01 Jul 1986

A Computerized Audio-Visual Speech Model

Joaquin Alberto Vila Ruiz

Arlan R. Dekock
Missouri University of Science and Technology, adekock@mst.edu

Recommended Citation

Ruiz, Joaquin Alberto Vila and Dekock, Arlan R., "A Computerized Audio-Visual Speech Model" (1986). *Computer Science Technical Reports*. 77.
https://scholarsmine.mst.edu/comsci_techreports/77

A COMPUTERIZED AUDIO-VISUAL SPEECH MODEL

J. A. Vila Ruiz* and A. R. DeKock

CSc-86-4

Department of Computer Science
University of Missouri-Rolla
Rolla, Missouri 65401 (314) 341-4491

*This report is substantially the M.S. thesis of the first author, completed July 1986.

ABSTRACT

This thesis describes an expert system able to animate (graphically) and reproduce (acoustically) a text in any language which uses the Latin alphabet. A rule editor has been developed to create and modify the set of letter-to-sound rules of the target language. A parser has been implemented to apply the set of rules and translate text-to-speech. Each phoneme has a unique sound and thus requires its particular positioning of the vocal organs which are displayed in two different projections: a front view and a profile cross view of a human face in synchronization with the output sounds of the speech synthesizer.

ACKNOWLEDGEMENTS

The author wishes to acknowledge the moral support of his advisor Dr. Arlan Dekock and to express his sincere appreciation for the guidance and patience shown to him during his graduate studies at UMR.

For his assistance in this research, gratitude is expressed to Prof. Ralph Lee.

Appreciation is also extended to Dr. James Joiner and Prof. Wayne Little.

A special acknowledgement is expressed to Dr. Lon Pearson for his concerned efforts in the editorial process of this research and for his introduction to the field of linguistics which proved invaluable.

This research was made possible by the sponsorship of the Consejo Nacional de Ciencia y Tecnologia (C.O.N.A.C.Y.T) of Mexico.

TABLE OF CONTENTS

Page

LIST OF ILLUSTRATIONS

Figures Page

LIST OF TABLES

Tables Page

I. INTRODUCTION

A. PROBLEM DEFINITION

In order to state the problem and to understand the goal of this thesis it is useful to cite the definition of phonetics.

> Phonetics is the study of speech sounds, their physiological production and acoustic qualities. It deals with configurations of the vocal tract used to produce speech sounds, the resulting acoustic patterns and the manner of combining sounds so as to make syllables, words and sentences. On another level it compares the speech sounds of different languages. Phonetics, since it is part of the analysis of the phenomenon of human language and speech behavior, contributes to the fields of anthropology, linguistics, psychology, sociology and related fields. Furthermore it has several practical objects: (1) to help in learning to pronounce foreign languages; (2) to contribute to speech therapy; (3) to record previously unwritten languages and to develop writing systems for them; (4) to develop more efficient electrical and electronic speech transmission systems (speech compression). Encyclopedia Britannica [1966].

Without question one of the most critical difficulties in learning a foreign language is its correct pronunciation. It seems that the student gets so accustomed to using only the set of sounds required by his mother tongue, that when he deals with the pronunciation of new sounds demanded by any other language, he has serious trouble trying first of all to hear and discriminate these sounds and secondly, and--perhaps the most critical problem--to pronounce such sounds.

The research described in this paper has produced a teaching aid for proper pronunciation. The system is developed around a microcomputer which depicts the processes of speech, animating (graphically) and reproducing (acoustically) a text in any language which uses the Latin alphabet.

The reason the Latin alphabet has been selected is because it is the most widely used alphabet system in the Western World. This alphabet is currently used for English, French, Italian, German, Spanish, Portuguese, Turkish, Polish, Dutch, Czech, Croatian, Welsh, Finnish, Hungarian, and other western languages. Besides there are new adaptations of the Latin alphabet to other languages in the Americas, Africa, Asia, Polynesia, etc. Furthermore, and, perhaps the most important reason, the ASCII characters deal with the Latin alphabet.

Due to the variety of natural written and spoken languages and the complexity that each of them offer when studying them, this thesis has used Spanish as a model and target for almost all the examples that serve for purposes of illustration.

B. PROBLEM SIGNIFICANCE

Spanish is the mother tongue of more than 300 million people all over the world. It is the official language of Spain and 19 countries in the Americas and the Caribbean. Spanish is the second most important language in the United States and is spoken by 20 million people here. In Africa Spanish is heard in Morocco and in Equatorial Guinea. Furthermore, Spanish is the mother tongue of several hundred thousand Sephardic Jews, whose ancestors were exiled from Spain in 1492. At present Sephardic Jews live in Holland, Greece, the Balkans, Bulgaria, Yugoslavia, Turkey, Israel, Egypt, Lebanon, Syria, the United States and Latin America.

Spanish is often used in the Philippines, and several other Caribbean islands such as Trinidad. giving rise to a minimum of 36 countries where Spanish is spoken. Among the languages of the world Spanish is fourth in the number of native speakers, after Chinese, English and Indian. Of the principal Romance languages, Spanish is the most common. (For further information see Resnick [1981].) Furthermore, any other language has its relevance and uniqueness, as pointed out by Bassnet-McGuire [1980] in a reference to Edward Sapir:

> Edward Sapir claims that 'language is a guide to social reality' and that human beings are at the mercy of the language that has become the medium of expression for their society. Experience, he asserts, is largely determined by the language habits of the community, and each separate structure represents a separate reality:
>
>> No two languages are ever sufficiently similar to be considered as representing the same social reality. The worlds in which different societies live are distinct worlds, not merely the same world with different labels attached.

As firmly as Sapir, Martin Heidegger (translated by Steiner [1975]) declares that:

> Man acts as if he were the shaper and master of language, while it is language which remains mistress of man. When this relation of dominance is inverted, man succumbs to strange contrivances. Language then becomes a means of expression. Where it is expression, language can degenerate to mere impression (to mere print). Even where the use of language is no more than this, it is good that one should still be careful in one's speech. But this alone can extricate us from the reversal, from the confusion of the true relation of dominance as between language and man. For in fact it is language that speaks. Man begins speaking and man only speaks to the extent that he responds to, that he corresponds with language, and only in so far as he hears language addressing, concurring with him. Language is the highest and everywhere the foremost of those assents which we human beings can never articulate solely out of our own means.

The relevance of language as a medium of communication is irrefutable, but there are serious problems in international communication as clearly pointed out by Seleskovitch [1978]:

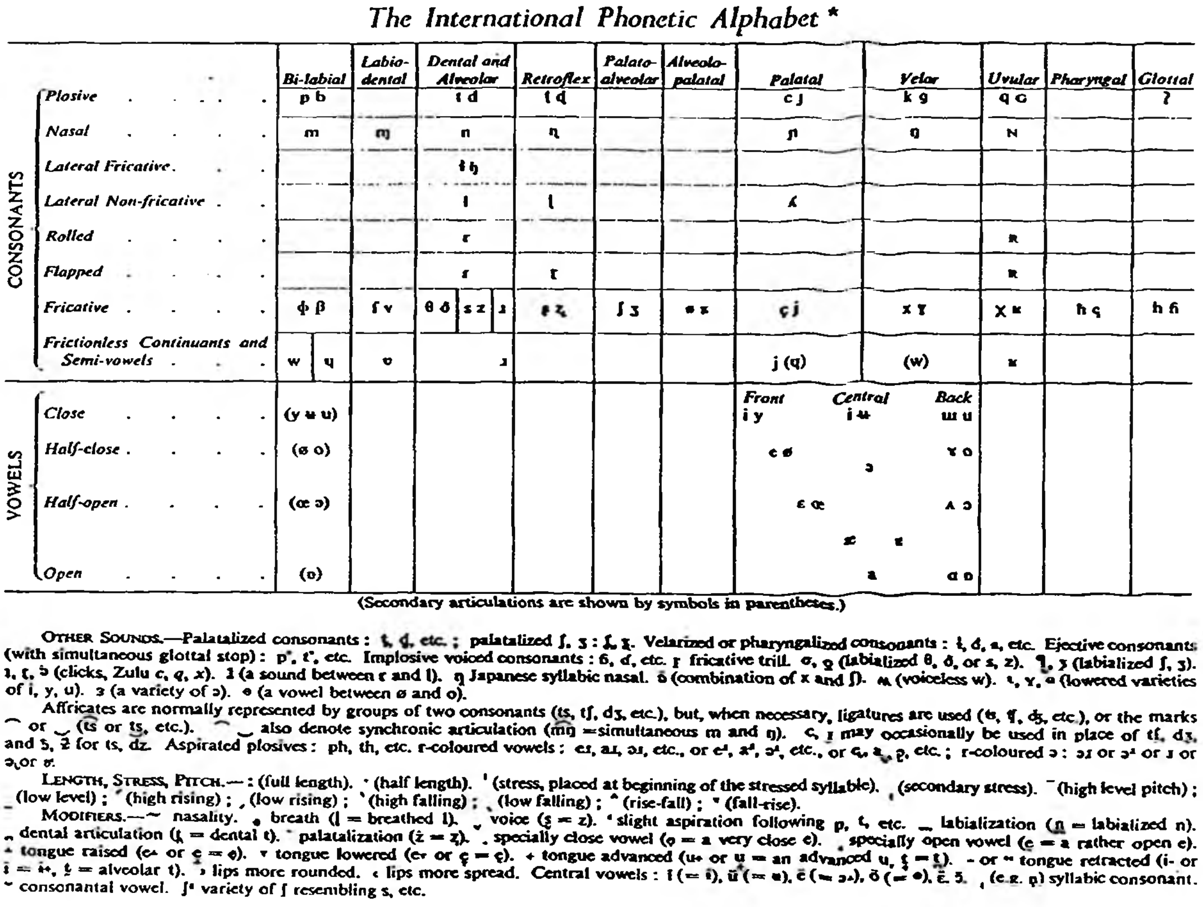
> Although science and technology have succeeded in eliminating most of the barriers which have separated men from each other, one barrier remains which runs counter to this trend, and which constitutes a hurdle that has proved well nigh insurmountable despite all the forces working to enhance human contact. This is the language barrier, the original curse, the Biblical tower of Babel.

In the verbal communication process, many events happen at once. Among them, we will focus on only one, the pronunciation of the language. It is indeed simple to grasp the importance of pronouncing a language correctly. If you do not speak clearly, you are not understood.

There are countless reasons for people to continue learning foreign languages. This fact need not be further discussed, but it is fundamental when considering the significance of this thesis.

C. PROBLEM SOLUTION

Almost all the phoneticians and teachers of foreign languages approach the teaching of phonemes (the basic sound unit of speech) by using a profile cross-section of the voice organs (or some times a front view of the the human face). By this dissection of the mouth they are able to depict the point of articulation and the manner of articulation of the speech organs. Most of the phonemes that human languages produce are classified in the international phonetic alphabet. See figure 1. This alphabet was designed to produce phonetic writing of all languages.



The International Phonetic Alphabet *

		Bi-labial	Labio-dental	Dental and Alveolar	Retroflex	Palato-alveolar	Alveolo-palatal	Palatal	Velar	Uvular	Pharyngal	Glottal
CONSONANTS	Plosive	p b		t d	ʈ ɖ			c ɟ	k ɡ	q ɢ		ʔ
	Nasal	m	ɱ	n	ɳ			ɲ	ŋ	ɴ		
	Lateral Fricative			ɬ ɮ								
	Lateral Non-fricative			l	ɭ			ʎ				
	Rolled			r						ʀ		
	Flapped			ɾ	ɽ					ʀ		
	Fricative	ɸ β	f v	θ ð s z ɹ	ʂ ʐ	ʃ ʒ	ɕ ʑ	ç ʝ	x ɣ	χ ʁ	ħ ʕ	h ɦ
	Frictionless Continuants and Semi-vowels	w ɥ	ʋ	ɹ				j (ɥ)	(w)	ʁ		
VOWELS	Close	(y ʉ u)						Front Central i y ɨ ʉ	Back ɯ u			
	Half-close	(ø o)						e ø	ɤ o			
									ə			
	Half-open	(œ ɔ)						ɛ œ	ʌ ɔ			
								æ	ɐ			
	Open	(ɒ)							a ɑ ɒ			

(Secondary articulations are shown by symbols in parentheses.)

OTHER SOUNDS.—Palatalized consonants : ƫ, ȡ, etc. ; palatalized ʃ, ʒ : ʆ, ʓ. Velarized or pharyngalized consonants : ɫ, ᵭ, ᵴ, etc. Ejective consonants (with simultaneous glottal stop) : pʼ, tʼ, etc. Implosive voiced consonants : ɓ, ɗ, etc. ɼ fricative trill. σ, ƍ (labialized θ, ð, or s, z). ʮ, ʯ (labialized ʃ, ʒ). ʇ, ʗ, ʖ (clicks, Zulu c, q, x). ɺ (a sound between ɾ and l). ɴ Japanese syllabic nasal. ɧ (combination of x and ʃ). ʍ (voiceless w). ɪ, ʏ, ᴜ (lowered varieties of i, y, u). ɜ (a variety of ə). ɵ (a vowel between ø and o).

Affricates are normally represented by groups of two consonants (ts, tʃ, dʒ, etc.), but, when necessary, ligatures are used (ʦ, ʧ, ʤ, etc.), or the marks ͡ or ‿ (t͡s or t‿s, etc.). ͡ ‿ also denote synchronic articulation (m͡ŋ =simultaneous m and ŋ). c, ɟ may occasionally be used in place of tʃ, dʒ, and ƽ, ƻ for ts, dz. Aspirated plosives : ph, th, etc. r-coloured vowels : eɹ, aɹ, ɔɹ, etc., or eʴ, aʴ, ɔʴ, etc., or ᶒ, ᶏ, ᶗ, etc. ; r-coloured ə : əɹ or əʴ or ɹ or ə˞ or ɚ.

LENGTH, STRESS, PITCH.—ː (full length). ˑ (half length). ˈ (stress, placed at beginning of the stressed syllable). ˌ (secondary stress). ˉ (high level pitch) ; ˍ (low level) ; ˊ (high rising) ; ˏ (low rising) ; ˋ (high falling) ; ˎ (low falling) ; ˆ (rise-fall) ; ˇ (fall-rise).

MODIFIERS.—˜ nasality. ̥ breath (l̥ = breathed l). ̬ voice (s̬ = z). ʻ slight aspiration following p, t, etc. ̫ labialization (n̫ = labialized n). ̪ dental articulation (t̪ = dental t). ̇ palatalization (ż = ʓ). ̝ specially close vowel (e̝ = a very close e). ̞ specially open vowel (e̞ = a rather open e). ˔ tongue raised (e˔ or e̝ = e). ˕ tongue lowered (e˕ or e̞ = ɛ). ˖ tongue advanced (u˖ or u̟ = an advanced u, t̟ = ʈ). ˗ or ̠ tongue retracted (i˗ or i̠ = i˖, t̠ = alveolar t). ̹ lips more rounded. ̜ lips more spread. Central vowels : ï (= ɨ), ü (= ʉ), ë (= ə˔), ö (= ɵ), ɛ̈, ɔ̈. ̩ (e.g. n̩) syllabic consonant. ̯ consonantal vowel. ʃˢ variety of ʃ resembling s, etc.

Figure 1: The International Phonetic Alphabet

Before phonetic writing emerged, there were no connections between the depicted symbol (picture-writing and the pure ideographic scripts) and the spoken name for it. These hieroglyphic symbols can be read in many languages.

> Phonetic writing is a great step forward. Writing has become the graphic counterpart of speech. Each element in this system of writing correspond to a specific element, that is sound, in the language to be represented. The signs thus no longer represent objects or ideas, but sounds or groups of sounds; in short, the written forms become secondary forms of the spoken ones. A direct relationship has thus been established between the spoken language and the script, that is, writing has become a representation of speech. (Diringer [1968].)

Phonetic and linguistic researchers have assembled the pronunciation rules for most languages. However, new accents are heard everywhere. Apparently there is not a reliable set of rules, because modes and categories inherited from the past no longer seem to fit the reality experienced by a new generation. Furthermore, Navarro [1968] asserts that "the same phonemes appear in very different forms. In fact, there is no phoneme that is heard in the same way from all lips." Indeed, we deal with an evolving phenomenon, that of language itself.

The research on these pronunciation problems showed that a solution for teaching purposes could be achieved using an extremely flexible expert system able to update the pronunciation rules demanded by the language being dealt with, and using a parser (rule-applier), to actually apply the letter-to-sound rules to any input text, and eventually translate text-to-speech synchronized with the graphic animation images of two projections of the human face. (Both projections are the most common used in phonetics.) This system was designed to run on an IBM/PC (with graphics capabilities). Equipped with a Votalker IB, which embodies the Votrax SC-02 phoneme synthesizer. The Votalker IB incorporates 64 phonemes with the additional capability of producing allophones (variations upon phonemes).

II. LITERATURE REVIEW

A. HISTORICAL OVERVIEW

Phonetics became an independent field of study in the 19th century. However there are some traces of human speech awareness in old civilizations.

> The use of alphabetic or syllabic notation, as opposed to pure ideographs, to record human speech shows at least an intuitive knowledge of speech units and their relationships. One of the earliest attempts at a systematic description and classification of speech sound was that of the Sanskrit grammarians, whose work came to its climax with Panini (4th century B.C.). (Encyclopedia Britannica [1966].)

> Throughout history man's voice has been a source of interest and wonderment. Ancient man often took his ability of speech as a symbol of divine origin. Not unnaturally, he sometimes ascribed the same ability to his gods. Zealous priests, not insensitive to the opportunity, frequently tried to make their idols speak directly to the people. Talking statues, miraculous voices, and oracles were well known in the Greek and Roman civilizations. (Flanagan [1972].)

> From the classical era until the end of the 17th century, however, more emphasis was placed on the process of phonation rather than on an explicit analysis of speech units. Such descriptions can be found among the works of Hippocrates, Aristotle, Galen and later, Leonardo da Vinci. (Encyclopedia Britannica [1966].)

Close to the end of the 18th century interest was shown in speech synthesis. One of the first attempts at mechanical speech generation, occurred around 1770 at the Imperial Academy of St. Petersburg when Kratzenstein demonstrated the physiological speech differences between five vowels. Kratzenstein constructed a speaking device using acoustic resonators. In 1791 unknown to Kratzenstein, Wolfang von Kempelen was working in parallel on a more elaborate speaking machine for generating connected speech. Later, Sir Charles Wheatstone perfected the Kempelen machine. Following that little research was done until the turn of the 19th century.

> In the late 1800s, a young experimenter from Edinburgh, Scotland had the chance to view von Kempelen's machine as constructed by Wheatstone. The impression that this mechanical speaking device made on the young Alexander Graham Bell not only influenced the direction of his interest in the scientific field of speech, but also probably had a far reaching effect on the evolution of modern science. Some years later in February of 1876 he applied for and was granted U. S. Patent No. 174,465 describing the workings and practicality of a device called the telephone.
>
> Alexander Graham Bell, after having seen the Wheatstone speaking bellows, was prompted to construct, under his father's guidance, a model of his own. With the assistance of his brother Melville, Alexander created a physical working model of the human vocal tract from wood, rubber, and cotton. The cabinet or case chosen for his speaking machine was an actual mold of the human skull with the various vocal tract organs such as the lips, cheeks, tongue, and vocal cords replicated with the use of rubber and a soft batting. The movable parts of the simulated head were controlled by a variety of levers, and the vocal cords created by air passing through a slotted rubber membrane. Although the model may have looked somewhat grotesque. Mr. Bell claimed that the device could speak vowels, nasals, and even, as he gained more practice, simple connected phrases. As he played the talking mechanical head, he was reminded of his early days when he really began to understand the physiology and mechanics of speech. (Cater [1983] p. 74)

> During the 19th century several recording devices, such as the kymograph, became available for the quantitative measure of speech phenomena. These devices, along with the artificial palate and X-ray photography, gave impetus to the quantitative study of the physiology of speech... By 1881, when Eduard Sievers published his Grundzuge der phonetik, phonetics had become an independent discipline... another group of phoneticians became preoccupied with the classification of speech sounds and the creation of a notation system for recording speech in a written form. Leaders among this group were Paul Passy, Henry Sweet and Daniel Jones, all of them prominent in the sponsorship of the International Phonetic association which was inaugurated in France in 1886. The association soon drew up the International Phonetic Alphabet, which was designed to be applicable to all languages and which has become not only the most widely used phonetic alphabet but also the basis for many succeeding systems. (Encyclopedia Britannica [1966].)

