS SEPARATED BY A BLANK.
*  THE SEQUENCE IS PRECEDED BY A RECORD ON CODE MEANT
*  FOR THE 733 ASR CASSETTE TO TURN ON THE CASSETTE.
*  THE MESSAGE IS ENDED BY AN END OF MESSAGE CHARACTER
*  WHICH HAS MEANING TO THE SPEECH LAB PROGRAMS RUNNING
*  ON THE TI 960A, FOLLOWED BY A DELETE CODE TO WRITE OVER THE
*  RECORD OFF IN THE CASSETTE BUF, AND THE FINAL CODE IS A
*  RECORD OFF TO SHUT THE CASSETTE OFF.
*
*       DC2 = RECORDON
*       DC4 = RECORDOFF
*       '^' IS USED BY THE TI SPEECH LAB AS AN
*       END OF MESSAGE CODE.
*
*  ENDOFMESSAGE  IS INSERTED
*  BEFORE THIS ROUTINE IS CALLED IF IT IS WANTED IN THE RECORD.
*
****************************************************************
*
*
*  REMOVE BRACKETS ] AND [ FROM THE TEXT.
*
FILEOUT      TEMPOUT = REPLACE(BUF,'][',BLANK BLANK)
RELOOP1  TEMPOUT  DOUBLEBLANK = BLANK            :S(RELOOP1)
*
*  SEND THE TEXT TO THE FILE.
*  ALSO BREAK UP INTO BLOCKS AT A BLANK SO THAT
*  THE COMMUNICATIONS PROCESSOR DOESN'T ELIMINATE
*  IMPORTANT BLANKS.
*
        TEMPOUT = RECORDON TEMPOUT DUPL(DELETE,86)
+                RECORDOFF DELETE BLANK
REDOO  TEMPOUT (TAB(70) BREAK(BLANK)) . T =     :F(LAST)
        TRANSTEXT = T                           :(REDOO)
*
LAST    TRANSTEXT = TEMPOUT                     :(RETURN)
*
*
```

