38

DERIVING AND TABULATING ENGLISH SPELLING-TO-SOUND CORRESPONDENCES

by

Richard L. Venezky

Technical Report #45

September 1968

ABSTRACT

A series of programs has been written for deriving and tabulating English spelling-to-sound and sound-to-spelling correspondences. One program scans the spelling and pronunciation of a word and decides which sounds in the pronunciation are derived from which letters in the spelling. Alternate vowel and consonant clusters are formed in each symbol string (spelling and pronunciation) and are matched by a routine that searches for such irregularities as silent letters and vocalic consonants. A dictionary containing the spelling and pronunciation of approximately 20,000 English words is the input to this program. The output is a separate record for each spelling-to-sound correspondence in a word, including the spelling of the word, its code number in the dictionary, its frequency of occurrence, and the correspondence type (consonant or vowel). These records are then sorted and input to another program which tabulates the frequency of occurrence of every spelling-to-sound and sound-to-spelling correspondence found by the first program. Separate tabulations are made for the different positions which the spelling cluster occupies in the words in which it appears (initial, medial, and final positions). This information along with the words which comprise each correspondence statistic are then listed.

Separate listings of these same data for the 5,000 most common words in English, and for the graphic monosyllables have also been obtained, along

with reversed and alphabetized lists of spellings and of pronunciations.

Data obtained from this study represent the most complete tabulation of spelling-to-sound and sound-to-spelling correspondences ever compiled. These results have formed the basis of a recently published analysis of English orthography and are being used in a number of universities in research aimed towards the improvement of reading and spelling instruction.

I. INTRODUCTION

1.0 The English writing system evolved sometime around the Seventh Century and was derived primarily from the extant Latin alphabet. In its original form it is assumed to have been a phonemic writing system; that is, each graphemic symbol corresponded to a unique significant set of sounds (phoneme) in the language.[1] (This was probably not one-hundred percent true, even in the Seventh Century). As the sound system followed its natural and apparently random course of change through the centuries, the graphemic system changed also, but the directions of change were seldom parallel.

The vowel sounds shifted, the initial /g/ and /k/ sounds before consonants were dropped, and numerous other phonetic and phonemic changes occurred, but the spelling rules either remained rigidly attached to the archaic forms (e.g., house, gnat, knee, light), or compounded the chaos through additional changes (e.g., of, quack, where). By the time of the last changes in the graphemic system in the seventeenth and eighteenth centuries, the original direct correspondences between graphemes and phonemes had become, from all outward appearances, a twisted mass of irregular correspondences. In spite of these problems, however, most people today learn to relate graphemes and phonemes that is, can predict with some accuracy the pronunciations of words they have never encountered before.

1.1 The exact nature of the correspondences between spelling and sound, however, is at present a concern of several different disciplines. Primary school teachers need this information to teach reading more efficiently than they do now. To facilitate this learning process, the first words that a child learns to pronounce in reading should be those words that have the most regular spelling-to-sound correspondences; that is, those that have the greatest transfer value to new words that the child will meet. This requirement holds true for teaching reading to adults and for second language learning, where a knowledge of spelling-to-sound correspondences that differ from those in one's native language serves as a short cut towards pronouncing the new language.

In the development of a reading machine, a machine that will optically scan written material and read the text aloud, a spelling-to-sound correspondence network in connection with a dictionary of exceptions could serve as a communication link between the optical scanner and the vocal output unit. While this link could conceivably be a large computer which stores the pronunciations of every word in the English language, such a system is not practical and should not be necessary once all spelling-to-sound correspondences for English are know.[2]

While numerous studies of spelling-to-sound correspondences have been made in the past, none have tabulated the frequencies of occurrences of all occurring correspondences in all positions within words and none have produced exhaustive listings of words with these correspondences. What is needed is, first, a complete tabulation of every spelling-to-sound correspondence that occurs in the words found in an average recognition vocabulary (which I am arbitrarily assuming in this report to be in the vicinity of 20,000 words), and, second, a listing of these words dichotomized into regular and irregular pronunciation groups. The labels "regular" and "irregular" will, of course, be designated on the basis of the spelling-to-sound statistics derived in the tabulation process.