In 1922, J. Q. Stewart conceived an electrical buzzer that could simulate the vocal cords. It also used a pair of inductive/capacitive resonators to create resonances of the throat and mouth. In 1937, R. R. Riesz exhibited his mechanical talker (reminiscent of a musical instrument) which was shaped like the human vocal tract. In 1939 and in 1940 an electrical-keyboard-operated speaking system named Voder was shown in the World's Fairs of New York and San Francisco. It was designed by Riez with the help of H. Dudley and S. A. Watkins. Voder was one of the first electrical speech synthesizers to produce connected speech. Its construction is very similar in principle to that of the phonetic speech synthesis systems in use today. These new systems have substituted computer output ports for the keyboard controls.

After World War II, the invention of the sound spectrograph permitted the analysis of speech sounds in great detail by showing sound waves of voice timbre. This progress in recording devices placed more emphasis on the study of speech sound and allowed more sophisticated methods of speech synthesis. Speech synthesis is a marriage between digital electronics and linguistics. In short, when one experiments with speech synthesis--including speech analysis--it forces the investigators to determine the operating rules and techniques for reproducing speech. Currently there are three fundamental approaches to electronic speech synthesis.

- The waveform encoder for direct speech reconstruction.
- The analog formant synthesizer for phonetic speech synthesis.
- Mathematical reconstruction of speech from the frequency domain such as linear predictive coded (LPC) synthesizers.

B. TEXT-TO-SPEECH SYSTEMS

With the modern understanding of speech and the current electronics technology, several text-to-speech systems have been developed. The simplest and earliest approaches to unrestricted text-to-speech translation used a small set of letter-to-sound rules, each specifying a pronunciation for one or more letters in some context. Later other approaches were created, embodying large pronunciation dictionaries or linguistic analysis; but they were not practical.

1. NRL System

One of the must successful text-to-speech systems by rule, was developed by the Naval Research Laboratory (NRL), and published in December, 1976 under the title "Letter-to-Sound Rules for Automatic Translation of English Text to Phonetics" (Elovitz [1976]). The NRL system demonstrated the practicality of routine text-to-speech translation. A set of 329 letter-to-sound rules was developed. Actually these rules translate English text into the international phonetic alphabet (IPA), producing correct pronunciations for approximately 90 percent of the words, or nearly 97 percent of the phonemes. A second set of rules translates IPA into the phonetic coding for a particular commercial speech synthesizer.

The rule structure developed by the NRL team is analogous to that used in this research. However, variations have been made in these investigations in order to generate a flexible rule syntax capable of including the requirements of diverse languages.

2. Text-to-Speech Development System

In 1983 Texas Instruments Inc. developed a powerful text-to-speech system with notable features (Fisher [1983]). However the source rule format of this system is a quasi-linguistic generalization of that used by the NRL system. From all its features the most significant contribution to this research was the introduction of the User Defined Symbols (UDS), which will be explained later.

3. Papers about Speech Synthesis

There are other useful articles about speech synthesis, including text-to-speech technology. The most significant in regard to this research are:

- Unlimited text-to-speech system: description and evaluation of a microprocessor based device (Bernstein [1980]).
- Text-to-speech conversion in Spanish. A complete rule-based synthesis system (Santos [1982]).
- Real time text to speech conversion system for Spanish (Olabe [1984]).
- The klattalk text-to-speech conversion system (Klatt [1982]).
- Bliss communication with speech or text output (Carlson [1982]).
- A multi-language text-to-speech module (Carlson [1982]).

C. ARTICULATORY SYNTHESIS

Another facet of innovative graphic devices is the branch of articulatory synthesis. The first articulatory synthesis system was proposed and developed by Coker and Fujimura at the end of the 1960s. A method for generating synthetic speech was devised by them in which synthesizer control signals are derived by rules from phonetic input data through intermediate-step vocal-tract area computations. Phonemes, the basic elements of the input data, are characterized as static context-independent ideal vocal-tract shapes. These are tabulated in the program as sets of parameters for the vocal tract model. This proposal was developed later on a Honeywell DDP-516. However, by current standards it seems to be an unrefined system.

D. LIP-READER TRAINER

A Lip-Reader Trainer system was written by Robin L. Hight of St. Louis. This software package converts typed input sentences into a corresponding sequence of lip, teeth and tongue positions on a graphics display (for an Apple II). This system only shows the positions of the lips in animation when a text is input to the system in phonetic form. That system was intended to aid deaf people. The contribution of lip-reader trainer to this research is the knowledge that there is only a small set of lip positions distinct enough to be read clearly and they provide sufficient information to the user to distinguish one phoneme from another. Hight pointed out that there are only 19 readable lip positions for English. However, in some languages, such as French and Spanish the set might be different, just because the positions of the lips in pronouncing some phonemes are different. For example in describing the vowel sound u in French it is important to see the lip rounding and mouth opening, a fact that is not critical in other languages.

III. A SHORT DESCRIPTION OF SPEECH

A. THE ORGANS OF SPEECH

In the process of speech, man utilizes various organs: the lungs supply compressed air; the vocal cords (larynx) supply voice vibration; the tongue, teeth, and lips have the role in speech of articulation (the process of forming meaningful sounds). In addition the cavities in which these organs are located act as resonators which modify the voice vibrations from the vocal folds. See Figure 2.

There are three principal resonance cavities: the mouth, the nose, and the throat. (The most significant is the mouth.) The roof of the mouth is formed by the hard palate and the soft palate. The hard palate is divided, in phonetic descriptions, into teeth, alveolar region, and palate proper. The soft palate is often called the velum.

B. THE SOUNDS

Human speech sounds are described scientifically using three criteria:

- Place or point of articulation (how the speech organs are placed or come in contact to produce the voice sound).

- Manner of articulation (how the sounds are produced, such as by friction, stopping or nasalizing the sound).

- Voiced or Unvoiced (whether or not the vocal cords vibrate).

1. Places of Articulation

In Figure 2, places of articulation are shown by numbered arrows:

According to Bolinger [1960] the places of articulation are:

1. Bilabial. Involving both lips, which are either pressed together or brought close together.
2. Labiodental. Involving contact with little pressure between the lower lip and the upper teeth.
3. Dental. Involves contact between the tongue tip and the upper teeth.
4. Alveolar. Involves the tongue, either the tip or the blade, touching or approaching the alveolar ridge, just back of the upper front teeth.
5. Palatal. Indicates either complete or partial contact between the tongue and the hard palate.
6. Velar. Involves contact or narrowing of the passage between the tongue and the soft palate or velum.
7. Nasal. Requires the opening of the nasal cavity, by lowering the velum.

2. Manner of Articulation

According to Bolinger [1960] The following terms describe the kinds of articulations found in Spanish (manner of articulation):

a. Stop [also called explosives or occlusives]. Refers to consonants stopping the flow of air.
b. Nasal. Refers to consonants which involve closure of the air passage somewhere in the mouth, but with voiced air admitted from the throat into the nasal cavity, resonating there and passing out through the nose.
c. Fricative. Refers to consonants made by narrowing the mouth passage at some point so that the air, voiced or voiceless, is forced out with an audible sound of hissing or friction.
d. Affricate. Refers to consonants involving momentary closure of the mouth passage at a given point, followed by a rather slow movement of release, during which a friction noise is produced.
e. Lateral. Refers to the kind of consonant made when the tongue touches only the median line of the palate, so that the air escapes around one or both sides of the tongue. [Formerly called liquids in English.]
f. Tap. Refers to the kind of consonant in which there is a single rapid contact made with the tip of the tongue against the alveolar ridge.
g. Vibrant. Refers to the kind of consonant in which the tip of the tongue vibrates against the alveolar ridge.

For further details see Navarro, Tomas [1967] p. 14-30.

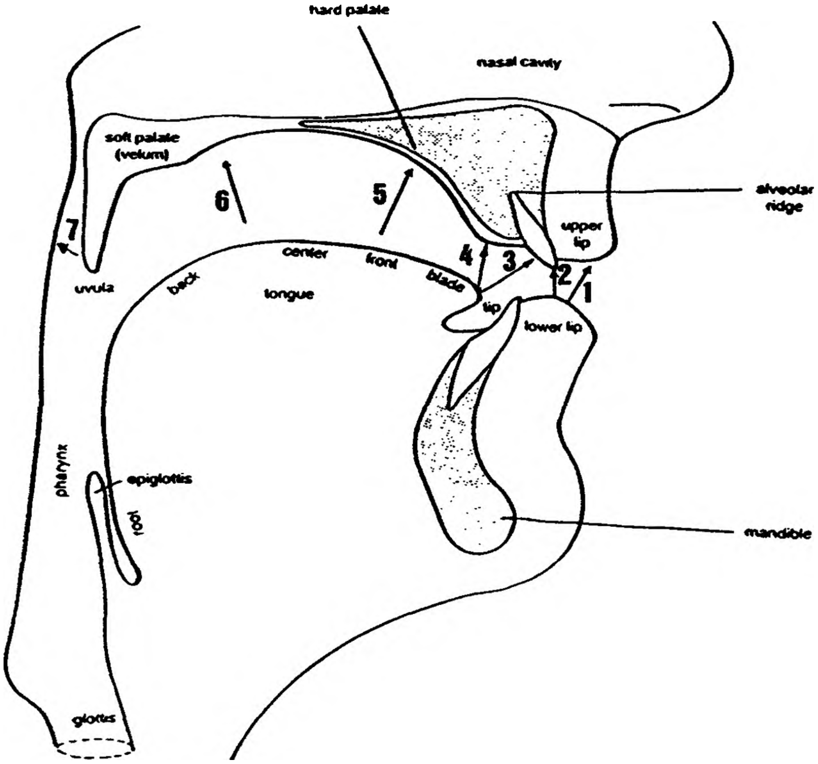


Figure 2: The Organs of Speech. (The Places of Articulation.)

3. VOICED OR UNVOICED

For every point of articulation the voiceless sound will generally have a voiced counterpart. Thus an [s], [p] and [t] are voiceless while [z], [b] and [d] are their voiced counterparts.

C. SPANISH PHONOLOGY

The number of sounds that the human being is able to emit using his speech organ is immense. However, each language has a unique pattern of sounds. Navarro Tomas asserts that:

> Some phonemes are of universal extent; others are found only in certain languages. Phonemes of a general character do not appear in the same proportion in all languages. The sound image of a language depends greatly on the proportion it uses the phonemes with [sic] and specially on the particular modality it follows within the number of variants that such units permit. In describing the oral shapes of the word, it is difficult to establish precise boundaries between sound and phoneme, between phonetics and phonology. At any rate, the general appearance of sounds, the effects produced by their combinations, and, specially, the role they play in relation to the meaning of words are all part of phonology. (Navarro, T. 1968, p. 14)

As in English, the sounds are classified in Spanish as vowels and consonants. This classification is owed to the affectation of the air when passing through the speech organs. When emitting a vowel the air passes over the vocal cords. The vibration of the vocal cords produces sounds that are modified in the cavities via the phonation organs (articulation). The difference is that when a vowel is produced there is no interruption or obstruction of the air. In the creation of consonants, however, the flowing of air is interrupted or obstructed by the vocal organs.

The Spanish official orthography, though more phonetic than other languages, is not even close to an appropriate representation of its pronunciation. The phonological series of Spanish consists of forty-two phonemes. The number of variants (allophones) that these phonemes assume in the pronunciation of all the countries where this language is spoken is incalculable.

D. FREQUENCY OF PHONEMES

Table I exhibits almost the complete set of phonemes of five different languages. Navarro Tomas clarifies the problem (Navarro, T. p. 17)...

> The clarity and sonority of phonemes can vary a great deal from one language to another, though their frequency may present similar proportions. The rate of frequency of phonemes is an indispensable norm for knowing the composition of each language, for comparing some languages with others, and for indicating the appropriate order in the teaching of pronunciation.

Furthermore, the knowledge of the frequency of the phonemes in each language is relevant to this specific research; it is important that one be careful in generating the rules of pronunciation of the phonemes with high frequency. Navarro Tomas asserts that the vowels a, e, o, and the consonant s represent 40% of the phonetic material used in any Spanish written text. A second category is that formed by n, r, l, d, t, i. A third category belongs to k(c,q), m, p, b, z, u, and g. And finally the phonemes with less frequency are: rr, f, j, ll, y, ñ, ch, and the diphthongs and triphthongs of the language. Table II reproduces the proportions established by Navarro Tomas (Navarro, T. p.25-26).

TABLE I

Phoneme Chart of five languages.

Symbol	Spanish	English	French	German	Italian
[a]...	sal	tall	pas	Gast	patria
[ɛ]...	en	pen	mere	Bett	pesca
[e]...		chaotic	thé	leben	era
[ə]...		about	ville	Erde	
[i]...	si	see	si	wie	si
[o]...	llorar		beau	wohl	dove
[ɔ]...	ojo	Scotch	fort	Sonne	cosa
[ø]...			peu	schön	
[œ]...			oeuf	zwölf	surf
[u]...	Tula	doom	tout	gut	subito
[y]...			lune	uber	
[ā]...			ange		
[ɛ̄]...			infame		
[ō]...			ombre		
[w]...	cuando	suave	oui		quando
[j]...	tibio	you	hier	Jahr	ieri
[b]...	bueno	bet	bien	Bett	bene
[β]...	hablar	love			
[ç]...	yo	hue		ich	
[d]...	duerme	doom	dormir	du	dormire
[ð]...	lado	this			
[θ]...	corazon	thunder			
[f]...	fuerte	fast	fort	fast	forte
[g]...	guerra	get	guérir	gerne	gamba
[gs]...		exile	exil		
[ɣ]...	luego				
[h]...		high	haine	hoch	
[k]...	calidad	cold	carte	kalt	caro
[ks]...	excavar	lax	luxe	Luchs	
[l]...	leche		long	lachen	lungo
[ʎ]...	llamar				figlia
[j]...	llamar	yonder	fille	Joch	
[m]...	amor	man	amour	Mann	amore
[n]...	andar	name	nez	Name	onda
[ɲ]...	senor	onion	agneau		agnus
[ŋ]...	cinco	sing		singen	angolo
[p]...	padre	pour	père	Pore	padre
[r]...	rojo	red	rouge	rot	rosso
[s]...	sangre	see	sang	rasten	sangue
[z]...	rasgar	zeal	zèle	Sohn	sdegno
[dz]...		adze			zero
[ʒ]...	alli	measure	jour	Gendarm	
[dʒ]...		jar			giubilo
[ʃ]...		show	chaud	schön	scioco
[t]...	todo	tan	tout	Tonne	tutto
[ts]...		Tsar		Zahn	zio
[tʃ]...	charro	church	caoutchouc	rutschen	cielo
[v]...	verde	vast	vert	warm	verde
[x]...	jabon	loch		Loch	

TABLE II

Frequency of Spanish Phonemes

	Vowels
a	13.00%
e	11.75
o	8.90
i	4.76
u	1.92
	40.33%

	Diphtongs
ie	0.86%
ia	0.54
ue	0.52
io	0.32
ua	0.20
ai	0.15
ei	0.15
oi	0.15
au	0.09
eu	0.05
iu	0.05
ui	0.05
uo	0.03
ou	0.00
	3.16%

	Voiced consonants
n	6.94%
r	5.91
l	5.46
d	5.00
m	3.09
b	2.54
g	1.04
s	1.00
rr	0.80
ll	0.60
y	0.40
ñ	0.36
	33.14%

	voiceless consonants
s	7.50%
t	4.82
k	4.23
p	3.06
z	2.23
f	0.72
j	0.51
ch	0.30
	23.37%

IV. DESIGN AND IMPLEMENTATION

Fluent and accurate pronunciation is not just an appropriate social grace; rather it is essential for true linguistic accomplishment and adequate communication. The desire to allow others to see the need for correct pronunciation was the principal motivation for this research. The hypothesis is: If the speech process is animated (via microcomputer), the students' appreciation and pronunciation of a language will improve through his understanding of the speech system.

The system created as a result of this research is named Babel. Babel is divided into two subsystems, the rule-editor and the parser (rule-applier).

A. RULE-EDITOR

The Rule-editor is the core of the system. With it, letter-to-sound rules can be developed to translate text to speech. The Rule-editor was provided with a very friendly interface to create and update the pronunciation rules of different languages.

The Rule-editor is mainly composed of four kinds of windows. At the left side of the screen is the WORKING RULE AREA where the rules are defined; in the middle is the MENU AREA where the main menu and edit menu are displayed; at the right is the INFORMATION AREA where the User Defined Symbols are exhibited, the phoneme chart, and the character chart; and finally at the bottom is the INPUT/OUTPUT AREA where information relevant to the knowledge to load or save is supplied as well as all the operations involving User Defined Symbols and other utilities. Figure 3 and Figure 4 show two different states of the Rule-editor where all the windows can be recognized.

1. Rule Syntax

The rule formalism of this system is very similar to that of the NRL system. However variations were made in order to increase the rule's possibilities.

Each rule has the form:

A[B]C=D

The character string B (body rule), occurring with left context A (prefix rule) and right context C (suffix rule), induces the pronunciation D (rule consequence or value).

- D is one or more phonemes or in other words is one or more of the 64 Votrax input symbols. See Table III. Each of these phonemes can be altered through the rule editor to produce allophones adjusting the speech parameters provided by the Votrax SC-02 synthesizer: duration, inflection, slope, pitch extension and filter frequency.

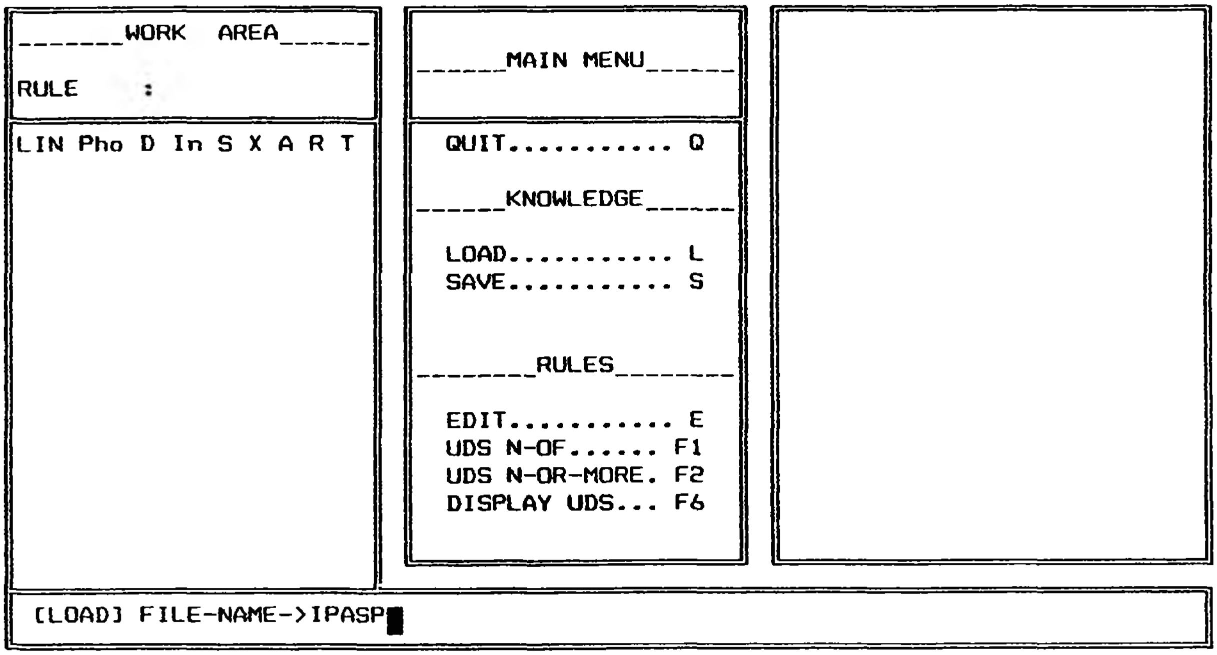
- B is the character or character string to be translated. In this case all the Spanish letters with all the special characters, accents and exceptions. Figure 5 and Figure 6 display the accents and special letters (used in non-English languages) availables in Babel system, and how to invoke them.