```
*****************************************************************
*
*
*  DEFINE SOME ERROR MESSAGES.
*
NORULEAPPLIES  OUTPUT = ' NO RULE APPLIES. RULES ATTEMPTING '
+               'TO USE ARE <' GRAPHEME 'RULE.' QUAL  '>'
        OUTPUT = ' THE CONTENTS ARE <' $(GRAPHEME 'RULE.' QUAL) '>'
        OUTPUT = ' CHARACTER ATTEMPTING TO PROCESS IS <'
+               GRAPHEME '>'                     :(REREED)
*
RULESYNTAXERROR  OUTPUT = ' SYNTAX ERROR IN RULE FORMATION '
+               'RULE IS ' RULE                  :(REREED)
*
EOF     OUTPUT = '  EOF ENCOUNTERED IN INPUT FILE'
        OUTPUT = '  DO YOU WISH TO CONTINUE? '
        ANS = INPUT
        ANS     NOANS                            :F(BEG)
        ENDFILE(TRANSNO)
        ENDFILE(STATNO)
DONE    OUTPUT = ' ALL DONE '
*
*
END
```

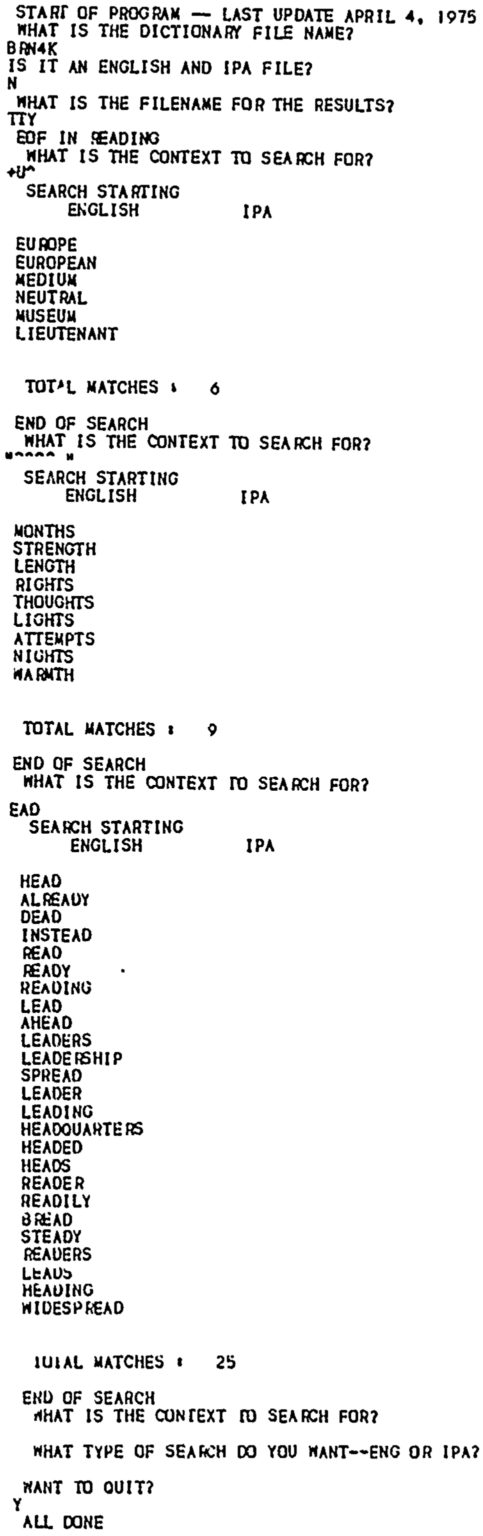
# Appendix B
# PROGRAM DOCUMENTATION FOR DICT

DICT searches an English dictionary file specified by the user for all the words which match a specified rule. The rule, similar to the rules for the translation program, consists of only the left part of the rule, as no translation is needed. Since the program must find all occurrences of a match, the brackets, '[' and ']', are not needed. Special symbols retaining their meaning from the translation program may also be used in the rule. A double quote '"' may be used to delimit the rule on the left or right but is necessary only to make a trailing blank unambiguous.

This program permits the testing of a proposed alteration or addition to the rules by finding all the words which would match the new rule. A sample dialog is shown in Fig. B1.

Initially the program requests the names of the dictionary file, the result file, and the size of the dictionary file. The input terminal (TTY) is accepted as a valid file name. The routine FILEDEFINE also makes the logical name correspondences between ENTRY and the dictionary file and RESULT and the output file.

The program will assign space for two arrays having the size specified by the user in FILEDEFINE. The routine READFILE inputs the dictionary file into these arrays, one array (ENGDICT) for the English text and the other array (IPADICT) for the IPA representation, if present. DICT can search the IPA array for a match with the rule if wished. Therefore the program determines from the user whether an English or IPA search is required. After this information is recorded, the new rule to be tested is read into RULE. The routine FIND scans through the specified dictionary array, searching for a match with the rule specified. On finding a match, the matched word is written to the output file. A count of the number of matches is kept in TOTAL and written after all the matches are found. When special symbols are included in the rule, a special pattern must be built. This pattern is built in the same way that SPECIALCASEPROC and SPECIALBREAK of the translation program builds the pattern.



```
 START OF PROGRAM — LAST UPDATE APRIL 4, 1975
 WHAT IS THE DICTIONARY FILE NAME?
BRN4K
IS IT AN ENGLISH AND IPA FILE?
N
 WHAT IS THE FILENAME FOR THE RESULTS?
TTY
 EOF IN READING
  WHAT IS THE CONTEXT TO SEARCH FOR?
+U^
  SEARCH STARTING
      ENGLISH             IPA

 EUROPE
 EUROPEAN
 MEDIUM
 NEUTRAL
 MUSEUM
 LIEUTENANT


  TOTAL MATCHES :    6

 END OF SEARCH
  WHAT IS THE CONTEXT TO SEARCH FOR?
"^^^^ "
  SEARCH STARTING
      ENGLISH             IPA

 MONTHS
 STRENGTH
 LENGTH
 RIGHTS
 THOUGHTS
 LIGHTS
 ATTEMPTS
 NIGHTS
 WARMTH


  TOTAL MATCHES :    9

 END OF SEARCH
  WHAT IS THE CONTEXT TO SEARCH FOR?
 EAD
  SEARCH STARTING
      ENGLISH             IPA

  HEAD
  ALREADY
  DEAD
  INSTEAD
  READ
  READY
  READING
  LEAD
  AHEAD
  LEADERS
  LEADERSHIP
  SPREAD
  LEADER
  LEADING
  HEADQUARTERS
  HEADED
  HEADS
  READER
  READILY
  BREAD
  STEADY
  READERS
  LEADS
  HEADING
  WIDESPREAD


   TOTAL MATCHES :    25

  END OF SEARCH
   WHAT IS THE CONTEXT TO SEARCH FOR?