1.2 The Reading Research Project at Cornell University,[3] in studies directed toward the designing of efficient reading materials for the primary schools, started compiling this information in 1959. A complete study of graphic monosyllables and a similar study of disyllables were done under the direction of C. F. Hockett.[4] Both of these studies were done completely by hand and required almost two years to complete. It became evident with the conclusion of this work that only a large computer could be expected to complete the task in a reasonable length of time. With the impending purchase of a Control Data 1604 computer by Cornell, the final impetus was given for finding a machine solution to this problem, and the programs described here were begun.

The purpose of this paper is to describe the programs that have been written for deriving and tabulating spelling-to-sound correspondences for English. While the motivation for and the primary orientation of this work is toward the improvement of teaching materials for the primary grades, the material derived from this project is equally applicable to the other two areas mentioned in the introduction.[5]

2. THE PROBLEM

Programming a computer to derive spelling-to-sound correspondences involves two distinct processes. In the first, correlations are derived for each dictionary word and stored. In the second, tabulations are made of the correspondences formed in the first part, based upon position in the word and possible environmental influences. Since the second part involves for the most part standard sorting procedures and presents no problems unique to the derivations of spelling-to-sound correspondences, it will be discussed only briefly in section 4. The first part of this program, however, is more complex and will be discussed in detail.

2.1 For purposes of this discussion, a _grapheme_ (G) is any of the 26 letters of the English alphabet. A _dictionary word_, or _graphemic word_, is a sequence of graphemes that appears as a single entry in a dictionary.[6] A _phoneme_ (P) is any of the forms used in the Thorndike Century Senior Dictionary to indicate the pronunciation of a graphemic word.[7] These forms along with their IPA equivalents are shown in Appendix A . (IPA symbols will be used in this paper.) Some phonemes are represented in print by the same forms used to represent graphemes. Specific graphemes will be italicized, _e.g.,_ _x,_ while specific phonemes will be enclosed by diagonal lines, e.g., /x/ .

2.2 The basic task is to derive a program that accepts as input a sequence of graphemes which compose (spell) an English word and a correponding sequence of phonemes and decide which phonemes correspond with which graphemes. If the English alphabet were an ideally phonemic alphabet so that each grapheme had one and only one corresponding phoneme, then this would be a trivial matching problem. Unfortunately, this is not the case in English. Given a sequence of only three graphemes which correspond to a string of three phonemes, any of the correspondence patterns shown in Figure 1 and undoubtedly many more are possible. As the number of graphemes increase linearly, furthermore, the number of possible correspondence patterns increases in an irregular manner. This problem is illustrated further in Figure 2 where a number of possible correspondence patterns for one-, two-, and three-grapheme sequences are shown.

Figures 1 and 2 here

2.3 In the abstract sense there exists a finite set of elements, called graphemes, which can be connected in a string, a string being any finite, linear sequence of these elements. Well-formed strings are formed by applying construction rules to the set of graphemes, the construction rules being an open ended set of rules that generates from the graphemes all and only those words that occur in the dictionary at that time.

Corresponding to every well-formed string of graphemes there exists a string of phonemes into which the graphemes can be mapped. The phonemes, like the graphemes, are selected from a finite set and are combined into phoneme strings according to a set of construction rules for phonemes. The basic problem is to find a minimal description of the complete mapping.

If every possible mapping were possible, then one could go no further than describe, for every given grapheme string, the particular corresponding phoneme string. This is not true, however, in English. There are, as illustrated in Figure 3 a large number of mappings that do not occur in English. Those mappings shown in Figure 3 violate rules on what is called the first level. This level contains all rules that can be stated without reference to particular phonemes or graphemes.

Figure 3 here

What is desired from the set of all possible mappings is the minimum number of rules necessary for an accurate and complete correspondence deriving process. Given only the rules on level one, the problem can probably be solved. The procedure would be to write a program which would form for every phoneme-grapheme pair a prediction pool of all possible correspondence structures, using only rules on level one. When this is completed, it would begin with grapheme strings of length one, which have only one possible structure, and store in a probability pool the single grapheme-phoneme correspondences found.