- A,C are the characters, string or special symbols (UDS user defined symbols) representing a class of character strings which denotes categories of sound such as vowels, voiced consonants, etc.

- Blanks are significant as they denote beginnings and ends of words.

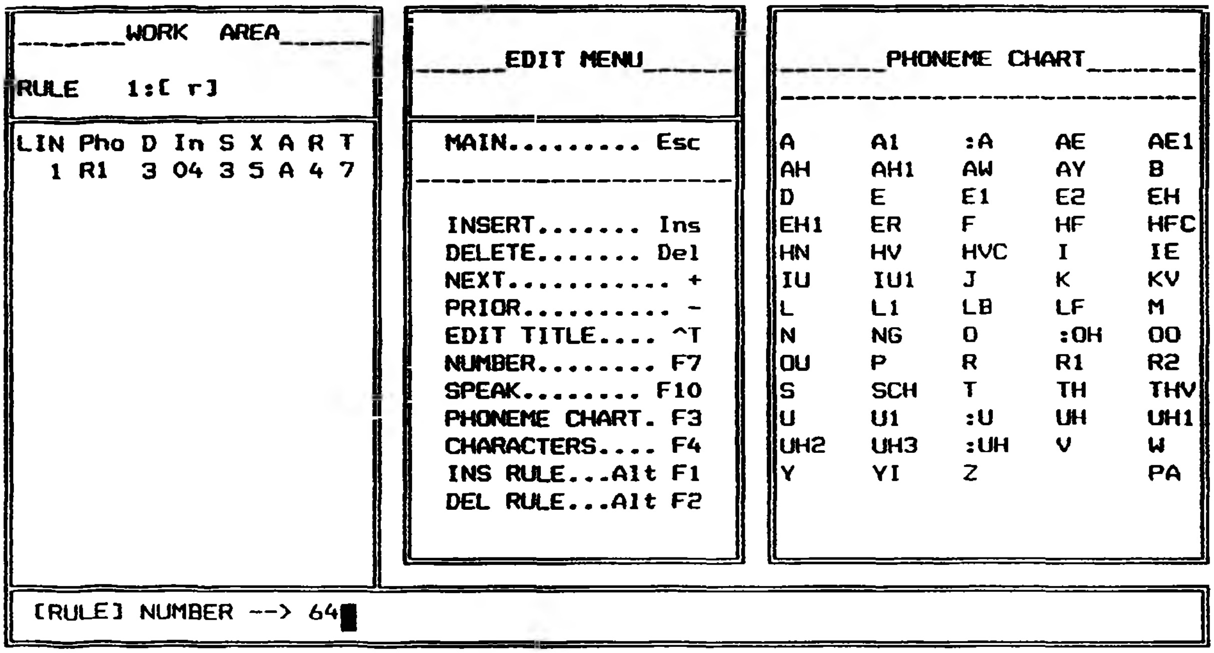
- Rule-order is extremely important

- The absence of A or C in a rule means that the corresponding context is irrelevant.



```
[LOAD] FILE-NAME->IPASP
```

Figure 3: Rule-Editor (Main Menu)



```
_______WORK  AREA______      _____EDIT MENU_____        _______PHONEME CHART_______

RULE   1:[ r]

LIN Pho D In S X A R T        MAIN......... Esc         A     A1    :A    AE    AE1
  1 R1  3 04 3 5 A 4 7                                  AH    AH1   AW    AY    B
                                                        D     E     E1    E2    EH
                              INSERT....... Ins         EH1   ER    F     HF    HFC
                              DELETE....... Del         HN    HV    HVC   I     IE
                              NEXT.......... +          IU    IU1   J     K     KV
                              PRIOR......... -          L     L1    LB    LF    M
                              EDIT TITLE.... ^T         N     NG    O     :OH   OO
                              NUMBER........ F7         OU    P     R     R1    R2
                              SPEAK........ F10         S     SCH   T     TH    THV
                              PHONEME CHART. F3         U     U1    :U    UH    UH1
                              CHARACTERS.... F4         UH2   UH3   :UH   V     W
                              INS RULE...Alt F1         Y     YI    Z           PA
                              DEL RULE...Alt F2

 [RULE] NUMBER --> 64
```

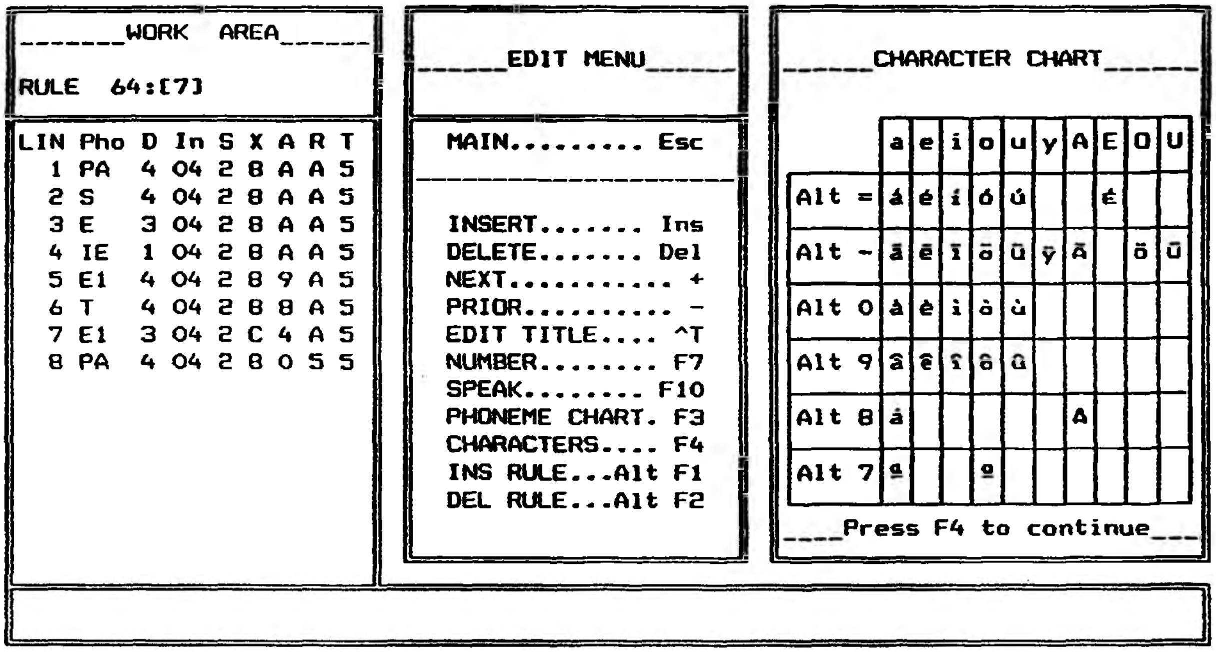
Figure 4: Rule-Editor (Edit Menu)

TABLE III

Votrax phonemes

Symbol	Votrax	Example	Symbol	Votrax	Example
[ ]	PA	(pause)	[l]	L	lady
[i]	E	keep,eat	[l]	L1	Louvre
[ɪ]	E1	become	[l]	LF	call
[ë]	Y	marry	[w]	W	want, why
[ ]	YI	year	[b]	B	big
[ɑ̈]	AY	made	[d]	D	said
[ɩ]	IE	ear	[g]	KV(HVC)	give
[ ]	I	mit	[p]	P	part
[e]	A	made	[t]	T	taste
[e]	A1	attainment	[k]	K	kite
[ɛ]	EH	said	[*]	HV	(voiced)
[ɛ]	EH1	enter	[g]	HVC	(g)
[œ]	AE	can	[h]	HF	hand
[œ]	AE1	happy	[*]	HFCT(k)	
[ɑ]	AH	pop	[*]	HN	(m,n,ng)
[ɑ]	AH1	honest	[z]	Z	zip, pays
[ɔ]	AW	lost	[s]	S	sing, city
[o]	O	for	[ʒ]	J	measure
[o]	OU	told	[ʃ]	SCH	ship
[ ]	OO	look	[v]	V	vault
[ ]	IU	you	[f]	F	fat, phone
[ ]	IU1	should	[ð]	THV	the, lathe
[u]	U	you	[θ]	TH	thing, with
[ ]	U1	unit	[m]	M	man
[ə]	UH	under	[n]	N	name
[ə]	UH1	common	[ŋ]	NG	long, finger
[ə]	UH2	constant	[*]	:A	Marchen
[ə]	UH3	what	[*]	:OH	Lowe
[ ]	ER	word	[*]	:U	fun
[r]	R	ring	[*]	:UH	bluhen
[r]	R1	error	[*]	E2	bitte
[r]	R2	Mutter	[*]	LB	bluhen

* Unassigned



```
_______WORK  AREA______

RULE  64:[7]

LIN Pho  D  In S X A R T
  1 PA   4  04 2 8 A A 5
  2 S    4  04 2 8 A A 5
  3 E    3  04 2 8 A A 5
  4 IE   1  04 2 8 A A 5
  5 E1   4  04 2 8 9 A 5
  6 T    4  04 2 8 8 A 5
  7 E1   3  04 2 C 4 A 5
  8 PA   4  04 2 8 0 5 5
```

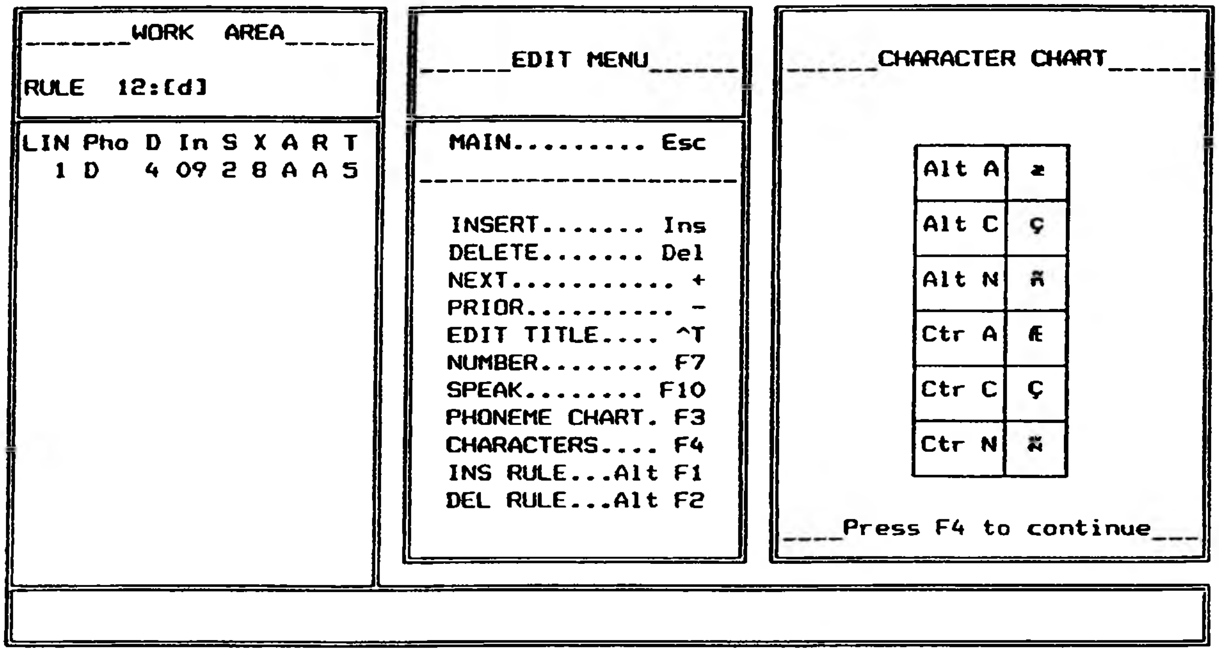
```
______EDIT MENU______

 MAIN......... Esc
____________________

 INSERT....... Ins
 DELETE....... Del
 NEXT........... +
 PRIOR.......... -
 EDIT TITLE.... ^T
 NUMBER........ F7
 SPEAK........ F10
 PHONEME CHART. F3
 CHARACTERS.... F4
 INS RULE...Alt F1
 DEL RULE...Alt F2
```

______CHARACTER CHART______

	a	e	i	o	u	y	A	E	O	U
Alt =	á	é	í	ó	ú			É		
Alt -	ā	ē	ī	ō	ū	ȳ	Ā		Ō	Ū
Alt 0	à	è	ì	ò	ù					
Alt 9	â	ê	î	ô	û					
Alt 8	å						Å			
Alt 7	ª			º						

____Press F4 to continue___

Figure 5: Special accents available in Babel



```
_______WORK  AREA______

RULE  12:[d]

LIN Pho D In S X A R T
  1 D   4 09 2 8 A A 5
```

```
______EDIT MENU______

 MAIN......... Esc
____________________

 INSERT....... Ins
 DELETE....... Del
 NEXT.......... +
 PRIOR......... -
 EDIT TITLE.... ^T
 NUMBER........ F7
 SPEAK........ F10
 PHONEME CHART. F3
 CHARACTERS.... F4
 INS RULE...Alt F1
 DEL RULE...Alt F2
```

```
______CHARACTER CHART______
```

Alt A	æ
Alt C	ç
Alt N	ñ
Ctr A	Æ
Ctr C	Ç
Ctr N	Ñ

```
____Press F4 to continue___
```

Figure 6: Special letters available in Babel

The main difference between Babel and the NRL rule composition is that in Babel the expert is welcome to define his own special symbols (UDS), unlike the NRL system where symbols are already defined and do not facilitate the creation of rules for other languages.

2. UDS (User Defined Symbol)

The UDSs are special defined symbols representing a class of character strings which denote categories of sounds such a vowels, consonants etc. The UDSs were introduced by Fisher [1983] in a text-to-speech development system. However, there are some variations in the process of defining a UDS in this system.

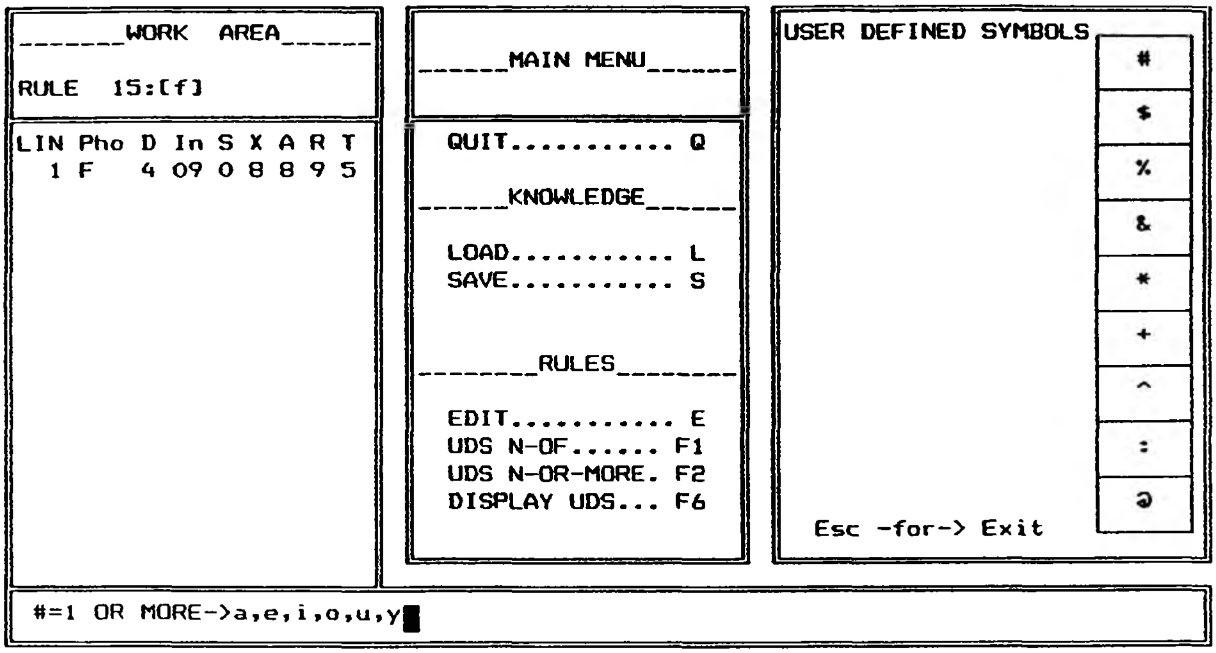
Babel supports two types of UDS's.

```
SYMBOL = __ OR-MORE = ( SET )
SYMBOL = __    OF   = ( SET )
```

Where SYMBOL is any of "#,$,%,&,*,+,¬,:,@" and (SET) is a list of character strings separated by commas. Examples of UDS are:

```
# = 1 OR-MORE = A,E,I,O,U,Y
: = 0 OR-MORE = B,C,D,F,G,H,J,K,L,M,N,O,P,Q,R,S,T,V,W,X,Z
* = 1 OF      = B,D,V,G,J,L,M,N,R,W,Z
```

In Figure 7 is displayed the main menu of the Rule-editor and the process of defining a UDS is shown.



```
_______WORK  AREA______

RULE  15:[f]

LIN Pho D In S X A R T
  1 F   4 09 0 8 8 9 5
```

```
______MAIN MENU______

QUIT........... Q

______KNOWLEDGE______

LOAD........... L
SAVE........... S

________RULES________

EDIT........... E
UDS N-OF...... F1
UDS N-OR-MORE. F2
DISPLAY UDS... F6
```

```
USER DEFINED SYMBOLS
#
$
%
&
*
+
^
:
@

  Esc -for-> Exit
```

```
 #=1 OR MORE->a,e,i,o,u,y
```

Figure 7: Rule-Editor (Edit Menu). The process of defining a UDS.

A representative rule for English using a UDS (according to the previous UDS's examples) is

#:[e]b̸

which means e at the end of a word preceded by # (one or more vowels) and : (zero or more consonants) is silent.

B. PARSER

The parser is the pragmatic side of Babel. This subsystem has been designed to animate a human speech model using the stimuli of the input text and screening it through a set of rules loaded (knowledge selected) in the parser.

The general block diagram exhibited in Figure 8 shows the process of the parser, which involves the following:

1. Knowledge Selection.

   - A welcoming display appears on the screen, requesting the user to name the knowledge to be loaded. See Figure 9. (The parser accepts any Knowledge created by the rule-editor.)