   WHAT TYPE OF SEARCH DO YOU WANT--ENG OR IPA?

  WANT TO QUIT?
 Y
  ALL DONE
```

Fig. B1 — Sample dialog with DICT

DICT Program Listing

```
*****************************************************************
*
*                *****  DICT  *****
*
*  THIS PROGRAM SEARCHES A DICTIONARY FILE OF ENGLISH WORDS
*  AND THEIR IPA TRANSCRIPTIONS ACCORDING TO A RULE SPECIFIED
*  BY THE USER.
*
*****************************************************************
*
*
*  DEFINE THE FUNCTIONS.
*
*  FILEDEFINE ASKS FOR INPUT DICTIONARY FILE NAME AND RESULT FILE
*  NAME.  TTY IS LEGAL INPUT TO EITHER QUESTION.
*
       DEFINE('FILEDEFINE()')
*
*  READFILE INPUTS THE DICTIONARY FILE INTO THE ARRAYS
*  ENGDICT AND IPADICT.  TO DO THIS READFILE BREAKS EACH RECORD
*  OF THE FILE INTO TWO PIECES, THE ENGLISH AND IPA.
*
       DEFINE('READFILE()')
*
*  FIND IS THE ROUTINE WHICH SCANS THE DICTIONARY ARRAYS FOR A
*  RULE MATCH. ON FINDING ONE FIND OUTPUTS THE RESULT TO EITHER
*  THE TTY OR A SPECIFIED FILE AS PREDEFINED.
*  PARAMETERS ARE THE RULE SPECIFIED TO SEARCH ON,
*  THE ARRAY TO SEARCH EITHER IPA OR ENG,
*  AND THE INDEX SET--THIS IS IN CASE THE FILE IS AN INDEXED FILE.
*  IN THIS CASE THE USER MAY SEPECIFY AN INDEX SET--THIS FEATURE
*  NOT IMPLEMETED YET.
*
       DEFINE('FIND(RULE,QUAL,INDEX)')
*
*
*  MAIN PROGRAM STARTS HERE.
*
*
       INPUT('INPUT',2,80)
*
*  SET TRIM OPTION SO ALL INPUT DONE WITH TRAILING BLANKS TRUNCATED
*
       &TRIM = 1
       &STLIMIT = 100000000
       DICTSIZE = 4000
*
```

```
*  INIT SOME VARIABLES.
*
        NULL =
        BLANK = ' '
        DOUBLEBLANK = '  '
        SLASH = '/'
        INDEX = NULL
        ENDTEXT = '#'
        QUOTE = '"'
        TTY = POS(0) 'TTY' RPOS(0)
        YESANS = POS(0) ('Y' ! 'YES') RPOS(0)
*
*  INIT SOME VARIABLES USED IN SPECIAL CASE ROUTINES.
*  THIS INIT IS DONE IN BEGINNING  FOR EFFICIENCY.
*
        SPECIALCASE = '#*.$%&@^+:'
*
*  DEFINE THE SPECIAL PATTERNS TO BE USED.
*
        VOWEL = 'AEIOUY'
        CONSONANT = 'BCDFGHJKLMNPQRSTVWXZ'
        VOICED = 'BDVGJLMNRWZ'
        FRONT = 'EIY'
        SUFFIX = 'ER ' ! 'E ' ! 'ES ' ! 'ED ' ! 'ING ' ! 'ELY '
        SIBILANT = ANY('SCGZXJ') ! 'CH' ! 'SH'
        NONPAL = ANY('TSRDLZNJ') ! 'TH' ! 'CH' ! 'SH'
        $'PATTERN#' = ANY(VOWEL) ARBNO(ANY(VOWEL))
        $'PATTERN*' = ANY(CONSONANT) ARBNO(ANY(CONSONANT))
        $'PATTERN.' = ANY(VOICED)
        $'PATTERN$' = ANY(CONSONANT) ANY('EI')
        $'PATTERN%' = SUFFIX
        $'PATTERN&' = SIBILANT
        $'PATTERN@' = NONPAL
        $'PATTERN^' = ANY(CONSONANT)
        $'PATTERN+' = ANY(FRONT)
        $'PATTERN:' = ARBNO(ANY(CONSONANT))
        ENGL = 'ENG' ! 'E'
        IPAR = 'IPA' ! 'IP' ! 'I'
        ENGANS = POS(0) ENGL RPOS(0)
        IPAANS =  POS(0) IPAR RPOS(0)
*
*
*  START PROGRAM.
*
*
        OUTPUT = ' START OF PROGRAM — LAST UPDATE APRIL 4, 1975'
*
```