It would then be instructed to search for all occurrences of particular graphemes in different positions in words of length two, and, using the information in the prediction pool and the probability pool and a set of decision rules, derive a set of correspondences for all graphemes found in two-grapheme words. This process would then be repeated for three-grapheme strings, four-grapheme strings, and so on, using the information derived in each case in the examination of the next graphemic word-length. Eventually (we assume) this program should derive all spelling-to-sound correspondences that exist in the words fed into it. Unfortunately, there is no guarantee that such a system would always converge to the correct solution; if it did, furthermore, the decision rules would probably be more complex than necessary.

2.4 The more practical approach is to use rules on level two, rules that apply to particular graphemes and phonemes, to reduce further the number of possible structures, and then to design a routine that employs these rules to form correspondences. This is the general approach that has been employed in the program described here. The rules on level two limit the particular phonemes that can correspond to particular graphemes. The computing procedure, which was derived by a trail-error process, forms alternate clusters of vowels and consonants in both the grapheme words and the phoneme words, searches the clusters for irregularities, and finally matches corresponding grapheme-phoneme clusters. Special routines determine whether w and y function as vowel letters or consonant letters in a particular word, whether a particular interconsonantal vowel letter or an intervocalic consonant letter is silent, and whether r and gh are members of vowel letter clusters or consonant-letter clusters.

A special routine continually senses for any correspondences that are formed in violation of any of the rules on either level one or level two. If violations are found, they are stored and later printed out for examination.

2.5 What the program does, therefore, is to start at the beginning of the grapheme word and form a cluster of all consecutive graphemes that are the same type (vowel or consonant). As soon as a grapheme is found that is not the same type as the grapheme before it, the program jumps to the phoneme word and repeats the process. Once a pair of clusters has been formed, a matching routine tests for such irregularities as silent interconsonantal vowels, silent initial consonants, and vocalic consonants (consonants that have a vowel quality to their pronunciation, like l in table). Then the code number, spelling, position of stress, and frequency of occurrence of the entry are stored and the process is repeated on the next dictionary word.

3. PROGRAM DETAIL

The correspondence-forming program and the other programs to be described presently are written in a combination of JOVIAL, FORTRAN, and CODAP1 for the Control Data 1604-A, and have been run at the Control Data Programming Center in Palo Alto, California.

3.1 Input words were card-punched from the Thorndike Century Senior Dictionary with one dictionary entry on each card, and then stored on magnetic tape. Each entry consists of a grapheme field (spelling), phoneme field (pronunciation), and a frequency number, if it exists. All frequency numbers are given in thousands, but only the 20,000 most common words in the Thorndike Century Senior Dictionary have frequency numbers.

A "1", therefore, means that the word is one of the thousand most commonly used words; a "2" means that the word is one of the second thousand most commonly used words, and so on up to "20". While these numbers reflect the usage of nearly a quarter century ago and were compiled by a group that exhibited strong learnings toward mythological and botanical terms, they represent the only available information of this type and can be easily discarded if they are found to be totally useless.

The Thorndike Century Senior Dictionary was selected for this study because it includes frequency numbers for common words and also because it uses a pronunciation system which closely approximates that employed in

John S. Kenyon and Thomas Knott, _A Pronouncing Dictionary of American English_. (Springfield, Mass. 1951.)

3.2 Since there are no standard keypunch symbols for vowels with diacritical markings, stress, juncture, and the phonemes /š, č, ž, θ, ð/, symbols and symbol sequences had to be assigned to these items. To aid the keypunch operator, an attempt was made to select symbols that appeared similar to the dictionary markings they represented. These symbols are listed in Appendix A, in the columns labeled 'Keypunch and tape coding'.

Phonemes with diacritical markings were punched with a sequence of two symbols, the first being the base form and the second being the symbol representing the diacritical marking. The phonemes /š, č, ž, θ / were punched as the two symbol pairs: sh, ch, zh, th, and converted by a subroutine to a single symbol. /ð/ was punched as a left parenthesis.

3.3 In the G-cluster formation process, a master table indicates whether a grapheme is to be classified as part of a vowel cluster or part of a consonant cluster. With the graphemes r, l, n, y, gh, h, the table directs the program to subroutine for additional processing. These subroutines are explained in Section 5. For each grapheme, the table is consulted, a subroutine is employed if necessary, and finally, the grapheme is stored either as part of the present cluster or as the first item in the next cluster, depending on whether or not the grapheme-type is the same as the cluster-type being formed. Each grapheme is also entered in the appropriate memory location in the output field.