2. Load Rules.

   - The rules bearing the name of the knowledge selected are loaded into the system.

   - Next, two projections of the human face (front and profile cross view) are displayed on the screen. See Figure 10.

3. Input Text.

   - The user is free to type any text. (The input text is echoed at the bottom of the screen, in the input window.)

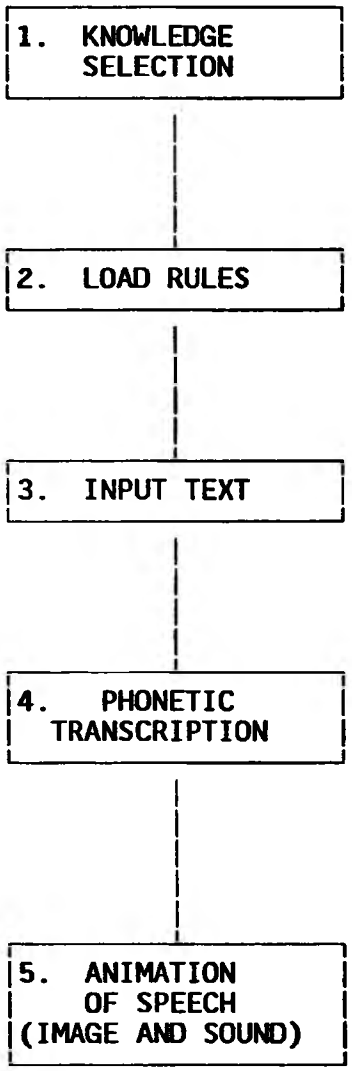

1. KNOWLEDGE SELECTION
2. LOAD RULES
3. INPUT TEXT
4. PHONETIC TRANSCRIPTION
5. ANIMATION OF SPEECH (IMAGE AND SOUND)

Figure 8: Parser process: general block diagram

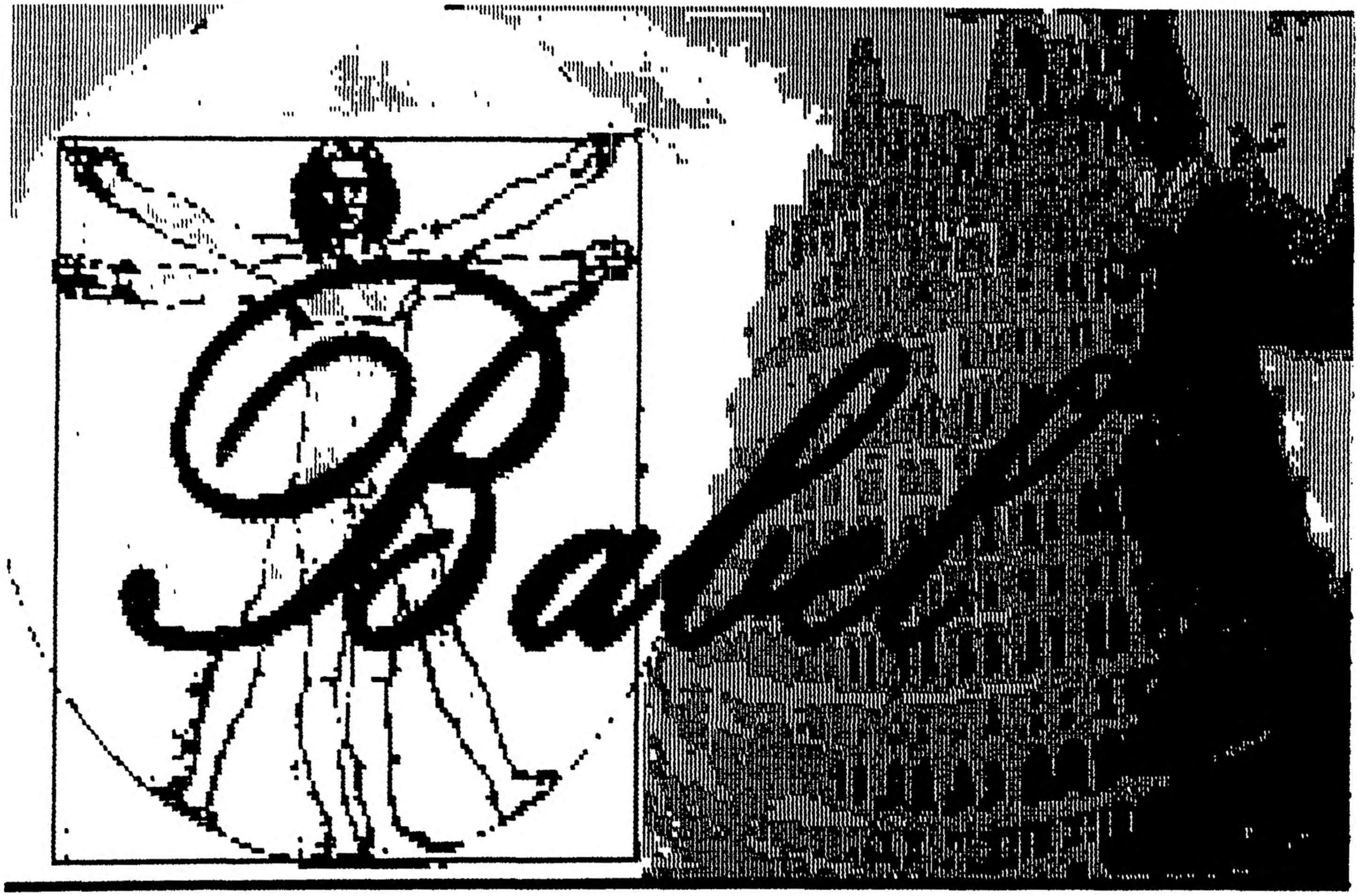


Figure 9: Parser's welcoming display screen

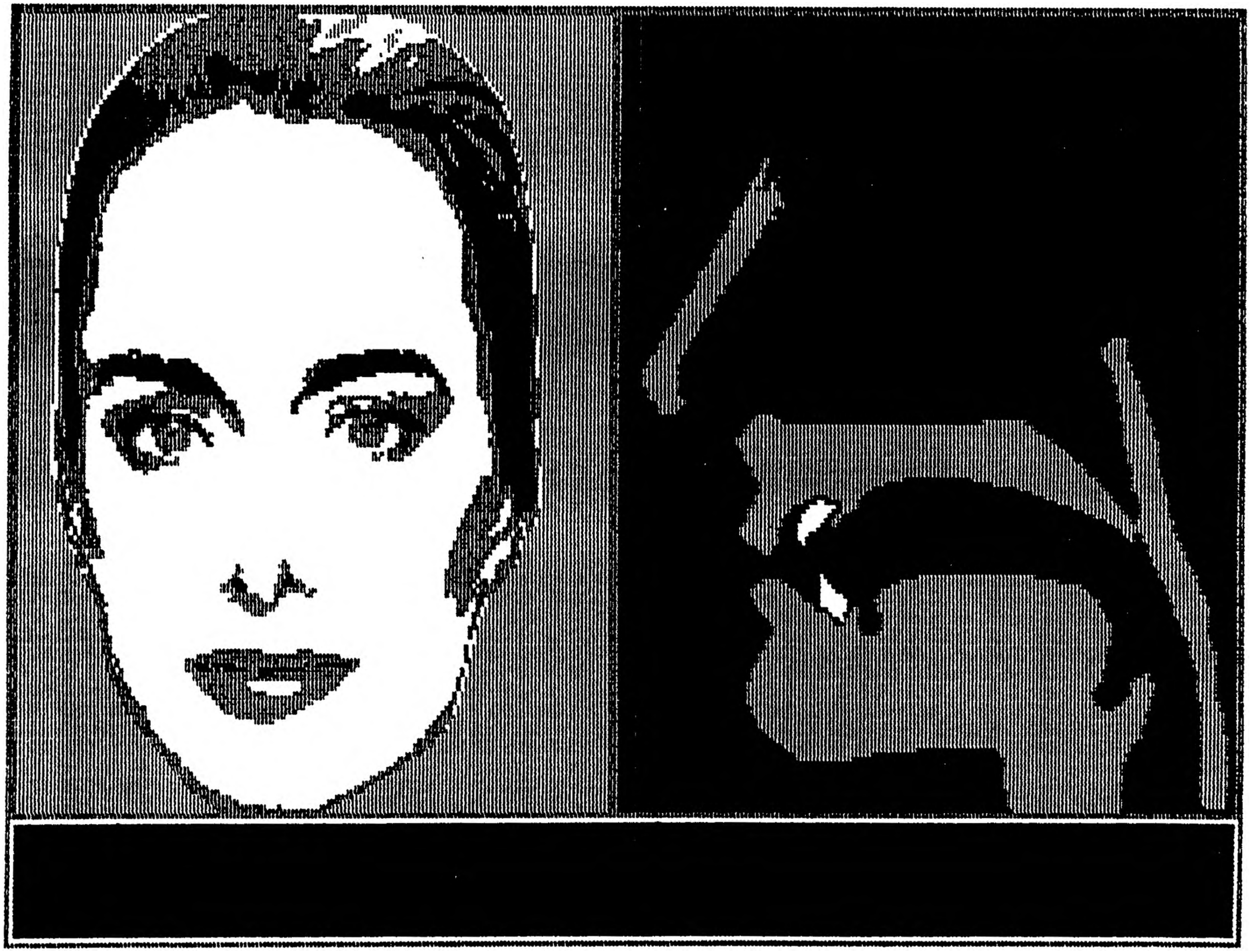


Figure 10: Parser's projection of the human face

4. Phonetic Transcription.

- The parser scans the text and produces a phonetic transcription of it.
  The phonetic transcription process is: "The process of transcribing a spoken word [text] into its phonetic components...". Votalker IB [1985]. (In other words, a set of phonemes that have to be reproduced in order to originate the speech of such a text.)

- The phonetic transcription process involves the following

  * The input text is scanned from left to right.

  * The subset of rules pertinent to the single character pointed to at any given time is scanned.

  * The parser that interprets and applies the rules ceases scanning when the first rule success (match) is encountered.

  * When a match is found the value of 'D' of the rule (the sequence of phonemes) is transmitted to a temporary buffer.

  * The last rule in the scanned subset is always the default pronunciation of 'B' (body rule or character string to be translated).

  * The pointer advances as many characters over the source text as the number of characters of 'B' (the body rule).

  * The scan process is over when all the characters of the source text are exhausted.

- Table IV shows how the phrase "le rogue, Enrique" is scanned.

5. Animation of Speech (Image and Sound).

- A succession of pictures showing the vocal speech organs of each phoneme generated by the phonetic transcription are exhibited on the screen at the same time that the sounds are uttered by the synthesizer.

  * Each phoneme requires a particular representation of the speech organs. Thus, Babel system has a specific image (of the vocal speech organs) for almost all the 64 Votrax phonemes. See table V, where the numbers appearing in the columns Front (mouth) and Profile (tongue) are related to Figure 11 and Figure 12 respectively.

TABLE IV

Phonetic Transcription of a phrase:

Knowledge : IPASP
Input text: le rogue, Enrique.

Rule structure

A[B]C	D

Pointer position	Rule used	Phonemes buffered
le rogue, Enrique.	[l]	L
le rogue, Enrique.	[e]	EH EH1
le rogue, Enrique.	[r]	R1
le rogue, Enrique.	[o]	O
le rogue, Enrique.	[gue]	KV HVC EH EH1
le rogue, Enrique.	[,]	PA
le rogue, Enrique.	[ ]	
le rogue, Enrique.	[e]	EH EH1
le rogue, Enrique.	[nr]	N R1
le rogue, Enrique.	[i]	E E
le rogue, Enrique.	[qu]	K
le rogue, Enrique.	[e]	EH EH1
le rogue, Enrique.	[.]	PA PA

TABLE V

Current configuration between Votrax phonemes and Vocal Tract Representations

Votrax	Front	Profile	Votrax	Front	Profile
PA	1	1	L	12	12
E	6	10	L1	12	12
E1	10	10	LF	12	12
Y	6	10	W	12	18
YI	6	6	B	3	13
AY	10	10	D	5	12
IE	6	10	KV(HVC)	8	8
I	10	10	P	3	13
A	10	10	T	5	12
A1	9	6	K	4	4
EH	10	10	HV	*	*
EH1	10	10	HVC	8	8
AE	4	2	HF	9	9
AE1	4	2	HFCT(k)	4	4
AH	2	2	HN	*	*
AH1	2	2	Z	17	18
AW	14	9	S	17	18
O	14	9	J	16	16
OU	14	9	SCH	16	16
OO	18	4	V	7	17
IU	18	4	F	7	17
IU1	18	4	THV	17	18
U	18	4	TH	17	18
U1	18	4	M	3	13
UH	2	2	N	5	12
UH1	2	2	NG	11	4
UH2	2	2	:A	*	*
UH3	2	2	:OH	*	*
ER	13	15	:U	*	*
R	13	15	:UH	*	*
R1	13	15	E2	*	*
R2	13	15	LB	*	*

* Unassigned

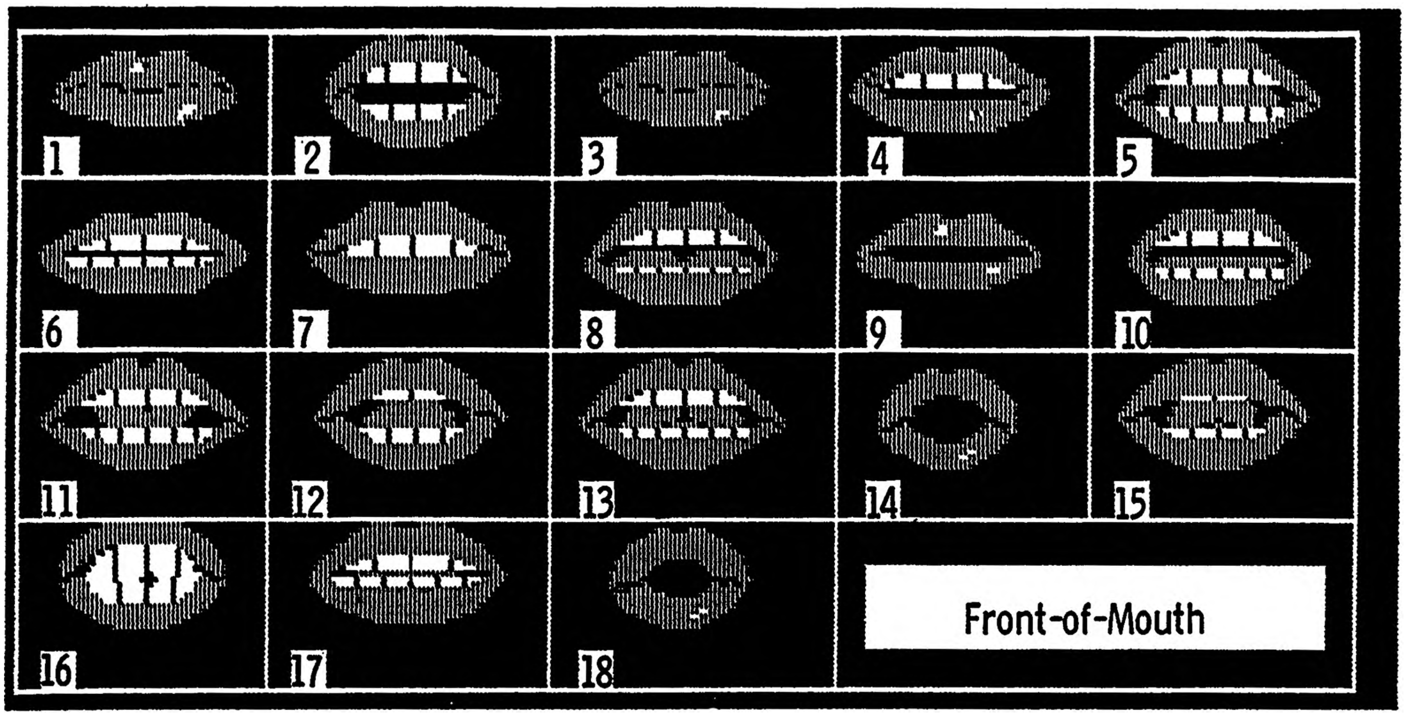


Figure 11: Set of Front-of-Mouth positions available in Babel system.

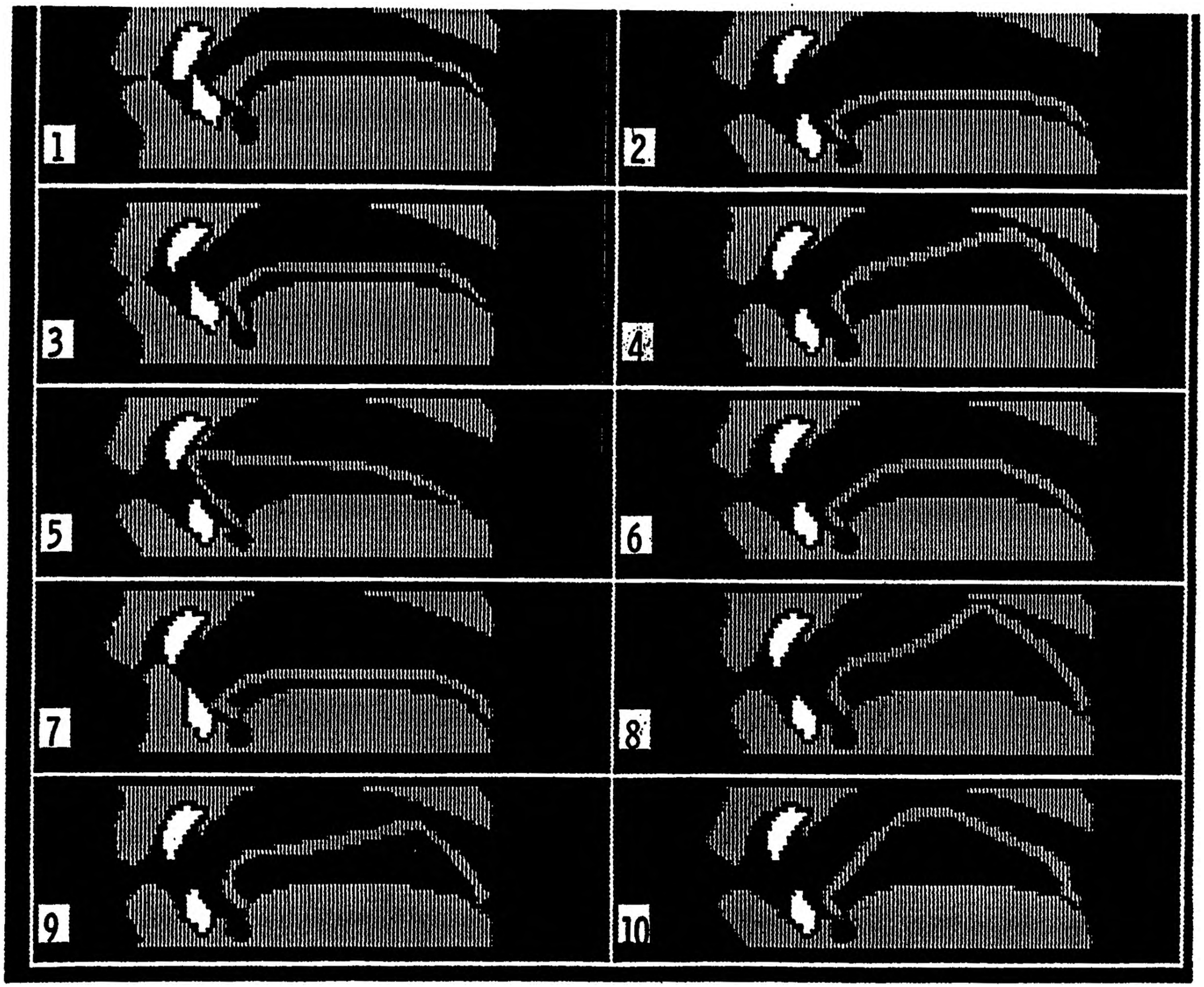


Figure 12: Set of Tongue Positions shown in Profile
available in Babel system.