```
REDEFINE   FILEDEFINE()
*
*  DEFINE THE ARRAYS.
*  4000 IS CURRENTLY THE LIMIT ON THE SIZE  OF THE ARRAYS.
*
        ENGDICT = ARRAY(DICTSIZE)
        IPADICT = ARRAY(DICTSIZE)
*
*  READ IN THE FILE AND PLACE INTO ARRAYS.
*
        READFILE()
*
*  IF SINGLE FILE SET SEARCHTYPE AUTOMATICALLY ENG.
*
        SINGLEFLAG 'ON'                                :S(QUALSET)
*
*  QUERY FOR SEARCH TYPE.
*
SEARCHCHANGE OUTPUT = ' WHAT TYPE OF SEARCH DO YOU WANT--ENG OR IPA?'
        QUAL = INPUT
*
*  SEE IF NOTHING RESPONDED. IF NONE MAY WISH TO QUIT.
*
        IDENT(QUAL,NULL)                               :S(QUIT)
        QUAL ENGANS = 'ENG'                            :S(ASKAGAIN)
        QUAL IPAANS = 'IPA'                            :F(SEARCHCHANGE)
*
*  QUERY FOR PATTERN TO SEARCH FOR.
*
ASKAGAIN  OUTPUT = ' WHAT IS THE CONTEXT TO SEARCH FOR?'
        RULE = INPUT
        RULE POS(0) QUOTE = NULL
        RULE QUOTE RPOS(0) = NULL
*
*  SEE IF USER WISHES TO REDEFINE THE SEARCH TYPE.
*  IF SO A NULL ANS GIVEN.
*
        IDENT(RULE,NULL)                               :S(SEARCHCHANGE)
*
        OUTPUT = ' SEARCH STARTING '
        RESULT = '     ENGLISH          IPA  '
        RESULT = ' '
*
*  FIND THE APPLICABLE ENTRIES.
*
        FIND(RULE,QUAL,INDEX)
*

        RESULT = ' '
        RESULT = ' '
        RESULT = ' TOTAL MATCHES :    ' TOTAL
        RESULT = ' '
        OUTPUT = ' END OF SEARCH.'                     :(ASKAGAIN)
*
QUALSET   QUAL = 'ENG'                                 :(ASKAGAIN)
*
*
```