The first grapheme in the word is automatically stored as the first item of a cluster and additional graphemes are added to that cluster until a grapheme is found whose type (vowel or consonant) is different from the cluster type. This item is stored as the first item of the next cluster and the program jumps to the P-cluster forming process.

3.4 The forming of P-clusters is identical to the G-cluster forming, except that the phonemes are not stored in the output field until they have been matched with the appropriate graphemes. Special routines are used only for the phonemes /w/, /j/, /l/, and /r/, and for converting the input symbols for diacritical markings, stress, and two symbol phonemes to internal symbols. These routines are explained in Section 6.

3.5 At the beginning of the matching routine, a subroutine tests for a silent interconsonantal vowel immediately following the cluster. If one is found, it is marked as silent and stored in the output area. The program then goes to the G-clustering process and adds the next consonant cluster to the present one, and returns to the matching process.

Subroutines for initial and final G-clusters test for silent G's. If there is no P cluster in final position to correspond to the final G-cluster, the G cluster is marked as silent. The program will list two consecutive silent G-clusters as a fault. If the clusters in initial position are not the same type, the program tests for silent <u>h</u>, or for <u>y</u> phonemicized by

Thorndike as /ū/ as in you, or for grapheme words beginning on- that are phonemicized as /wu-/ as in one . Any other vowel-to-consonant correspondence is listed as a fault. Examples of these irregularities are shown in Figure 4.

Figure 4 here

3.6 The correspondences along with the corresponding identifying information (code number, graphemic form, and frequency of occurrence) are output onto magnetic tape in 2000 word blocks. These records are then sorted, using the standard CDC Sort/Merge routine.[8] Sorting keys, in order of importance are:

1. correspondence type (consonant or vowel)
2. grapheme string
3. phoneme string
4. position of grapheme string in word (initial, medial, final)
5. spelling of word

4. TABULATION

4.1 The output from the sort/merge routine is read in by a second program (TABUL) which tabulates the frequency of occurrence of each grapheme-phoneme cluster correspondence. Separate tabulations are made for initial, medial, and final positions; that is, the tabulation lists for each correspondence found, the percentage of that total which occurred as the first cluster in a word, the percentage which occurred as the final cluster, and the percentage which occurred as the medial cluster (defined as not initial and not final).

While these statistics are being compiled and stored, the word lists for each statistic are being output to the line printer. A page from the word list output is shown in Figure 5.

Figure 5 here

The tabulation program forms two first-letter tables for the grapheme clusters, one for vowel clusters and one for consonant clusters (the type of a cluster is marked during the correspondence-forming process). Branching from each first-letter position is a string of all the clusters that begin with that particular letter. Each cluster in this string has a sub-string of all the phoneme clusters that correspond to it, and each phoneme cluster has a sub-string that includes the frequency counts and the code number of the first word processed in which the particular correspondence occurred.

Once all of the output blocks from the correspondence-forming program have been processed, the lists are retrieved, the statistics tabulated, and the information inserted into the proper format and output. A page of this output data is shown in Figure 6. By reversing the positions of the grapheme and phoneme clusters in Program TABUL, sound-to-spelling correspondences were also tabulated. In addition, the entire program was rerun twice, once on the 5000 most common English words and once on the graphic monosyllables.

Finally, the spelling of each word was reversed, and the records sorted into alphabetical order; the same was also done with the pronunciations, yielding two separate backwards listing which have been invaluable for studying suffixes and other word endings.

Figure 6 here

5. GRAPHEME SUBROUTINES

5.1 If the character before r is a vowel or r, preceded by a vowel, r is classed as part of the vowel cluster. If the character before r is not a vowel, then the program tests for syllabic r; that is, tests to see if r corresponds to /ə r/. If r is syllabic, the P-clustering routine is set to include both /ə / and /r/ in a consonant cluster.

5.2 The program tests for syllabic l in the same manner as it checks for syllabic r .

5.3 If h is preceded and followed by vowels, the program determines if it has a consonant phoneme correspondent. If not, h is included in the vowel cluster with the vowel before it. Otherwise, h is classed as a consonant.

5.4 Initial w is classed as a consonant and final w is classed as a vowel. w preceded by a consonant is a consonant. If w occurs in any other position, the phoneme word is examined for a consonant correspondent for w . If none is found, w is included in a vowel cluster.