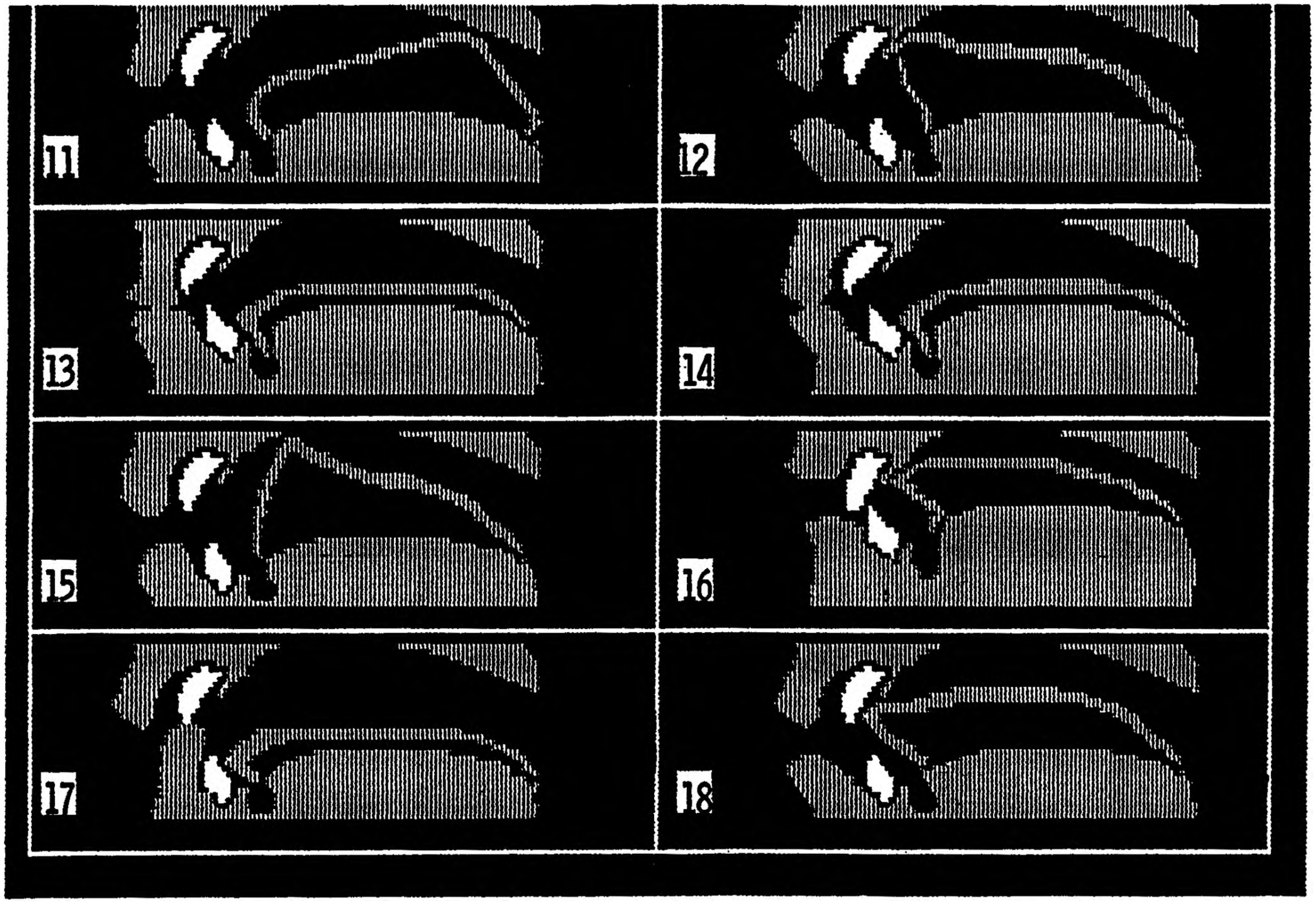


Figure 12: Set of Tongue Positions shown in Profile available in Babel system. (Continued.)

In summary this program translates text-to-speech by interpreting and applying the letter-to-sound rules (of the knowledge selected) over the input text. Once scanned, the system generates a smooth bass voice in conjunction with two projections of the human face (exhibiting the speech organs) which depict the desirable position of the organs of speech articulation to produce the phonemes determined by the text and generated by the synthesizer.

C. CREATING IMAGES

Two methods were used to create the images: Graphically drawing pixel by pixel on the screen using a utility program developed exclusively for this purpose; and digitizing images through an image digitizer with a Digital VAX 11/780 computer and a Gould DeAnza IP 8400 image processor. Most of the pictures were produced using the first method, that is, "drawing by hand". However, the front view faces displayed by the parser as well as the presentation window were digitized.

V. OPERATION AND EVALUATION.

There are two ways to interact with the system:

1. Through the Rule-editor to create and update pronunciation rules.

2. Through the parser to get speech animation of any input text.

A. RULE-EDITOR

The creation of the rules is the most important and delicate interaction with the system. In fact, the success of the parser rests entirely on accuracy of the rules. Therefore, it is necessary to invest considerable time with the Rule-editor before satisfactory performance can be achieved.

* Rule Development:

Spanish maintains a fairly good one to one relationship between letters and sounds. Taking advantage of this fact and following the words of Adelstein [1973] and Navarro [1967], the creation of the first draft of rules was feasible. Appendix A is a complete user's manual of the Rule-editor which explains the features of each window as well as how to create and update rules.

The creation of the rules was over as soon as the spoken output of the parser was understandable and pleasing. However, in several cases due to the limited set of phonemes provided by the synthesizer it was not possible to generate or improve the sounds of some phonemes. For example, the nasal voiced consonants n and ñ are currently causing problems in the pronunciation of some words. The phoneme n is provided by the synthesizer but the ñ is not. Moreover n is usually confused with the consonant l. The synthesizer pronounces both n and l as a voiced alveolar, but in human speech an l is a lateral and n is a nasal. It seems that the synthesizer failed to distinguish in its production between sounds with similar points of articulation, and especially in allowing the hearer to differentiate between nasal sounds and non-nasal sounds.

At present, there are a set of 68 letter-to-sound rules that translate Spanish text into speech. Appendix B illustrates these rules. IPASP is the name of the knowledge, where such rules are preserved. The current output of the system can be improved with more exhaustive rules. However, polishing the rules is a task that might take time yet one would be undoubtedly rewarded with more pleasing outputs.

Almost the complete set of letters used in Spanish are displayed in Appendix E (figure 14 through 21). This current configuration (between letters and images) could be changed if the user changes the rules or if the user alters the current settings of images and phonemes. (See in Appendix D the source text of the parser. Beginning on line 4000 one finds the procedure labeled "Assignation of mouths and tongues to phonemes" where the distribution mentioned in Table V is assigned. If one does not care for a certain pronunciation, then the user is welcome to make any change, selecting an alternate figure and compiling it again.)

B. PARSER

The parser is designed to animate a human speech model given a knowledge (set of rules) and any input text. The first and only query of the parser is the name of the knowledge to be used. Once the knowledge is loaded the user is welcome to type a limited size text that may include any character defined by the rules. The computer will repeat the speech animation as many times as the user keeps pressing any key but <return>. If <return> is pressed, the input window will be erased and the user may type again. When pressing the key F1 after the text has been input, the parser slows down the animation process in order to let the student appreciate in detail (that means phoneme by phoneme) the phonetic transcription of the input text. By toggling the key F1 again the parser returns to its normal animation speed.

* Parser Evaluation:

Several students were asked to sit in front of the microcomputer and work with the parser. They were allowed to type as much as they wanted. As soon as they became familiar with the system (particularly the voice) they were asked to listen carefully to the synthesizer output without any display of the text spoken. They were then requested to write down whatever they heard. Finally, they were asked to look at the screen and listen to a list of words pronounced correctly and incorrectly. These evaluations were made individually. The students were encouraged to express their comments.

This evaluations were done with two samples:

a. Spanish speakers.

b. Non-Spanish speakers.

VI. RESULTS

The results of the evaluation were obtained through comments of the students as well as observation of their reaction and attitude when interacting with the system. Below there is a list of comments and suggestions.

A. COMMENTS

- While interacting with the system, I could realize what was going on inside of my mouth. ...Suddenly I started to make and feel the position of my tongue and lips...
- It is a very friendly and flexible system. ...The fact that you can define and improve your rules so easily makes it worthy...
- If you create your own set of pronunciation rules, you experience a relabeling knowledge process and it is really refreshing.
- The system is very good in depicting the process of speech.
- The system is helpful for learning pronunciation of foreign languages. ...certainly you can make use of it in several languages...

B. SUGGESTIONS

- Change the voice of the synthesizer.
- Have a parameter to control the speed of the animation process.
- Improve the sound of the consonants n and ñ.
- Add stress or incorporate prosodic rules. (To enhance the quality, understability and naturality of the speech.)

VII. DISCUSSION

The hypothesis of this thesis proposes a system with a pragmatic goal: to help students properly pronounce any natural language which uses the Latin alphabet. In attempting to create a system to satisfy this ambitious goal, many beneficial experiences were gained which will be discussed next.

A. OBSERVATIONS

In the sample groups, enthusiasm was evident when individuals interacted with the system. This positive attitude contributed to an open dialogue and to a fruitful feedback. It was noticed that native Spanish speakers paid more attention to the stress of the words while non-Spanish speakers were more sensitive to the utterance of the phonemes. Both groups observed the lip movements while trying to follow the word patterns displayed by the computer. Anyhow, neither group spent enough time with the system to really see what happened during long periods of interaction. In developing a satisfactory pronunciation of a foreign language, the students must learn to hear and imitate sounds that are new for them. One of the hazards of this system is that in a self-learning course the students might get a synthetic pronunciation. Indeed, when using this system expert personal guidance is highly recommended.

B. CLARIFICATIONS

The Babel system sounds smooth because it does not have a prosodic analysis of the text and it was not the intention of this research to provide it with one. However, it is possible to generate an independent set of prosodic rules and add to the parser a second scan procedure to implement such an analysis. With a prosodic analysis the Babel system would improve its voice.

The voice of the synthesizer has a bass pitch while two of the faces supplied by the parser are females. This unavoidable conflict produced some expected suggestions. Nevertheless, the thoughtful comments in regard to the synthesizer were mainly concerned with the fidelity of some phonemes and the deficiency of others.

A control-window has been designed with options such as animation-speed, face-selection, knowledge-selection, text input-files, and set global synthesizer parameters (master rate, master amplitude, master inflection, master filter), but it has not been implemented in the parser. The importance of animation-speed is that the student could follow the phonemes in the vocal tract representations if they were slower. However, the technical problems entailed in this feature influence negatively the continuity of speech.

C. APPLICATIONS AND ENHANCEMENTS CONTINGENT ON MODIFICATION

The Babel system might be utilized in several practical applications:

* Phonetic course training: showing the student fundamental phonetic concepts. For example, the student could have an assortment of phonemes at his disposition and build words with them. The direct input of phonemes will allow the student to have practice discriminating sounds as well as gain familiarization with the vocal tract configuration needed to utter such a sounds. Furthermore, basic knowledge pertaining to pho-

netics might be added and accessed through help commands or direct questions to the system. "...rule-based systems can explain how and why they do things...." (Winston [1984]). The implementation of the classical "why" question in the Babel system undoubtedly would clarify the understanding of pronunciation.

* Speech pathology. Individuals trained in this area work with children who have difficulty pronouncing their mother tongue. Newest technology has children view X-rays of the organs, but this system does not allow for innovation and experimentation as the Babel computerized system does. Speech pathologists will find it to be an invaluable system for teaching purposes.

* File-text-reader: this application is quite obvious; it is just necessary to select the knowledge and the file-text to be spoken. The parser will do the rest. Handicapped individuals as well as others who would use it for other communication applications could take advantage of this system.

* Bilingual Transcription: "The place to start learning to make a Spanish sound is with the English sound that is closest to it. " (Bowen [1960]). That may be true for American students beginning to learn Spanish. However, this fact is perhaps applicable for any language. The bilingual transcription of a text seems to be supportive for beginning students. The application of a text transcription can be achieved through a program able to produce the phonetic output of any foreign language text using the writing system of the student's native language.

* Stages of articulations: Three stages can be observed in human articulation development: on-glide, tension, and off-glide. In the first stage (on-glide) the organs move from their resting state to the desirable position required by the sound to be emitted, in the second stage (tension) the organs keep the same position for a while; and eventually in the third stage (off-glide) the articulation organs return to their original resting position. The Babel system just utilizes the second stage. For more refined applications it might be desirable to include all three stages in the program. However, more speed, knowledge and memory would be required, and because each language varies the stress on which stage it emphasizes, it becomes very complicated.

D. SOLUTIONS

There is room for improvement and each suggestion for enhancement by evaluators will be considered. The inconvenience occasioned by the synthesizer will be remedied with the future technology.

VIII. CONCLUSION

Due to the large number of foreign language students, there is a perceptible advantage in having a flexible and widespread tool to assist them in areas such as language pronunciation, offering them standardization of knowledge via expert systems.

The phonetic transcription of a text produces its phonetic writing. In phonetic writing there is a more direct relationship between symbols and phonemes; and each phoneme has a specific vocal tract representation. Thereby, to achieve this effect it was possible to implement a computerized audio-visual speech model in an IBM/PC microcomputer equipped with a Votrax synthesizer for interactive translation (phonetization) of languages which use the Latin Alphabet from Text-to-Speech with animation of the vocal speech organs.

The fact that the students can see what they hear originates awareness of the speech process. Furthermore, if the student with the guidance of an expert learns to imitate properly the outputs of the Babel system he might really experience some unique learning.

The omission of a prosodic analysis might obscure some accomplishments of the Babel system. Nevertheless, the positive attitudes and comments gleaned from students did not reflect this.

The samples tested show great interest in the learning method. It is likely to have a good commercial value.

BIBLIOGRAPHY

Adelstein, Miriam. La Ensenanza del Espanol Como Idioma Extranjero: de la teoria a la practica. Madrid, Spain: Playor, S. A. 1973. p. 29-81.

Bassnett-McGuire, Susan. Translation Studies. New York: Methuen & Co., 1980, p. 13.

Bernstein, J., Pisoni, D. B. "Unlimited Text-to-Speech System: Description and Evaluation of a Microprocessor Based Device". IEEE-ICASSP, 1980 p. 576-579.

Bolinger, D. L., Bowen, J. D., Brady, A. M., Haden, E. F., Potson, L., Sacks, N. Modern Spanish a Project of the Modern Language Association. New York: Harcourt, Brace and company. 1960, p. 3-4.

Bowen, J. D., Stockwell, R. P. Patterns of Spanish Pronunciation a Drillbook. Chicago: The University of Chicago Press. 1960, p. 1.

Carlson, R., Granstrom, B., Hunnicutt, S. "A Multi-Language Text-to-Speech Module". IEEE-ICASSP, 1982 p. 1604-1607.

Carlson, R., Granstrom, B., Hunnicutt, S. "Bliss Communication with Speech or Text Output", IEEE-ICASSP, 1982 p. 747-750.

Cater, John P. Electronically Speaking: Computer Speech Generation. Indianapolis: Howard W. Sams & Co. 1983, p. 74.

Diringer, David. The Alphabet a key to the history of mankind. New York: Funk & Wagnalls, 1968, Volume I, p. 12.

Elovitz, H. S., Johnson, R., McHugh, A. and Shore, J. L. "Letter-to-Sound Rules for Automatic Translation of English Text to Phonetics", IEEE Transactions on Acoustic Speech and Signal Processing. December 1976 p. 446-459.

Encyclopedia Britanica. "Phonetics". Chicago: William Benton, 1966, Volume 17 p. 897-900.

Fisher, William M. "Text-to-Speech Development System", IEEE-ICASSP, 1983 p. 1344-1347.

Flanagan, James L. "Voices of Men and Machines" Speech Synthesis (Reprinted from JASA, 1972, p.1375). Pennsylvania: Dowden, Hutchinson & Ross, Inc. 1973, p. 9.

Klatt, Dennis H. "The Klattalk Text-to-Speech Conversion System", IEEE-ICASSP 1982 p.1589-1592.

Navarro, Tomas. Manual de la Pronunciacion Espanola. New York: Hafner Publishing Company, 1967 p. 13-145.

Navarro, Tomas. Studies in Spanish Phonology. Miami: University of Miami Press. 1968, p.14, 17, 25-26.

Olabe, J. C., Santos, A., Martinez, R., Munoz, E., Martinez, M., Quilis, A., and Bernstein, J. "Real Time Text to Speech Conversion System For Spanish", IEEE-ICASSP, 1984 p. 2.10.1-2.10.3

Resnick, Melvyn C. Introduccion a la historia de la lengua espanola. Washington, D.C. Georgetown University Press. 1981, p. 1.

Santos, J. M., Nombela, J. R. "Text-to-Speech Conversion in Spanish a Complete Rule-Based Synthesis System", IEEE-ICASSP, 1982 p. 1593-1596.

Seleskovitch, Danica. Interpreting for International Conferences. Washington, D. C. 1978, p. 1.

Steiner, George. After Babel. New York: Oxford University Press, 1975, p. xi.

Votalker IB Speech Synthesizer. (A Manual). Votrax, Inc. Artic Technologies: 1985, p. 4-3 to p. 4-14.

Winston, Patrick Henry. Artificial Intelligence. Massachusetts: Addison-Wesley Publishing Company. 1984, p. 164

VITA

Joaquin Alberto Vila Ruiz was born on November 21, 1959 in Tampico, Tamaulipas, Mexico. He received his primary and secondary education from Instituto Escuela Cervantes, Tampico, Tamaulipas. He received his Electronic Technical degree from Instituto Tecnologico Regional de Ciudad Madero, in Ciudad Madero, Tamaulipas. He received his Bachelor of Science degree from Instituto Tecnologico y de Estudios Superiores de Monterrey, Monterrey (ITESM), Nuevo Leon. Where he graduated with honors. He completed the study of English as a foreign language in the University of Houston, Houston, Texas.

He was Director of the Computer Center at ITESM, Unidad Tampico, Altamira, Tam. Mexico. He taught courses at the ITESM at the undergraduate level, Monterrey, N.L. and Altamira, Tam., Mexico. He was a consultant of information systems for Gremio Unido de Alijadores, Tampico, Tam., Mexico; Programmer & Analyst for Dime, S.A. Monterrey, N.L., Mexico; Project Analyst for BANAMEX, Mexico City, Mexico; System Analyst at Sociedad Espanola de Carburos, Barcelona, Spain.

He has been enrolled in the Graduate School of the University of Missouri-Rolla since August of 1984. He was sponsored by Consejo Nacional de Ciencia y Tecnologia (C.O.N.A.C.Y.T) of Mexico.