```
***************************************************************
*
*       FIND
*
*  DEFINE FIND ROUTINE WHICH SEARCHES THE DICTIONARY
*  REQUESTED FOR THE SEQUENCE OF CHARACTERS PASSED.
*
*  PARAMETERS ARE:
*
*  RULE          WHICH INDICATES THE SEQUENCE OF CHARACTERS
*                TO SEARCH FOR IN THE DICTIONARY.
*                A SPECIAL CASE SYMBOL MAY BE USED.  IN THIS CASE
*                SPECIALCASEPROC IS USED TO BUILD A PATTERN.
*
*  QUAL          WHICH INDICATES WHETHER THE ENGLISH (ENG) OR
*                OR IPA  DICTIONARY IS TO BE SEARCHED.
*
*  INDEX         WHICH INDICATES WHICH ENTRIES IN THE DICTIONARY MAY
*                FULLFILL THE RULE REQUIRED.
*                IF INDEX IS NULL, A SEQUENTIAL SEARCH IS PERFORMED.
*
***************************************************************
*
*  IT IS ASSUMED THAT ENGDICT AND IPADICT ARE INITIALIZED.
*  THIS INITIALIZATION IS DONE BY THE READDICT ROUTINE.
*
*  SEE IF ANY SPECIAL SYMBOLS OCCUR IN THE RULE PASSED.
*  IF SO SPECIALCASEPROC MUST BE INVOKED.
*
FIND    TOTAL = 0
        RULE    ANY(SPECIALCASE)                 :S(SPECIALCASEPROC)
*
*  NO SPECIAL CASE SYMBOLS OR ELSE RETURNED FROM PATTERN BUILDING
*  OF THE SPECIALCASEPROC.
*
INDEXTEST  IDENT(INDEX,NULL)                     :S(INC)
*
*  THERE ARE INDEXES SPECIFIED—GET THEM ONE BY ONE.
*
NEXT    INDEX   BREAK(', ') . I  ANY(', ')  = NULL      :F(RETURN)
*
*  SEE IF THE FIRST ENTRY SUGGESTED MATCHES THE RULE PASSED.
*  BUILD THE NAME OF THE ARRAY TO BE CHECKED.
*  INCLUDE THE ENTRY TO CHECK.  THIS IS INDICATED BY VARIABLE I.
*
        ITEM($(QUAL 'DICT'),I)  RULE            :F(NEXT)
        TOTAL = TOTAL + 1
        RESULT = ENGDICT<I> '           ' IPADICT<I>     :(NEXT)
```

```
*
*  SET THE INDEX TO 1.
*
INC      I = 1
*
*  SEE IF RULE APPLIES--IF SO; OUTPUT RESULTS.
*  CONTINUE SEARCH.
*
ITEM     ITEM($(QUAL 'DICT'),I)  RULE             :F(NEXT2)
*
*  SEE IF SINGLE SEARCH AND ONLY TO PRINT ONE ARRAY.
*
         SINGLEFLAG 'ON'                          :S(ONEOUT)
         TOTAL = TOTAL + 1
         RESULT = ENGDICT<I> '           ' IPADICT<I>
NEXT2    I = LT(I,SIZE)  I + 1                    :F(RETURN)S(ITEM)
*
ONEOUT   TOTAL = TOTAL + 1
         RESULT = ENGDICT<I>                      :(NEXT2)
*
*  DEFINE THE SPECIALCASE ROUTINE TO BUILD PATTERN AS SPECIFIED
*  BY THE SPECIAL SYMBOLS #*.$%&@^+ AND : .
*  ALSO NOTE THAT THESE SYMBOLS AND THEIR CORRESPONDING
*  PATTERNS ARE INITIALIZED IN THE BEGINNING OF THE
*  PROGRAM FOR EFFICIENCY.
*  THEY APPEAR HERE AS COMMENTS FOR READABILITY.
*
*
*        SPECIALCASE = '#*.$%&@^+:'
*        VOWEL = 'AEIOUY'
*        CONSONANT = 'BCDFGHJKLMNPQRSTVWXZ'
*        VOICED = 'BDVGJLMNRWZ'
*        FRONT = 'EIY'
*        SUFFIX = 'ER ' ! 'E ' ! 'ES ' ! 'ED ' ! 'ING ' ! 'ELY '
*        SIBILANT = ANY('SCGZXJ') ! 'CH' ! 'SH'
*        NONPAL = ANY('TSRDLZNJ') ! 'TH' ! 'CH' ! 'SH'
*        $'PATTERN#' = ANY(VOWEL) ARBNO(ANY(VOWEL))
*        $'PATTERN*' = ANY(CONSONANT) ARBNO(ANY(CONSONANT))
*        $'PATTERN.' = ANY(VOICED)
*        $'PATTERN$' = ANY(CONSONANT) ANY('EI')
*        $'PATTERN%' = SUFFIX
*        $'PATTERN&' = SIBILANT
*        $'PATTERN@' = NONPAL
*        $'PATTERN^' = ANY(CONSONANT)
*        $'PATTERN+' = ANY(FRONT)
*        $'PATTERN:' = ARBNO(ANY(CONSONANT))
*
*
```