5.5 Final y is classed as a vowel and initial y is classed as a consonant. Vowel phonemicizations of initial y are detected by a later subroutine in the matching process. If y occurs in any other position, the program examines the phoneme word for a consonant correspondent for y . If none is found, y is classed as a vowel.

5.6 If g is preceded by a vowel and followed by h , and does not correspond to /f/, /k/, or /p/, the cluster gh is included with the vowel cluster.

6. PHONEME SUBROUTINES

6.1 If /w/ is followed by a vowel, the program examines the G-cluster for a corresponding w . If none is found, /w/ is classed as a vowel. Otherwise, /w/ is classed as a consonant.

6.2 The /j/ subroutine is the same as 6.1.

6.3 The character immediately following /s/ is examined. If it is /h/, the program then examines the grapheme word to see if graphemes s and h both occur at word boundaries in a compound word like sheepshead . (Word boundaries are marked during keypunching). If s and h do not occur as word boundaries, then s is replaced by the internal symbol for the /š/, /h/ is deleted, and the remaining P's are moved up one place in the input area.

6.4 The subroutines for /θ, č, ž/ are the same as 6.3.

6.5 The character after /a/ is examined. If it is a symbol which represents a diacritic, then the internal symbol for the vowel with the diacritic is substituted for /a/, the diacritic is deleted, and the remaining P's are shifted up one place in the input table. The program then returns to the P-cluster formation process.

6.6 The subroutines for /e, i, o, u/ are the same as 6.5.

6.7 The stress mark is removed and its position is recorded in the first memory location in the output area. The remaining P's are shifted up one location in the input area and the program returns to the P-cluster process.

6.8 Whenever a fault occurs, a jump is made to the fault routine. This routine records in the fault storage the fault type and the code number of the dictionary word being examined. After 15 faults occur, the program outputs the fault area for inspection. The fault area is also output at the end of the entire correspondence program. After a fault has been detected and processed, the program starts processing the next dictionary word. The following faults are sensed by the program:

1. q not followed by u .

2. A vowel cluster with more than five graphemes or phonemes.

3. A consonant cluster with more than five graphemes or phonemes.

4. Unmatched P's.

5. Non-correspondence clusters.

6. Two successive silent G-clusters.

7. Medial consonant grapheme cluster corresponding to a vowel phoneme cluster.

8. Silent G-cluster with more than two items.

7. RESULTS AND CONCLUSIONS

Output from these programs has been distributed on microfilm to a number of research groups and has formed the basis of an extensive analysis of the patterns of English orthography.[9] In addition, a new spelling series utilizing this information is now being developed and is scheduled for publication (the first levels) in February of 1969.[10] With these data, further consideration has also been given to some of the problems in the design of the reading machine for the blind.

APPENDIX A

PRONUNCIATION CODING

Thorndike Symbol	IPA Symbol	Keypunch and tape coding	Internal and printer coding
a	æ	A	A
ā	e	A-	-
ã	ɛ	A$	$
ä	a	A=	•
b	b	B	B
ch	c	CH	3
d	d	D	D
e	ɛ	E	E
ē	i	E-	4
ėr	ɝ	E•R	5R
f	f	F	F
g	g	G	G
h	h	H	H
i	I	I	I
ī	aI	I-	6
j	j	J	J
k	k	K	K
l	l	L	L
m	m	M	M
n	n	N	N
ng	η	NG	NG
o	a	O	O
ō	o	O-	7
ô		O)	8
oi	I	OI	OI
ou	au	OU	OU
p	p	P	P
r	γ	R	R

Thorndike Symbol	IPA Symbol	Keypunch and tape coding	Internal and printer coding
s	s	S	S
sh	s	SH	9
t	t	T	T
th	θ	TH	2
FH	ð	(	(
u	ʌ	U	U
u̇	U	U·	Q
ū	ju	U-	=
ü	u	u=	≠
v	v	V	V
w	w	W	W
y	j	Y	Y
z	z	Z	Z
zh	z	ZH	l
ə	ə	*	*
primary stress	ˈ	/	/
null symbol			X