APPENDICES

APPENDIX A

RULE-EDITOR USER'S MANUAL

The Rule-editor allows you to create and update letter-to-sound rules. In order to run the Rule-editor or the Parser the Speech Operating System KERNEL must be loaded first (Votalker IB, [1985] p. 2-1)

> The Kernel is the heart of the Speech Operating System. It provides a software interface between the Votrax SC-02 Chip and application software for tasks like interrupt handling, phoneme buffering, data format expansion and lexicon management. To load the Speech Operating System KERNEL simply type...
>
> SONIX <return>

In most of the cases an autoexec.bat file will load the Speech Operating System KERNEL. Nonetheless, if KERNEL is not loaded and you attempt to run the Rule-editor or the Parser the following message will appear.

```
CANNOT LOCATE THE SONIX KERNEL!

 RETURN TO DOS AND LOAD SONIX
```

A. CREATE AND EDIT A KNOWLEDGE

1. Run EDITOR (Rule-editor) by typing: EDITOR <return>
2. The Main menu will appear. See Fig. 3. The Main menu presents a set of options to create and edit knowledge as well as to define and retrieve UDS (user defined symbols).

If you press:

- Q key (QUIT): The session is ended.

- L key (LOAD): A prompt will appear in the INPUT/OUTPUT AREA at the bottom of the screen asking for the Knowledge to load.

- S key (SAVE): A prompt will appear in the INPUT/OUTPUT AREA at the bottom of the screen asking for the Knowledge to save.

- F1 key (UDS N-OF): A prompt will appear in the INPUT/OUTPUT AREA at the bottom of the screen. See Fig 7. It requests first any permitted UDS symbol; next a limited-range integer number; and eventually a set of character strings that will define such a symbol.

- F2 key (UDS N-OR-MORE): A prompt will appear in the INPUT/OUTPUT AREA at the bottom of the screen. See Fig. 7. It requests first any permitted UDS symbol; next a limited-range integer number; and eventually a set of character string that will define such a symbol.

- F6 key (DISPLAY UDS): A column will be displayed in the INFORMATION AREA with all the permitted UDS symbols that can be selected using the selecting arrow (this is manipulated with the up and down arrows of the keyboard). Once the target symbol to display is selected just hit <return>. See Fig. 7.

- E key (EDIT): To get in the Edit mode. The Edit menu will appear. See Fig. 4. At the top of the WORKING RULE AREA you can identify the A,B and C parts of the rules. That means the prefix, body and suffix rule respectively. The remaining part of the WORKING RULE AREA exhibits the D component of the rule structure, which is a set of phonemes denoting the sounds of the rule. The Votrax phonemes can be altered by means of eight parameters. In other words, The parameter manipulation allows the generation of phoneme variations called allophones. Fig. 13 shows the meaning of the parameters (abbreviation at the top of the WORKING RULE AREA) and their range of values.

  The Edit Menu offers a set of commands to create and edit rules.

  If you press:

  * Esc key (MAIN), you return to the Main menu.

  * Ins (INSERT), you will split the current set of phonemes to insert a new one just after the cursor location.

  * Del (DELETE), the phoneme where the cursor is positioned

will be deleted.

* + (NEXT), the next rule (if any) will be displayed.

* - (PRIOR), the previous rule will be displayed.

* Ctrl T (Edit title-rule) you will be free to develop a rule of the form A[B]C. Remember you have to enclose B (the body rule) in brackets. As mentioned, if A and C are omitted, it means that the context is irrelevant.

* F7 (NUMBER), you can get any rule in the WORKING RULE AREA by typing after a prompt that will exhibited in the INPUT/OUTPUT AREA. If you choose any rule-number that does not exist a warning message will be displayed

* F10 (SPEAK), you will listen to the set of phonemes appearing in the WORKING RULE AREA.

* F3 (PHONEME CHART), the 64 Votrax phonemes will be displayed in the INFORMATION AREA. Each phoneme is labeled with a character string. See Fig. 4.

* F4 (CHARACTERS), all the special letters and accents available will be displayed in the INFORMATION AREA. This chart indicates how to get such characters if they are needed when defining a rule (A[B]C). See Fig. 5 and Fig. 6.

* Alt F1 (INS RULE), will insert a dummy rule between the current rule and the previous rule. This option helps in having a particular sequence of the rules, because the order of the rules is significant.

* Alt F2 (DEL RULE), erases a rule from the knowledge.

```
  RULE : A[B]C =   D      (Default Values)
 NUMBER
                Phoneme    D In S X A R T
                           4 04 2 8 A A 7


                           D In S X A R T
      Duration ________|  |  | | | | |________ Transition
                          |  | | | |
        Inflection ________|  | | | |_______ Rate
                             | | |
              Slope _______| | |_____ Amplitude
                               |
                               |
                           Extension
```

Duration: This parameter allows 4 level of phoneme duration.
Acceptable rank: 1 to 4. (maximum to minimum, 4 is used for fullest phoneme pronunciation).

Inflection: Allows variation in the pitch (tone) of the phoneme.
Acceptable rank: 00 to 1F (low to high, 32 possible inflection levels).

Slope: "The Slope of the Inflections is the parameter that determines the rate at which inflections glide from one level to another" (Votalker IB [1985], p. 4-11).
Acceptable rank: 0 to 7.

Extension: "With Pitch Extension you can raise or lower the pitch in smaller intervals than inflection levels allow..." (Votalker IB [1985], p. 4-14).
Acceptable rank: 0 to F.

Amplitude: This parameter allows to stress or accent the phonemes.
Acceptable rank: 0 to F (quiet to loud, 16 amplitude levels).

Rate: "Rate adjustments are made to accommodate the fine tuning of a phoneme sequence's pronunciation" (Votalker IB, p. 4-12).
Acceptable rank: 0 to F (slow to fast, 16 rate levels).

Transition: "The Formant Transition controls the speed at which bands of frequencies in the synthesizer move." (Votalker IB, p. 4-14).

Figure 13: Votrax Phoneme Parameters

APPENDIX B
IPASP KNOWLEDGE (LETTER-TO-SOUND RULES FOR SPANISH)

```
RULE     1:[ r]     Phoneme  D In S X A R T
                      R1     3 04 3 5 A 4 7


RULE     2:[ ]      Phoneme  D In S X A R T


RULE     3:[a]      Phoneme  D In S X A R T
                      AH     4 08 0 8 A A 7
                      UH3    1 07 0 8 A A 5


RULE     4:[a]      Phoneme  D In S X A R T
                      AH     4 08 0 8 D A 7
                      UH3    2 07 0 8 F A 5


RULE     5:[b]      Phoneme  D In S X A R T
                      B      4 04 2 8 A A 5


RULE     6:[ce]     Phoneme  D In S X A R T
                      S      4 0A 0 8 A A 5
                      EH     3 06 0 8 A A 5
                      EH1    1 05 2 8 A A 5


RULE     7:[ce]     Phoneme  D In S X A R T
                      S      4 0A 2 8 A A 5
                      EH     3 06 2 8 D A 5
                      EH1    1 05 2 8 F A 5


RULE     8:[ci]     Phoneme  D In S X A R T
                      S      4 0A 0 8 A A 5
                      E      4 06 0 8 A A 5


RULE     9:[ci]     Phoneme  D In S X A R T
                      S      4 0A 0 8 A A 5
                      E      4 06 0 8 F A 5
                      E      2 05 0 8 F A 5


RULE    10:[ch]     Phoneme  D In S X A R T
                      T      2 0B 0 8 A A 5
                      SCH    3 0B 0 8 A A 5
```

RULE	11:[c]	Phoneme	D	In	S	X	A	R	T
		K	4	04	2	8	A	A	5

RULE	12:[d]	Phoneme	D	In	S	X	A	R	T
		D	4	09	2	8	A	A	5

RULE	13:[e]	Phoneme	D	In	S	X	A	R	T
		EH	4	08	0	8	F	A	7
		EH1	2	07	0	8	F	A	5

RULE	14:[e]	Phoneme	D	In	S	X	A	R	T
		EH	4	07	0	8	A	A	7
		EH1	1	06	0	8	A	A	5

RULE	15:[f]	Phoneme	D	In	S	X	A	R	T
		F	4	09	0	8	8	9	5

RULE	16:[gue]	Phoneme	D	In	S	X	A	R	T
		KV	2	0A	0	8	0	D	5
		HVC	1	08	0	8	4	D	5
		EH	4	08	0	8	9	D	5
		EH1	2	08	0	8	9	A	5

RULE	17:[gue]	Phoneme	D	In	S	X	A	R	T
		KV	2	0A	0	9	0	D	5
		HVC	1	08	0	8	4	D	5
		EH	4	08	0	8	F	D	5
		EH1	2	08	0	8	C	A	5

RULE	18:[gui]	Phoneme	D	In	S	X	A	R	T
		KV	2	0A	0	8	0	D	5
		HVC	1	08	0	8	4	D	5
		E	4	08	0	8	A	C	7
		E	2	07	0	8	A	A	5

RULE	19:[gui]	Phoneme	D	In	S	X	A	R	T
		KV	2	0A	0	8	0	D	5
		HVC	1	08	0	8	4	D	5
		E	4	09	0	9	F	C	7
		E	2	04	2	8	F	A	5

```
RULE    20:[ge]      Phoneme  D In S X A R T
                       HF     1 0B 0 8 6 8 5
                       EH     3 08 0 8 A A 7
                       EH1    1 07 0 8 A A 5

RULE    21:[ge]      Phoneme  D In S X A R T
                       HF     1 0B 0 8 6 8 5
                       EH     4 08 0 8 C A 7
                       EH1    1 08 0 8 F A 5

RULE    22:[gi]      Phoneme  D In S X A R T
                       HF     1 0B 0 8 6 8 5
                       E      4 06 0 8 A A 5
                       E      1 05 0 8 A A 5

RULE    23:[gi]      Phoneme  D In S X A R T
                       HF     1 0B 0 8 6 8 5
                       E      4 06 0 8 F A 5
                       E      2 05 0 8 D A 5

RULE    24:[g]       Phoneme  D In S X A R T
                       KV     2 0A 0 8 0 D 5
                       HVC    1 08 0 8 4 D 5

RULE    25:[h]       Phoneme  D In S X A R T
                       PA     1 04 2 8 0 A 5

RULE    26:[i]       Phoneme  D In S X A R T
                       E      4 08 0 8 A A 7
                       E      1 07 0 8 A A 5

RULE    27:[i]       Phoneme  D In S X A R T
                       E      4 08 0 8 F A 7
                       E      2 07 0 8 F A 5

RULE    28:[j]       Phoneme  D In S X A R T
                       HF     1 0B 0 8 6 8 5

RULE    29:[k]       Phoneme  D In S X A R T
                       K      4 08 0 8 A A 5
```

```
RULE    30:[lr]     Phoneme  D In S X A R T
                      L      4 04 2 8 A A 5
                      R1     3 04 3 5 A 4 7

RULE    31:[ll]     Phoneme  D In S X A R T
                      Y      4 08 0 8 A D 5

RULE    32:[l]      Phoneme  D In S X A R T
                      L      4 04 2 8 A A 5

RULE    33:[m]      Phoneme  D In S X A R T
                      M      4 08 0 8 8 9 5

RULE    34:[nr]     Phoneme  D In S X A R T
                      N      4 09 0 8 A A 5
                      R1     3 09 3 5 A 4 7

RULE    35:[n]      Phoneme  D In S X A R T
                      N      4 09 0 8 A A 5

RULE    36:[n]      Phoneme  D In S X A R T
                      NG     4 04 2 8 F A 5
                      Y      1 04 7 7 9 7 7

RULE    37:[N]      Phoneme  D In S X A R T
                      NG     4 04 2 8 F A 5
                      Y      1 04 7 7 9 7 7

RULE    38:[o]      Phoneme  D In S X A R T
                      O      4 05 0 8 A A 5

RULE    39:[o]      Phoneme  D In S X A R T
                      O      4 05 0 8 F A 5
                      O      1 05 0 8 D A 5

RULE    40:[p]      Phoneme  D In S X A R T
                      P      4 04 2 8 A A 5

RULE    41:[qu]     Phoneme  D In S X A R T
                      K      4 04 2 8 A A 5
```

```
RULE    42:[q]      Phoneme  D In S X A R T
                      K      4 04 2 8 A A 5

RULE    43:[rr]     Phoneme  D In S X A R T
                      R1     3 04 3 5 A 4 7

RULE    44:[r]      Phoneme  D In S X A R T
                      R1     4 04 7 8 3 F 5

RULE    45:[sr]     Phoneme  D In S X A R T
                      S      4 04 2 8 A A 5
                      R1     3 04 3 5 A 4 7

RULE    46:[s]      Phoneme  D In S X A R T
                      S      4 04 2 8 A A 5

RULE    47:[t]      Phoneme  D In S X A R T
                      T      4 04 2 8 A A 5

RULE    48:[u]      Phoneme  D In S X A R T
                      U      4 06 0 8 A A 5
                      U      1 05 0 8 A A 5
RULE    49:[u]      Phoneme  D In S X A R T
                      U      4 06 2 A F A 5
                      U      2 06 2 A F 8 5
RULE    50:[u]      Phoneme  D In S X A R T
                      U      4 06 2 8 A 8 7
                      U      1 05 2 8 A A 5

RULE    51:[v]      Phoneme  D In S X A R T
                      V      4 04 2 8 A A 5

RULE    52:[w]      Phoneme  D In S X A R T
                      W      4 04 2 8 A A 5

RULE    53:[x]      Phoneme  D In S X A R T
                      HF     2 04 2 8 A A 5

RULE    54:[y]      Phoneme  D In S X A R T
                      E      4 04 2 8 A A 5
```

```
RULE    55:[z]      Phoneme  D In S X A R T
                      Z      4 04 2 8 A A 5


RULE    56:[,]      Phoneme  D In S X A R T
                      PA     4 04 2 8 A A 5


RULE    57:[.]      Phoneme  D In S X A R T
                      PA     4 04 2 8 A A 5
                      PA     4 04 2 8 A A 5


RULE    58:[1]      Phoneme  D In S X A R T
                      PA     4 04 2 8 A A 5
                      U      4 04 2 8 F A 5
                      N      4 04 2 8 A A 5
                      O      4 04 2 8 A A 5
                      PA     4 04 2 8 A A 5


RULE    59:[2]      Phoneme  D In S X A R T
                      PA     4 04 2 8 A A 5
                      D      4 04 2 8 A A 5
                      O      4 04 2 8 A A 5
                      O      1 04 2 8 A A 5
                      S      4 04 2 8 A A 5
                      PA     4 04 2 8 A A 5


RULE    60:[3]      Phoneme  D In S X A R T
                      PA     4 04 2 8 A A 5
                      T      4 04 2 8 A A 5
                      R1     1 04 7 4 F 5 5
                      E1     4 04 2 8 A A 5
                      S      4 04 2 8 A A 5
                      PA     4 04 2 8 A A 5


RULE    61:[4]      Phoneme  D In S X A R T
                      PA     3 04 2 8 A A 5
                      K      4 04 2 8 A A 5
                      U      3 04 2 8 A A 5
                      UH3    4 04 2 8 A A 5
                      T      3 04 2 8 A A 5
                      R1     2 04 7 4 F 5 5
                      O      3 04 2 8 6 A 5
                      PA     4 04 2 8 A A 5
```

RULE 62:[5]

Phoneme	D	In	S	X	A	R	T
PA	4	04	2	8	A	A	5
S	4	04	2	8	A	A	5
E	4	04	2	8	A	A	5
N	4	04	2	8	A	A	5
K	4	04	2	8	A	A	5
O	3	04	2	8	5	A	5
PA	4	04	2	8	A	A	5

RULE 63:[6]

Phoneme	D	In	S	X	A	R	T
PA	4	04	2	8	A	A	5
S	4	04	2	8	A	A	5
E1	4	04	2	8	A	A	5
E	4	04	2	8	A	A	5
S	4	04	2	8	A	A	5
PA	4	04	2	8	A	A	5

RULE 64:[7]

Phoneme	D	In	S	X	A	R	T
PA	4	04	2	8	A	A	5
S	4	04	2	8	A	A	5
E	3	04	2	8	A	A	5
IE	1	04	2	8	A	A	5
E1	4	04	2	8	A	A	5
T	4	04	2	8	A	A	5
E1	3	04	2	C	4	A	5
PA	4	04	2	8	A	A	5

RULE 65:[8]

Phoneme	D	In	S	X	A	R	T
PA	4	04	2	8	A	A	5
OO	1	0A	2	8	F	A	7
O	4	0A	2	8	F	A	5
T	2	0B	2	8	A	A	5
SCH	3	0B	2	8	A	A	5
O	3	0A	2	8	6	A	5
PA	4	04	2	8	A	A	5

RULE 66:[9]

Phoneme	D	In	S	X	A	R	
PA	4	04	2	8	A	A	5
N	4	04	2	8	A	A	5
U	4	04	2	8	A	A	5
E1	4	04	2	8	C	A	5
V	4	04	2	8	A	9	5
E1	3	04	2	8	5	A	5
PA	4	04	2	8	A	A	5

RULE 67:[0]

Phoneme	D	In	S	X	A	R	T
PA	4	04	2	8	A	8	5
S	4	04	2	8	A	8	5
E1	4	04	2	8	A	9	5
R1	3	04	7	8	8	A	5
O	3	04	2	8	5	A	5
PA	4	04	2	8	A	A	0

RULE 68:[$]

Phoneme	D	In	S	X	A	R	T
PA	4	04	2	8	A	A	5
P	4	04	2	8	A	A	5
E1	4	04	2	8	C	A	5
S	4	04	2	8	A	A	5
O	3	04	2	8	8	A	5
S	4	04	2	8	A	A	5
PA	4	04	2	8	A	A	5