```
*  START THE ROUTINE HERE.
*
SPECIALCASEPROC   PATTERN = NULL
*
*  REPLACE EVERYTHING UP TO THE SPECIAL CHAR BY NULL AND ASSIGN
*  WHAT MATCHED TO THE PATTERN BEING BUILT.
*
REMATCH  RULE   BREAK(SPECIALCASE) . PATTERN1 = NULL    :F(ALLDONE)
*
*  BREAK OFF SPECIAL CHAR AND REPLACE IT BY NULL IN ORIGINAL STREAM.
*
         RULE     LEN(1) . SYM = NULL
*
*  FIND THE PIECE OF PATTERN WHICH CORRESPONDS TO THE SPECIAL SYM.
*
         PATTERN = PATTERN  PATTERN1 $('PATTERN' SYM)    :(REMATCH)
*
*  AT CONCLUSION OF SCAN THRU RULE RETURN WITH RULE INIT TO
*  THE PATTERN WHICH WAS BUILT.
*
ALLDONE   RULE = PATTERN  RULE                      :(INDEXTEST)
*
*
```

```
******************************************************************
*
*       READFILE
*
*  DEFINE THE READFILE ROUTINE.
*  READS IN THE DICTIONARY FILE AS SPECIFIED BY THE USER.
*  BUILDS TWO ARRAYS FOR THE PROGRAM BASED ON THIS FILE.
*  THE ARRAYS ARE THE IPA AND THE ENGLISH DICTIONARY ARRAYS.
*  THE INPUT FILE MUST BE IN THE FORM :
*
*   ENGLISHWORD ENDTEXTMARK  IPAWORD  ENDTEXT
*
******************************************************************
*
*
READFILE   I = 1
        SINGLEFLAG 'ON'                                :S(READONE)
READAGAIN  ENTRY  BREAK(ENDTEXT) . ENGDICT<I>  ENDTEXT
                BREAK(ENDTEXT) . IPADICT<I>     :F(ERRORINREAD)
        ENGDICT<I> = BLANK ENGDICT<I> BLANK
        IPADICT<I> = SLASH BLANK IPADICT<I> BLANK SLASH
        IPADICT<I>  POS(1) DOUBLEBLANK = BLANK
        IPADICT<I>  DOUBLEBLANK RPOS(1) = BLANK
*
*  SEE IF INDEX IS OUT OF RANGE OF SIZE OF FILE.
*
        I = LT(I,SIZE)  I + 1            :F(RETURN)S(READAGAIN)
*
*  NOTE THAT ENTRY RESULTS IN ONE RECORD BEING READ FROM THE
*  INPUT FILE WHICH WAS SPECIFIED BY THE USER.
*
READONE  ENGDICT<I> = BLANK ENTRY                      :F(ERRORINREAD)
        ENGDICT<I>  ENDTEXT = BLANK
        I = LT(I,SIZE)  I + 1                          :S(READONE)F(RETURN)
*
ERRORINREAD  OUTPUT = ' EOF IN READING '
        SIZE = I - 1                                   :(RETURN)
*
*
```