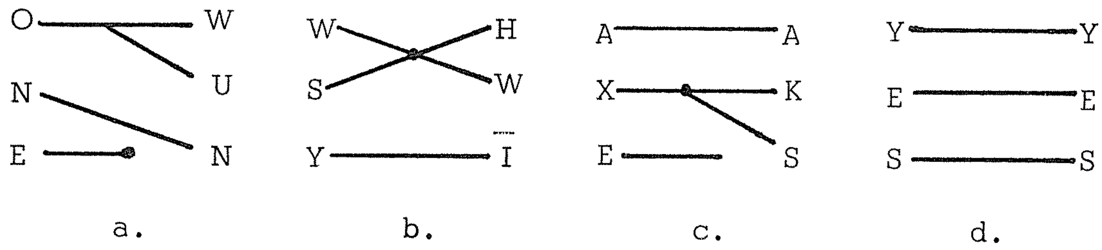


Figure 1

MAPPING OF THREE GRAPHEMES INTO THREE PHONEMES

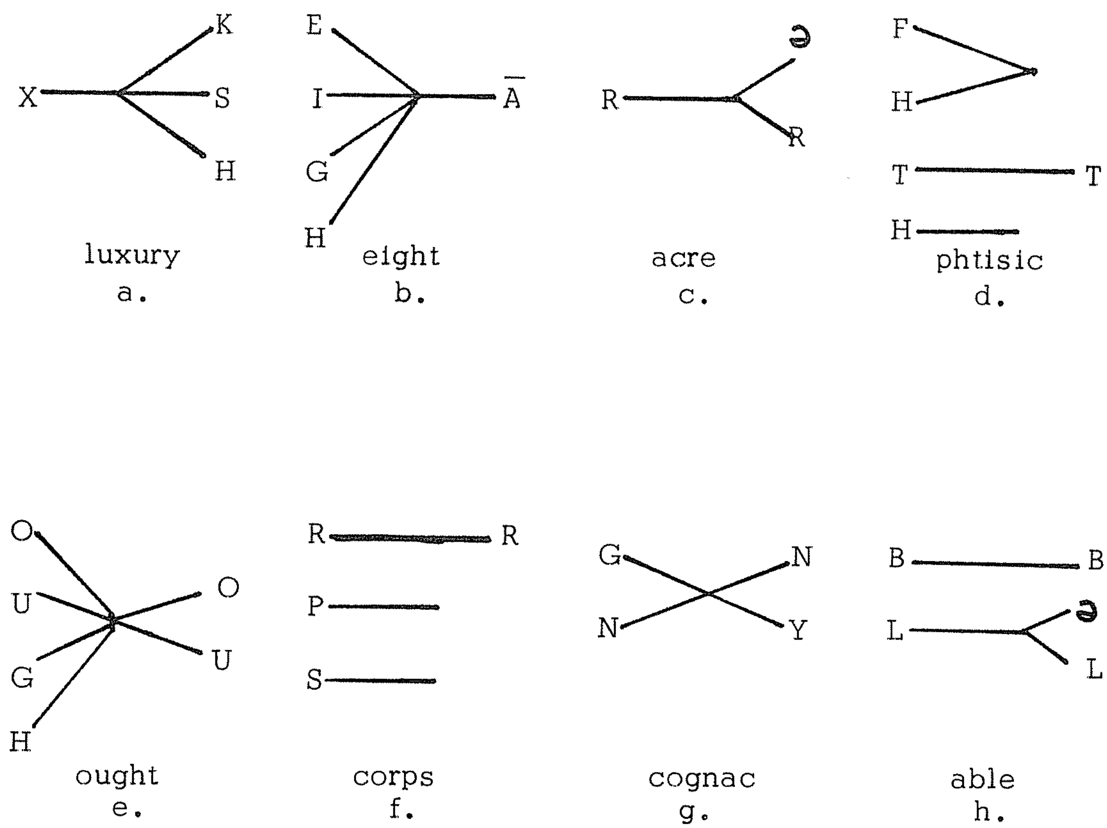
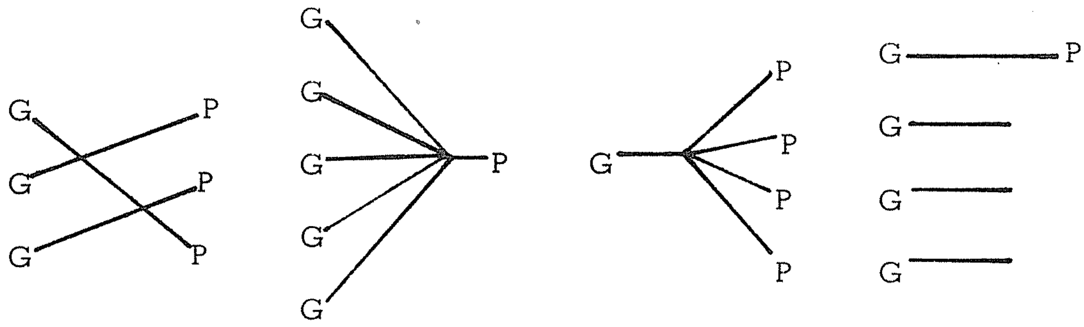