APPENDIX C
THE RULE-EDITOR PROGRAM

```
┌──────────────────────────────────────────────────────────────────────┐
│                         Language: BASIC                              │
│            Compiler: IBM BASIC Compiler Version 2.00                 │
└──────────────────────────────────────────────────────────────────────┘
```

```
10 '    ╔════════════════════════════════════════════════════════════════╗
12 '    ║                                                                ║
14 '    ║                    THE RULE EDITOR PROGRAM                     ║
16 '    ║                                                                ║
18 '    ╚════════════════════════════════════════════════════════════════╝
20 'UPDATED ON    March-20-1986
22 ON ERROR  GOTO 10000                      :'ERROR TRAPPING ROUTINE
24 COLOR 7,1,1:CLS
30 FOR I%=1 TO 10 :KEY I%,"":NEXT I%        :'DISABLED FUNCTION KEYS AS SOFT KEYS
100 DIM PHON$(15,11),SW.LIN%(16),RD1%(15),R1%(15),R2%(15),R3%(15),R4%(15),
       R5%(15),RULES$(999),UDS$(15)
210 PHONEMES$="PA E  E1 Y  YI AY IE I  A  A1 EH EH1AE AE1AH AH1AW O  OU OO IU "
220 PHONEMES$=PHONEMES$ + "IU1U  U1 UH UH1UH2UH3ER R  R1 R2 L  L1 LF W  B  D  "
230 PHONEMES$=PHONEMES$ + "KV P  T  K  HV HVCHF HFCHN Z  S  J  SCHV  F  THVTH "
240 PHONEMES$=PHONEMES$ + "M  N  NG :A :OH:U :UHE2 LB "
245 DC$="#$%&*+^:@"                                :'USER DEFINED SYMBOLS
250 SL%=3000                                       :'SLEEP (DELATE THE CURSOR)
260 IR%=7: IC%=6 :TR=4  :TC=11 :IPR%=IR%           :'IR,IC INITIAL ROW & COLUMN
270 LIN%=1: R5%=232
280 NUMBER.OF.RULES%=1 : RULE.NUMBER%=1 : TEXT$=" "
300 '    ╔════════════════════════════════════════════════════════════════╗
302 '    ║                                                                ║
304 '    ║                     LOCATE SONIX KERNEL                        ║
306 '    ║                                                                ║
308 '    ╚════════════════════════════════════════════════════════════════╝
350  X%=&H180                                      :'START OF USER INTERUPT VECTOR
360  WHILE X%>0 AND X%<&H1A0
370    DEF SEG= 0: SONIX.SEG#= PEEK(X%+2) + PEEK(X%+3)*256
380    X%=X%+4
390    DEF SEG= SONIX.SEG#
400 '    ╔════════════════════════════════════════════════════════════════╗
402 '    ║                  CHECK FOR SONIX'S SIGNATURE                   ║
404 '    ╚════════════════════════════════════════════════════════════════╝
410    IF PEEK(&H100)=83 AND PEEK(&H101)=79 AND PEEK(&H105)=170 THEN X%=0
420  WEND
430  WHILE NOT X%=0: 'If can't find SONIX then exit
440    PRINT "CANNOT LOCATE THE SONIX KERNEL!"
450    PRINT "RETURN TO DOS AND LOAD SONIX BEFORE CONTINUING": END
460  WEND
470 '    ╔════════════════════════════════════════════════════════════════╗
472 '    ║                  DEFINE SONIX-88 ENTRY POINTS                  ║
474 '    ╚════════════════════════════════════════════════════════════════╝
510 'SONIX Function                 Form of usage........................
```

```
540  SSI263.BUFFER%= 306         : 'CALL SSI263.BUFFER (R1%,R2%,R3%,R4%,R5%)
550  SSI263.DIRECT%= 309         : 'CALL SSI263.DIRECT (R1%,R2%,R3%,R4%,R5%)
570  SET.GLOBALS%=   318         : 'CALL SET.GLOBALS (MR%,MA%,MI%,MF%)
590  INIT.SPEECH%=   327         : 'CALL INIT.SPEECH
600  QUIET%=         330         : 'CALL QUIET
610  WAIT.EMPTY%=    333         : 'CALL WAIT.EMPTY
630  DEF SEG= SONIX.SEG#
1000 '
1002 '
1004 '                         SCREEN DEFINITION
1006 '
1010 '
1020 BACKG=1 : BOR= 1: WBACKG=7                       : ' COLOR PARAMETES
1030 COLOR 7,BACKG,BOR                                : ' WHITE; BLUE; BLUE
1035 CLS :SCREEN 0:WIDTH 80 :KEY OFF                  : ' CLEANNING THE SCREEN
1040 COLOR 7,1,1 : CLS
1050 TM=1 :BM=5 :NA=25                                : ' BOX (TOP WORK AREA)
1055 LM=1 :RM=25:WBACKG=0:GOSUB 5300:
     BACKG=0:CUR=7:GOSUB 5120:BACKG=1
1057 TM=5 :BM=22:NA=25                                : ' BOX (WORK AREA)
1059 LM=1 :RM=25:WBACKG=0:GOSUB 5300:BACKG=0:
     CUR=7:GOSUB 5120:GOSUB 5400:TM=6:BM=21:
     BACKG=1:LM=2:RM=4:WBACKG=7:NA=3:GOSUB 5300
1060 TM=1 :BM=21:NA=29                                : ' BOX (INFORMATION AREA)
1065 LM=51:RM=79:CUR=7:GOSUB 5120
1070 TM=1 :BM=5 :NA=23                                : ' BOX (TOP MENU AREA)
1075 LM=27:RM=49:WBACKG=7:GOSUB 5300:
     BACKG=7:CUR=1:GOSUB 5120:BACKG=1
1080 TM=5 :BM=21:NA=23                                : ' BOX (MENU AREA)
1085 LM=27:RM=49:CUR=7:GOSUB 5300:
     BACKG=7:CUR=1:GOSUB 5120:GOSUB 5400:
     BACKG=1:CUR=7
1090 TM=22:BM=24:NA=79:WBACKG=0 :BACKG=0              : ' BOX (BOTTOM INPUT DATA)
1095 LM=1 :RM=79:GOSUB 5300:GOSUB 5120
1098 LOCATE 22,1 :PRINT CHR$(204);:
     LOCATE 22,25:PRINT CHR$(202);:                   :' CORNERS' BOXES WORK AREAS
1100 GOSUB 6900                                       :'WORK AREA DISPLAY
1200 COLOR 7,1,1
2000 '
2002 '
2004 '                            MAIN  MENU
2008 '
2010 '
2200 GOSUB 5500                                       :'MAIN MENU DISPLAY
2210 IP$=INKEY$ : IF IP$="" THEN 2210                 :'READING FROM KEYBOARD
2220 IP%=ASC(IP$)                                     :'GETTING ASCII VALUE
2225 IF (IP%>96) AND (IP%<123) THEN
        IP%=IP%-32:IP$=CHR$(IP%)                      :'ASSURE UPPER CASE
2230 IF ASC(MID$(IP$,1,1))<>0 THEN 2300               :'COMMON ASCII VALUE
2240 SKEY%=ASC(MID$(IP$,2,1))                         :'SPECIAL KEY COMBINATIONS
2242 '
2244 '                 SPECIAL KEY  CODE ASSIGNATIONS
2246 '
```

```
2250 IF SKEY% = 59 THEN GOSUB 11000:GOTO 2210   :'F1      UDS N-OF
2260 IF SKEY% = 60 THEN GOSUB 11000:GOTO 2210   :'F2      UDS N-OR-MORE
2270 IF SKEY% = 64 THEN GOSUB 12000:GOTO 2210   :'F6      DISPLAY UDS
2300 ' ╔══════════════════════════════════════════════════════════════╗
2302 ' ║                DIRECT ASCII CODE ASSIGNATIONS                ║
2304 ' ╚══════════════════════════════════════════════════════════════╝
2310 IF IP%=81 THEN 2500                        :'Q QUIT
2320 IF IP%=76 THEN GOSUB 11800 : GOTO 2210     :'L LOAD
2330 IF IP%=83 THEN GOSUB 11500 : GOTO 2210     :'S SAVE
2340 IF IP%=80 THEN                             :'P PRINT
2350 IF IP%=69 THEN 3000                        :'E EDIT
2400 GOTO 2210
2500 COLOR 7,0,0:CLS:END
2900 ' ╔══════════════════════════════════════════════════════════════╗
2902 ' ║                      CURSOR'S RIGHT & LEFT                   ║
2904 ' ╚══════════════════════════════════════════════════════════════╝
2910 IF POS(0) > 6  THEN GOSUB 2960: LOCATE IR%,POS(0)-1:'MOVE CURSOR ONE LEFT
2920 RETURN
2950 IF POS(0) < 23 THEN GOSUB 2960: LOCATE IR%,POS(0)+1:'MOVE CURSOR ONE RIGHT
2952 RETURN
2960 ON POS(0)-5 GOSUB 2965,2965,2965,2980,2966,2980,2968,2968,2980,2970,2980,
                       2972,2980,2974,2980,2976,2980,2978,2980
2962 RETURN
2965 PRINT USING "!";PHON$(LIN%,POS(0)-5);:
     LOCATE IR%,POS(0)-1:RETURN                        :'THREE FIRST CHARACT
2966 PRINT USING "!";PHON$(LIN%,4);:
     LOCATE IR%,POS(0)-1:RETURN                        :'DURATION CHARACTER
2968 PRINT USING "!";PHON$(LIN%,POS(0)-7);:
     LOCATE IR%,POS(0)-1:RETURN                        :'TWO INFLECTION CHAR.
2970 PRINT USING "!";PHON$(LIN%,7);:
     LOCATE IR%,POS(0)-1:RETURN                        :'SLOPE CHARACTER
2972 PRINT USING "!";PHON$(LIN%,8);:
     LOCATE IR%,POS(0)-1:RETURN                        :'EXTENSION CHARACTER
2974 PRINT USING "!";PHON$(LIN%,9);:
     LOCATE IR%,POS(0)-1:RETURN                        :'AMPLITUDE CHARACTER
2976 PRINT USING "!";PHON$(LIN%,10);:
     LOCATE IR%,POS(0)-1:RETURN                        :'RATE CHARACTER
2978 PRINT USING "!";PHON$(LIN%,11);:
     LOCATE IR%,POS(0)-1:RETURN                        :'TRANSITION CHARACTER
2980 PRINT CHR$(32);: LOCATE IR%,POS(0)-1 : RETURN     :'CLEAR BLANK POSITIONS
2990 STOP
3000 ' ╔══════════════════════════════════════════════════════════════╗
3002 ' ║                                                              ║
3004 ' ║                          EDIT  MENU                          ║
3008 ' ║                                                              ║
3010 ' ╚══════════════════════════════════════════════════════════════╝
3020 GOSUB 5700                                 :'EDIT MENU DISPLAY
3030 COLOR 7,0                                  :'WHITE,BLACK
3050 IR%=7: IC%=6                               :'IR,IC INITIAL ROW & COLUMN
3060 LOCATE IR%,IC%
3070 LIN%=1
3210 CPOS%=SCREEN (IR%,POS(0))                  :'CURRENT CHARACTER
3215 IP$=INKEY$:IF IP$="" THEN PRINT CHR$(219);:LOCATE CSRLIN,POS(0)-1:
     FOR I%=1 TO SL%:NEXT I%:PRINT CHR$(CPOS%);:LOCATE CSRLIN,POS(0)-1:
     FOR I%=1 TO SL%:NEXT I%:GOTO 3215
3220 IP%=ASC(IP$)                                :'GETTING ASCII VALUE
3225 IF (IP%>96) AND (IP%<123) THEN
        IP%=IP%-32:IP$=CHR$(IP%)                 :'ASSURE UPPER CASE
3230 IF ASC(MID$(IP$,1,1))<>0 THEN 3300          :'COMMON ASCII VALUE
3242 ' ╔══════════════════════════════════════════════════════════════════╗
3244 ' ║                  SPECIAL KEY  CODE ASSIGNATIONS                  ║
3246 ' ╚══════════════════════════════════════════════════════════════════╝
3248 SKEY%=ASC(MID$(IP$,2,1))                    :'SPECIAL KEY COMBINATIONS
3250 IF SKEY% = 82 THEN GOSUB 7300: GOTO 3210    :'Ins     INSERT
3255 IF SKEY% = 83 THEN GOSUB 7100: GOTO 3210    :'Del     DELETE
3256 IF SKEY% = 65 THEN GOSUB 12500:GOTO 3210    :'F7      NUMBER
3260 IF SKEY% = 68 THEN GOSUB 8150: GOTO 3210    :'F10     SPEAK
3261 IF SKEY% = 61 THEN GOSUB 6000: GOTO 3210    :'F3      PHONEME CHART
3262 IF SKEY% = 62 THEN GOSUB 6300: GOTO 3210    :'F4      CHARACTER CHART
3265 IF SKEY% = 75 THEN GOSUB 2910: GOTO 3210    :'CURSOR LEFT
3270 IF SKEY% = 77 THEN GOSUB 2950: GOTO 3210    :'CURSOR RIGHT
3275 IF SKEY% = 72 THEN GOSUB 7000: GOTO 3210    :'CURSOR UP
3280 IF SKEY% = 80 THEN GOSUB 7050: GOTO 3210    :'CURSOR DOWN
3282 IF SKEY% =104 THEN GOSUB 11300:GOTO 3210    :'Alt F1  INSERT RULE
3284 IF SKEY% =105 THEN GOSUB 11200:GOTO 3210    :'Alt F2  DELETE RULE
3300 ' ╔══════════════════════════════════════════════════════════════════╗
3302 ' ║                  DIRECT ASCII CODE ASSIGNATIONS                  ║
3304 ' ╚══════════════════════════════════════════════════════════════════╝
3305 P%=POS(0)                                   :'CURRENT CURSOR POSITION
3310 IF IP% = 13 THEN GOSUB 3900: GOSUB 3450:
                  GOTO 3210                      :'RETURN  PRINT DEFAULT VALUES
3325 IF IP% = 27 THEN GOSUB 8000: GOSUB 8200:
                  GOTO 2000                      :'Esc     MAIN MENU
3330 IF IP% = 16 THEN                            :'Ctrl P  PRINT
3335 IF IP% = 43 THEN GOSUB 8800: GOTO 3210      :' +      NEXT
3340 IF IP% = 45 THEN GOSUB 8900: GOTO 3210      :' -      PRIOR
3350 IF IP% = 20 THEN GOSUB 4000: GOTO 3210      :'Ctrl T  RULE TITLE
3360 ON POS(0)-5 GOSUB 3490,3490,3490,2950,3625,2950,3680,3840,2950,3740,2950,
                   3840,2950,3840,2950,3840,2950,3740
3390 GOTO 3210                                   :'CLOSE THE LOOP
3400 ' ╔══════════════════════════════════════════════════════════════════╗
3402 ' ║                   PHONEMES INPUT & VALIDATION                    ║
3404 ' ╚══════════════════════════════════════════════════════════════════╝
3450 CPOS%=SCREEN (IR%,POS(0))                           :'CURRENT CHARACTER
3455 IP$=INKEY$:IF IP$="" THEN PRINT CHR$(219);:LOCATE CSRLIN,POS(0)-1:
     FOR I%=1 TO SL%:NEXT I%:PRINT CHR$(CPOS%);:LOCATE CSRLIN,POS(0)-1:
     FOR I%=1 TO SL%:NEXT I%:GOTO 3455
3460 IP%=ASC(IP$)
3462 IF ASC(MID$(IP$,1,1))<>0 THEN 3470                  :'COMMON ASCII VALUE
3464 SKEY%=ASC(MID$(IP$,2,1))                            :'EXTENDED CODES
3466 IF SKEY% = 75 THEN GOSUB 2910: GOTO 3450            :'CURSOR ONE POS. LEFT
3468 IF SKEY% = 77 THEN GOSUB 2950: GOTO 3520            :'CURSOR ONE POS. RIGHT
3470 IF IP%<>8 THEN 3490                                  :'IF NOT BACKSPACE
3472 IF POS(0)-IC% > 0 THEN LOCATE IR%,POS(0)-1:
     PRINT USING "\ \";"   ";: LOCATE IR%,POS(0)-3:
     GOTO 3450
```

```
3475 PLAY "L8N10": GOTO 3450
3490 IF (IP% > 96) AND (IP% < 123) THEN IP%=IP%-32 :
     IP$=CHR$(IP%)                                      :'ASSURE UPPER CASE
3492 IF ((IP%>63) AND (IP%<91)) OR IP%=32 OR IP%=58 OR ((IP%>48) AND (IP%<52))
     THEN 3500 ELSE PLAY "L8N10":GOTO 3450
3500 PRINT USING "!";IP$;                               :'PRINT & INCREMENT POS
3505 DEBG=POS(0)
3510 PHON$(LIN%,POS(0)-6)=IP$
3515 IF SW.LIN%(LIN%)=0 THEN SW.NOT.SKIP%=1:GOSUB 3920:
     SW.LIN%(LIN%)=1
3520 IF POS(0) < 9 THEN 3450                            :'CHECKING PHON LENGTH
3525 DEBG=POS(0)
3550 FOR I%=1 TO 190 STEP 3                             :'POSSIBLE PHONEMES
3560 IF MID$(PHONEMES$,I%,3)=
     PHON$(LIN%,1)+PHON$(LIN%,2)+PHON$(LIN%,3)
     THEN 3600
3570 NEXT I%
3575 PHON$(LIN%,1)=" ":
     PHON$(LIN%,2)=" ":
     PHON$(LIN%,3)=" "
3580 LOCATE IR%,IC% :PRINT "   ";               :'INVALID PHONEME, CLEAN SPACES
3590 PLAY "L8N10": LOCATE IR%,IC% :GOTO 3450    :'RETURN TO GENERAL EDIT INPUT
3600 R1%(LIN%)=(I%-1)/3                         :'R1% REGISTER PHONEME/DURATION
3605 LOCATE IR%,10                              :'CURSOR INCREMENT
3607 RETURN
3610 '
3612 '                    DURATION INPUT & VALIDATION
3614 '
3620 IP$=INKEY$:IF IP$="" THEN PRINT CHR$(219);:LOCATE CSRLIN,POS(0)-1:
     FOR I%=1 TO SL%:NEXT I%:PRINT CHR$(32);    :LOCATE CSRLIN,POS(0)-1:
     FOR I%=1 TO SL%:NEXT I%:GOTO 3620
3622 IP%=ASC(IP$)
3625 IF (IP% > 52) OR (IP% <49) THEN PLAY "L8N10":PRINT CHR$(32);:
     LOCATE CSRLIN,POS(0)-1: GOTO 3620
3630 PRINT USING "!";IP$;
3633 PHON$(LIN%,4)=IP$
3650 LOCATE IR%,12: RETURN
3670 '
3672 '                   INFLECTION INPUT & VALIDATION
3674 '
3675 IP$=INKEY$:IF IP$="" THEN PRINT CHR$(219);:LOCATE CSRLIN,POS(0)-1:
     FOR I%=1 TO SL%:NEXT I%:PRINT CHR$(32);    :LOCATE CSRLIN,POS(0)-1:
     FOR I%=1 TO SL%:NEXT I%:GOTO 3675
3678 IP%=ASC(IP$)
3680 IF (IP%<>48) AND (IP%<>49) THEN
     PLAY "L8N10" : GOTO 3675          :'FIRST CHARACTER VALIDATION (0-1)
3688 PRINT USING "!";IP$;
3689 PHON$(LIN%,5)=IP$
3690 RETURN
3720 '