```
****************************************************************
*
*       FILEDEFINE
*
*  DEFINE THE ROUTINE TO GET INPUT AND OUTPUT FILE NAMES.
*
****************************************************************
*
*
FILEDEFINE  INNO = 22
        SINGLEFLAG = 'OFF'
        SIZE = DICTSIZE
        OUTNO = 24
        OUTPUT = ' WHAT IS THE DICTIONARY FILE NAME? '
        DICT = INPUT
*
*  SEE IF IT IS TTY.
*
        DICT    TTY                             :F(OKAY)
        INNO = 2
        OUTPUT = ' WHAT IS THE SIZE OF THE INPUT FILE?'
        SIZE = INPUT
*
*  THIS RESULT IS NEEDED FROM THE EXTERNAL FILE ALSO.
*
OKAY    IFILE(INNO,DICT)
        OUTPUT = ' IS IT AN ENGLISH AND IPA FILE?'
        INPUT YESANS                            :S(DEFINEOUT)
        SINGLEFLAG = 'ON'
DEFINEOUT  OUTPUT = ' WHAT IS THE FILENAME FOR THE RESULTS?'
        OUTFILE = INPUT
        OUTFILE TTY                             :F(FILE)
        OUTNO = 2
FILE    OUTFILE POS(0) OLDOUTFILE RPOS(0)       :S(DEF)
        OLDOUTFILE = OUTFILE
        ENDFILE(OUTNO)
DEF     OFILE(OUTNO,OUTFILE)
        INPUT('ENTRY',INNO,80)
        OUTPUT('RESULT',OUTNO,'(1X,15A5)')      :(RETURN)
*
*
****************************************************************
*
*
*  DEFINE WHAT HAPPENS ON NULL REPONSE TO SEARCH TYPE.
*
QUIT    OUTPUT = ' WANT TO QUIT?'
        INPUT POS(0) 'N'                        :S(REDEFINE)
        ENDFILE(OUTNO)
        OUTPUT = ' ALL DONE '
*
*
END
```

# Appendix C
# CONVERSION OF SOFTWARE TO FASBOL

The SNOBOL processor on the PDP-10 system is an interpretive implementation of SNOBOL 4.* Since TRANS is itself an interpreter for the letter-to-sound rules, when it is running on the SNOBOL processor, it suffers from all the inefficiency one would expect of an interpreter interpreting an interpreter. TRANS was never intended for production runs; it and the other SNOBOL programs we wrote are research tools to facilitate the development of the letter-to-sound rules. We were therefore prepared to pay a price in efficiency for the convenience of working in a high-level pattern-matching language and being able to change the program or rules easily. We could not remain completely indifferent to efficiency however; we found that translating a single 1000-word sample from the Brown Corpus required an overnight computer run of many hours. When a complied version of SNOBOL became available to us, we consequently converted our software to take advantage of it.

The compiler, FASBOL II,† became usable on NRL's PDP-10 shortly before we were ready to start translating large samples from the Brown Corpus with version 3 of the rules. The FASBOL version of DICT was ready soon enough to be used in part of the work on version 3, and none of the third, longest, series of translations of large samples were begun until TRANS had been converted. STAT was converted as well.

The program sections that open and close input and output files had to be rewritten, but the source languages for the SNOBOL interpreter and the FASBOL compiler are compatible enough that no other significant changes would have been necessary. We made some further changes, following suggestions in the FASBOL manual† for enchancing the efficiency of FASBOL programs. We also used a FASBOL feature that provides a statement-by-statement analysis of execution time; once we had identified the critical statements, we tried rewriting them to speed up the programs further. This last attempt met with such indifferent success as only to reinforce a conclusion we had reached working with SNOBOL: one's intuition of what ought to be fast is no guide to what is fast.

After conversion to FASBOL, TRANS ran about 25 times as fast as it had before. DICT, while simply reading words from a file and storing them in an array, ran 35 times as fast after conversion; while searching the array for the words that match a pattern, it ran 3 to 8 times as fast after conversion. These speedup factors do not necessarily reflect the intrinsic difference in speed between the FASBOL and SNOBOL systems, since some of the program changes might well have increased the speed of the SNOBOL version as well; others might even have slowed it down. Nevertheless the speed increase brought with it a substantial increase in convenience. Translation rates are up from one word every half-minute or minute to one word every second or two. This is within a factor of 4 or 5 of real-time speech rates, and an implementation designed for effciency rather than convenient experimentation, by stripping away another layer of interpretive overhead, would certainly run much faster than that.

*R.E. Griswold, J.F. Poage, and I.P. Polonsky, *The SNOBOL 4 Programming Language*, Prentice-Hall, Englewood Cliffs, N.J., 2nd edition, 1971.

†P.J. Santos, Jr., "FASBOL II, a SNOBOL Compiler for the PDP-10," DECUS No. 10-179, Digital Equipment Computer Users' Society, Dec. 1972.

Not only did conversion to FASBOL increase our programs' speed, it reduced their memory requirements, in some cases threefold. It thus became possible to run DICT on much larger word lists than before.

We are reproducing the SNOBOL versions of the programs in this report, since SNOBOL 4 interpreters are more widely available than the FASBOL compiler. The FASBOL versions are available from the authors.