Figure 2.

IRREGULAR MAPPINGS



a. non-sequence preserving.

b. more than four graphemes mapped into one phoneme.

c. one grapheme mapped into more than three phonemes.

d. more than two silent graphemes in a sequence.

Figure 3

ILLEGAL MAPPINGS

1. Silent interconsonantal vowel.*

[SOLELY] ⟶ /SOLLI/

Without Match-Test	With Match-Test
[S] → /S/	[S] → /S/
[O] → /$\bar{O}$/	[O] → /$\bar{O}$/
[L] → /LL/	[L] → /L/
[E] → /I/	[E] → /X/
[L] → ?	[L] → /L/
[Y] → ?	[Y] → /I/

2. Silent initial consonant

[HEIR] ⟶ /AR/

Without Match-Test	With Match-Test
[H] → /$\tilde{A}$/	[H] → /X/
[EIR] → /R/	[EIR] → /$\widetilde{AR}$/

*See Appendix A for definition of symbols.

3. Vocalic Consonants

[TABLE]	/TAB*L/
<u>Without Match-Test</u>	<u>With Match-Test</u>
[T] → /T/	[T] → /T/
[A] → /$\bar{A}$/	[A] → /$\bar{A}$/
[BL] → /B/	[BL] → /B*L/
[E] → /*/	[E] → /X/
? → /L/	

Figure 4. IRREGULARITIES

FOOTNOTES

1. No major study on the origins and development of English orthography has been published. Brief sketches can be found in A. Campbell, Old English Grammar (Oxford: Clarendon Press, 1962), 12-29 and passim; G. L. Brook, A History of the English Language (New York, W. W. Norton & Company, Inc., 1958), Ch. 5; and J. E. Blomfield, "The Origins of Old English Orthography," unpublished dissertation, Somerville College, England. (Bodleian Library MS. B. Litt. d. 263).

2. This topic is discussed in my article "Automatic Spelling-to-Sound Conversion," in P. Garvin, ed., Computation in Linguistics: A Casebook (Bloomington, Indiana: Indiana University Press, 1966), 146-61.

3. Supported by a grant from the U. S. Office of Education.

4. C. F. Hockett, "Analysis of Graphic Monosyllables," in Harry Levin, et al. A Basic Research Program on Reading (Ithaca: Cornell University, 1963).

5. Most of the material which follows is derived from my M.A. Thesis, "A Computer Program for Deriving Seplling-to-Sound Correlations," (Cornell University, 1962). Published in part in Harry Levin, et al., A Basic Research Program on Reading. (Ithaca: Cornell University, 1963).

6. English graphemics is discussed by W. Nelson Francis, The Structure of American English (New York, The Ronald Press, 1958), ch. 0. Various linguistic approaches to grapheme theory are compared by John C. McLaughlin, A Graphemic-Phonemic Study of a Middle English Manuscript (The Hague: Mouton & Co., 1963), Ch. II.

7. E. L. Thorndike, Thorndike Century Senior Dictionary. New York: D. Appleton-Century, 1941.

8. 1604/1604A SORT Reference Manual. CDC Publication No. 60053500. (Minneapolis, Minn., 1964).

9. See, in particular, Richard L. V nezky, "English Orthography: It's Graphical Structure and its Relation to Sound," Reading Research Quarterly, II (Spring, 1967), 75-106.

10. R. W. Day and P. Lightbody, Patterns and Words: A Spelling Series, Science Research Associates, Chicago, Illinois (forthcoming.)