3722 '               SLOPE, TRANSITION  (INPUT & VALIDATION)
3724 '
3730 IP$=INKEY$:IF IP$="" THEN PRINT CHR$(219);:LOCATE CSRLIN,POS(0)-1:
     FOR I%=1 TO SL%:NEXT I%:PRINT CHR$(32);    :LOCATE CSRLIN,POS(0)-1:
     FOR I%=1 TO SL%:NEXT I%:GOTO 3730
3735 IP%=ASC(IP$)
3740 IF (IP%>47) AND (IP%<56) THEN 3750
     ELSE PLAY"L8N10" : GOTO 3730              :'SLOPE VALIDATION (0-7)
3750 PRINT USING "!";IP$;
3752 IF POS(0)=16 THEN PRINT CHR$(32);:
     PHON$(LIN%,7)=IP$:GOTO 3755             :'IF SLOPE NEXT CURSOR POSITION
3753 PHON$(LIN%,11)=IP$:LOCATE CSRLIN,POS(0)-1 :'SAVE TRANSITION VALUE
3755 RETURN
3820 '
3822 '     INFLECTION, EXTENSION, AMPLITUDE, RATE   (INPUT & VALIDATION)
3824 '
3830 IP$=INKEY$:IF IP$="" THEN PRINT CHR$(219);:LOCATE CSRLIN,POS(0)-1:
     FOR I%=1 TO SL%:NEXT I%:PRINT CHR$(32);    :LOCATE CSRLIN,POS(0)-1:
     FOR I%=1 TO SL%:NEXT I%:GOTO 3830
3835 IP%=ASC(IP$)
3840 IF (IP%>96) AND (IP%<103) THEN
     IP%=IP%-32:IP$=CHR$(IP%)            :'ASSURE UPPER CASE
3845 IF ((IP%>47) AND (IP%<58)) OR
     ((IP%>64) AND (IP%<71)) THEN 3850
     ELSE PLAY"L8N10" : GOTO 3830        :'EXTENSION CHAR. VALIDATION (0-F)
3850 PRINT USING "!";IP$;
3852 P%=POS(0)
3853 IF P%=14 THEN PHON$(LIN%,6)=IP$     :'SAVE SECOND INFLECTION CHARACTER
3854 IF P%=18 THEN PHON$(LIN%,8)=IP$     :'SAVE EXTENSION PARAMETER
3855 IF P%=20 THEN PHON$(LIN%,9)=IP$     :'SAVE AMPLITUDE PARAMETER
3856 IF P%=22 THEN PHON$(LIN%,10)=IP$    :'SAVE RATE PARAMETER
3860 PRINT CHR$(32);: RETURN             :'MOVE RIGHT CURSOR, RETURN TO EDIT
3900 '
3902 '                   LINE INCREMENT  /   DEFAULT VALUES
3904 '
3905 IF SW.LIN%(LIN%)=0 THEN 3948                           :'EMPTY LINE
3906 IF CSRLIN+1 > 21 THEN PLAY "L8N10" :GOTO 3948
3910 IR%=CSRLIN+1 : LIN%=IR%-6                              :'NEXT LINE
3915 IF SW.LIN%(LIN%)=1 THEN 3945
3920 CR%=CSRLIN : CC%=POS(0) : C% = SCREEN(CR%,CC%)         :'CURSOR POS.
3921 IF C%=95 OR C%=219 OR C%=32 THEN
     PRINT CHR$(32); ELSE PRINT CHR$(C%);
3922 COLOR 0,7 : LOCATE IR%,IC%-4 : PRINT USING "###";LIN%;
3925 COLOR 7,0 : LOCATE IR%,IC%+4 : PRINT "4 04 2 8 A A 5";
3930 PHON$(LIN%,4)="4":PHON$(LIN%,5)="0":PHON$(LIN%,6)="4"
3935 PHON$(LIN%,7)="2":PHON$(LIN%,8)="8":PHON$(LIN%,9)="A"
3940 PHON$(LIN%,10)="A":PHON$(LIN%,11)="5"
3943 IF SW.NOT.SKIP%=1 THEN LOCATE CR%,CC%:SW.NOT.SKIP%=0 :
      GOTO 3948
3945 LOCATE IR%,IC%                                         :'INITIAL LOC.
3948 RETURN
4000 '
4002 '
4005 '                         EDIT RULE TITLE
4007 '
4008 '
```

```
4010 COLOR 7,0                      : 'WHITE,BLACK
4012 NUM.CHAR=0 :TEXT$=""           : 'NUMBER OF CHARACTERS IN THE TITLE TEXT
4015 BUF.LENGHT=11                  : 'TEXT BUFFER LENGHT
4020 TR=4 : TC=11                   : 'TR (TITLE ROW)  : TC (TITLE COLUMN)
4022 LOCATE TR,TC:
     PRINT STRING$(12,32);:
     LOCATE TR,TC
4023 GOSUB 4030                     : 'INPUT TITLE TEXT
4025 LOCATE IR%,IC% : RETURN        : 'END OF RULE TITLE PROCEDURE
4030 ' ╔══════════════════════════════════════════════════════════════════════╗
4031 ' ║                                                                      ║
4032 ' ║ KEYBOARD SELECTION AND ASSIGNATION OF SPECIAL SYMBOLS AND ACCENTS .  ║
4033 ' ║                                                                      ║
4034 ' ╚══════════════════════════════════════════════════════════════════════╝
4035 SW.INPUT=1                     : 'SWITCHE FOR INPUT DATA (GET IN THE WHILE)
4040 WHILE SW.INPUT
4050 PRINT CHR$(219);:LOCATE TR,POS(0)-1
4055 T$=INKEY$:IF T$="" THEN 4055
4060 T=ASC(T$)
4070 IF T<>8 THEN 4080                                   :'IF NOT BACKSPACE
4072 IF NUM.CHAR>0 THEN LOCATE TR,POS(0)-1:
      PRINT USING "\ \";"   ";: LOCATE TR,POS(0)-3:
      NUM.CHAR=NUM.CHAR-1: GOTO 4050
4080 IF T=13 THEN SW.INPUT=0:PRINT CHR$(32);:T$="":GOTO 4780
4090 IF NUM.CHAR>BUF.LENGHT THEN PLAY "L12GB-B-4L2" :
        GOTO 4050
4095 IF T>64 AND T<91 THEN T=T+32 : T$=CHR$(T)           :'ASSURE LOWER CASE
4100 ' ╔══════════════════════════════════════════════╗
4102 ' ║  ACCENT    :                                 ║
4104 ' ╚══════════════════════════════════════════════╝
4110 IF ACCENT1 = 0 THEN 4160
4120    ACCENT1=0
4130    IF T=97  THEN T$=CHR$(166) : SW=1 : GOTO 4150
4140    IF T=111 THEN T$=CHR$(167) : SW=1
4150 IF SW=0 THEN 4570 ELSE SW=0: GOTO 4760
4160 ' ╔══════════════════════════════════════════════╗
4162 ' ║  ACCENT    :                                 ║
4164 ' ╚══════════════════════════════════════════════╝
4165 IF ACCENT2 = 0 THEN 4210
4170    ACCENT2=0
4180    IF T=97  THEN T$=CHR$(134) : SW=1 : GOTO 4200
4190    IF T=65  THEN T$=CHR$(143) : SW=1
4200 IF SW=0 THEN 4570 ELSE SW=0: GOTO 4760
4210 ' ╔══════════════════════════════════════════════╗
4212 ' ║  ACCENT    :                                 ║
4214 ' ╚══════════════════════════════════════════════╝
4215 IF ACCENT3 = 0 THEN 4290
4220    ACCENT3=0
4230    IF T=97  THEN T$=CHR$(131) : SW=1 : GOTO 4280
4240    IF T=101 THEN T$=CHR$(136) : SW=1 : GOTO 4280
4250    IF T=105 THEN T$=CHR$(140) : SW=1 : GOTO 4280
4260    IF T=111 THEN T$=CHR$(147) : SW=1 : GOTO 4280
4270    IF T=117 THEN T$=CHR$(150) : SW=1
```

```
4280 IF SW=0 THEN 4570 ELSE SW=0: GOTO 4760
4290 '  ╔══════════════════════════════════════════╗
4292 '  ║ ACCENT    :                              ║
4294 '  ╚══════════════════════════════════════════╝
4295 IF ACCENT4 = 0 THEN 4370
4300    ACCENT4=0
4310    IF T=97  THEN T$=CHR$(133) : SW=1 : GOTO 4280
4320    IF T=101 THEN T$=CHR$(138) : SW=1 : GOTO 4280
4330    IF T=105 THEN T$=CHR$(141) : SW=1 : GOTO 4280
4340    IF T=111 THEN T$=CHR$(149) : SW=1 : GOTO 4280
4350    IF T=117 THEN T$=CHR$(151) : SW=1
4360 IF SW=0 THEN 4570 ELSE SW=0: GOTO 4760
4370 '  ╔══════════════════════════════════════════╗
4372 '  ║ ACCENT    :                              ║
4374 '  ╚══════════════════════════════════════════╝
4375 IF ACCENT5= 0 THEN 4480
4380    ACCENT5=0
4390    IF T=97  THEN T$=CHR$(132) : SW=1 : GOTO 4470
4400    IF T=101 THEN T$=CHR$(137) : SW=1 : GOTO 4470
4410    IF T=105 THEN T$=CHR$(139) : SW=1 : GOTO 4470
4420    IF T=111 THEN T$=CHR$(148) : SW=1 : GOTO 4470
4430    IF T=117 THEN T$=CHR$(129) : SW=1 : GOTO 4470
4435    IF T=121 THEN T$=CHR$(152) : SW=1 : GOTO 4470
4440    IF T=65  THEN T$=CHR$(142) : SW=1 : GOTO 4470
4450    IF T=79  THEN T$=CHR$(153) : SW=1 : GOTO 4470
4460    IF T=85  THEN T$=CHR$(154) : SW=1
4470 IF SW=0 THEN 4570 ELSE SW=0: GOTO 4760
4480 '  ╔══════════════════════════════════════════╗
4482 '  ║ ACCENT    :                              ║
4484 '  ╚══════════════════════════════════════════╝
4485 IF ACCENT6 = 0 THEN 4570
4490    ACCENT6=0
4500    IF T=97  THEN T$=CHR$(160) : SW=1 : GOTO 4560
4510    IF T=101 THEN T$=CHR$(130) : SW=1 : GOTO 4560
4520    IF T=105 THEN T$=CHR$(161) : SW=1 : GOTO 4560
4530    IF T=111 THEN T$=CHR$(162) : SW=1 : GOTO 4560
4540    IF T=117 THEN T$=CHR$(163) : SW=1 : GOTO 4560
4550    IF T=69  THEN T$=CHR$(144) : SW=1 : GOTO 4560
4560 IF SW=0 THEN 4570 ELSE SW=0: GOTO 4760
4570 IF (T>34 AND T<123) OR T=32 OR T=33 THEN 4760
4580 IF ASC(MID$(T$,1,1))<>0 THEN GOTO 4710
4590 SPECIAL.KEY=ASC(MID$(T$,2,1))                       :'EXTENDED CODES
4600 IF SPECIAL.KEY<126 OR SPECIAL.KEY>131 THEN 4680
4610 ON SPECIAL.KEY-125 GOTO 4620,4630,4640,4650,4660,4670
4620 ACCENT1=1 : GOTO 4790
4630 ACCENT2=1 : GOTO 4790
4640 ACCENT3=1 : GOTO 4790
4650 ACCENT4=1 : GOTO 4790
4660 ACCENT5=1 : GOTO 4790
4670 ACCENT6=1 : GOTO 4790
4680 IF SPECIAL.KEY=49 THEN T$=CHR$(164) : GOTO 4760    :' ñ . Alt N
4690 IF SPECIAL.KEY=46 THEN T$=CHR$(135) : GOTO 4760    :' ç . Alt C
4700 IF SPECIAL.KEY=30 THEN T$=CHR$(145) : GOTO 4760    :' æ . Alt A
```

```
4710 IF T=14 THEN T$=CHR$(165) : GOTO 4760                  :' Ñ . Ctrl N
4720 IF T=3  THEN T$=CHR$(128) : GOTO 4760                  :' Ç . Ctrl C
4730 IF T=1  THEN T$=CHR$(146) : GOTO 4760                  :' Æ . Ctrl A
4740 '
4742 '              CONCLUDE SELECTION
4744 '
4750 GOTO 4790                                              :' WEND
4760 '
4762 '       PRINT CHARACTER & UPDATE COUNTERS
4764 '
4770 PRINT USING "!";T$;
4780 TEXT$=LEFT$(TEXT$,NUM.CHAR)+T$ : NUM.CHAR=NUM.CHAR+1
4790 WEND
4800 RETURN
5000 '
5002 '
5004 '
5020 BACKG=1 : BOR= 1: WBACKG=7
5030 COLOR 7,BACKG,BOR       : ' WHITE; BLACK; BLACK
5040 CLS :SCREEN 0:WIDTH 80 :KEY OFF
5050 TM=1: BM=21: NA=24: LM=1 : RM=24 : CUR=7 : GOSUB 5120
5060 NA=29: LM=51: RM=79 : CUR=7 : GOSUB 5120
5070 TM=1:BM=5:NA=23:LM=26:RM=49:GOSUB 5300:BACKG=7:CUR=1:GOSUB 5120:BACKG=1
5080 TM=5:BM=21:NA=23:LM=26:RM=49:CUR=7:GOSUB 5300:BACKG=7:CUR=1:GOSUB 5120:
     GOSUB 5400:BACKG=1:CUR=7
5090 TM=22:BM=24:NA=79: LM=1: RM=79 : GOSUB 5120
5100 GOTO 5500
5120 COLOR CUR,BACKG,BOR
5130 A$=STRING$(NA,CHR$(205))
5140 LOCATE TM,LM:PRINT A$;
5150 LOCATE BM,LM: PRINT A$;
5160 FOR I= TM+1 TO BM-1
5170 LOCATE I,LM: PRINT CHR$(186);
5180 LOCATE I,RM: PRINT CHR$(186);
5190 NEXT I
5200 'CORNERS
5210 LOCATE TM,LM: PRINT CHR$(201);
5220 LOCATE BM,LM: PRINT CHR$(200);
5230 LOCATE BM,RM: PRINT CHR$(188);
5240 LOCATE TM,RM: PRINT CHR$(187);
5250 RETURN
5300 COLOR WBACKG
5310 FOR I=TM TO BM
5320 LOCATE I,LM:PRINT STRING$(NA,CHR$(219));
5330 NEXT I
5340 RETURN
5400 LOCATE TM,LM: PRINT CHR$(204);
5410 LOCATE TM,RM: PRINT CHR$(185);
5420 RETURN
5500 '
5502 '                      MAIN MENU WINDOW DEFINITION
5504 '
5510 COLOR 1,7
```

```
5515 ERA$=STRING$(21," ")
5520 '                   12345678901234567890123
5530 LOCATE  6,28:PRINT "  QUIT........... Q";
5535 LOCATE  7,28:PRINT    ERA$
5537 LOCATE  9,28:PRINT    ERA$
5540 '                  "_______KNOWLEDGE______";
5550 LOCATE 10,28:PRINT "  LOAD........... L";
5560 LOCATE 11,28:PRINT "  SAVE........... S";
5570 LOCATE 12,28:PRINT    ERA$
5575 LOCATE 13,28:PRINT    ERA$
5580 '                  "_________RULES________";
5585 LOCATE 15,28:PRINT    ERA$
5590 LOCATE 16,28:PRINT "  EDIT........... E ";
5600 LOCATE 17,28:PRINT "  UDS N-OF...... F1 ";
5610 LOCATE 18,28:PRINT "  UDS N-OR-MORE. F2 ";
5630 LOCATE 19,28:PRINT "  DISPLAY UDS... F6 ";
5640 LOCATE 20,28:PRINT    ERA$
5650 LOCATE  3,28:PRINT "______MAIN MENU______";
5655 COLOR 4
5660 LOCATE  8,28:PRINT "______KNOWLEDGE______";
5670 LOCATE 14,28:PRINT "________RULES________";
5680 COLOR 7,0:RETURN
5700 '  ╔════════════════════════════════════════════════════════════════════════╗
5702 '  ║                      EDIT MENU WINDOW DEFINITION                       ║
5704 '  ╚════════════════════════════════════════════════════════════════════════╝
5710 COLOR 1,7
5715 R=28 : 'ROW=28
5720 '                   12345678901234567890123
5722 LOCATE  3,R :PRINT "______EDIT MENU______";
5725 LOCATE  6,R :PRINT "  MAIN......... Esc  ";
5730 LOCATE  7,R :PRINT "_____________________";
5740 LOCATE  8,R :PRINT ERA$;
5760 LOCATE  9,R :PRINT "  INSERT....... Ins  ";
5770 LOCATE 10,R :PRINT "  DELETE....... Del  ";
5780 LOCATE 11,R :PRINT "  NEXT........... +  ";
5790 LOCATE 12,R :PRINT "  PRIOR.......... -  ";
5800 LOCATE 13,R :PRINT "  EDIT TITLE.... ^T  ";
5805 LOCATE 14,R :PRINT "  NUMBER........ F7  ";
5810 LOCATE 15,R :PRINT "  SPEAK........ F10  ";
5820 LOCATE 16,R :PRINT "  PHONEME CHART. F3  ";
5830 LOCATE 17,R :PRINT "  CHARACTERS.... F4  ";
5835 LOCATE 18,R :PRINT "  INS RULE...Alt F1  ";
5840 LOCATE 19,R :PRINT "  DEL RULE...Alt F2  ";
5845 LOCATE 20,R :PRINT ERA$;
5900 RETURN
6000 '  ╔════════════════════════════════════════════════════════════════════════╗
6002 '  ║                   PHONEME CHART  WINDOW DEFINITION                     ║
6004 '  ╚════════════════════════════════════════════════════════════════════════╝
6010 COLOR 7,1
6015 C=52 : R=6
6017 LOCATE 2,   C:PRINT "                           ";
6020 LOCATE 3,   C:PRINT "_______PHONEME CHART_______";
6025 LOCATE 4,   C:PRINT "___________________________";
```

```
6030 '                        12345678901234567890123456789
6035 LOCATE 5,   C:PRINT "                             ";
6040 LOCATE R   ,C:PRINT "A      A1     :A     AE     AE1";
6050 LOCATE R+1 ,C:PRINT "AH     AH1    AW     AY     B  ";
6060 LOCATE R+2 ,C:PRINT "D      E      E1     E2     EH ";
6070 LOCATE R+3 ,C:PRINT "EH1    ER     F      HF     HFC";
6080 LOCATE R+4 ,C:PRINT "HN     HV     HVC    I      IE ";
6090 LOCATE R+5 ,C:PRINT "IU     IU1    J      K      KV ";
6100 LOCATE R+6 ,C:PRINT "L      L1     LB     LF     M  ";
6110 LOCATE R+7 ,C:PRINT "N      NG     O      :OH    OO ";
6120 LOCATE R+8 ,C:PRINT "OU     P      R      R1     R2 ";
6130 LOCATE R+9 ,C:PRINT "S      SCH    T      TH     THV";
6140 LOCATE R+10,C:PRINT "U      U1     :U     UH     UH1";
6150 LOCATE R+11,C:PRINT "UH2    UH3    :UH    V      W  ";
6160 LOCATE R+12,C:PRINT "Y      YI     Z             PA ";
6165 LOCATE R+13,C:PRINT "                             ";
6168 LOCATE R+14,C:PRINT "                             ";
6200 LOCATE IR%,IC%:COLOR 7,0:RETURN
6300 ' ╔═══════════════════════════════════════════════════════════════════╗
6302 ' ║                        CHARACTER CHART                            ║
6304 ' ╚═══════════════════════════════════════════════════════════════════╝
6310 COLOR 7,1 :C=52 :R=5
6315 LOCATE 2,   C:PRINT "                            ";
6320 LOCATE 3,   C:PRINT "______CHARACTER CHART______";
6325 LOCATE 4,   C:PRINT "                            ";
6330 IF SWF4%=1 THEN SWF4%=0 : GOTO 6500                :'CHARACTER CHART [2]
6335 '                        12345678901234567890123456789
6340 LOCATE R   ,C:PRINT "      ┌─┬─┬─┬─┬─┬─┬─┬─┬─┬─┐ "
6342 LOCATE R+1 ,C:PRINT "      │a│e│i│o│u│y│A│E│O│U│ "
6345 LOCATE R+2 ,C:PRINT "┌─────┼─┼─┼─┼─┼─┼─┼─┼─┼─┼─┤
6355 LOCATE R+3 ,C:PRINT "│Alt =│á│é│í│ó│ú│ │ │É│ │ │ ";
6370 LOCATE R+4 ,C:PRINT "├─────┼─┼─┼─┼─┼─┼─┼─┼─┼─┼─┤
6375 LOCATE R+5 ,C:PRINT "│Alt -│ā│ē│ī│ō│ū│ȳ│Ā│ │Ō│Ū│
6380 LOCATE R+6 ,C:PRINT "├─────┼─┼─┼─┼─┼─┼─┼─┼─┼─┼─┤
6390 LOCATE R+7 ,C:PRINT "│Alt 0│à│è│ì│ò│ù│ │ │ │ │ │
6395 LOCATE R+8 ,C:PRINT "├─────┼─┼─┼─┼─┼─┼─┼─┼─┼─┼─┤
6400 LOCATE R+9 ,C:PRINT "│Alt 9│ă│ĕ│ĭ│ŏ│ŭ│ │ │ │ │ │
6405 LOCATE R+10,C:PRINT "├─────┼─┼─┼─┼─┼─┼─┼─┼─┼─┼─┤
6410 LOCATE R+11,C:PRINT "│Alt 8│ȧ│ │ │ │ │ │A│ │ │ │
6415 LOCATE R+12,C:PRINT "├─────┼─┼─┼─┼─┼─┼─┼─┼─┼─┼─┤
6420 LOCATE R+13,C:PRINT "│Alt 7│ª│ │ │º│ │ │ │ │ │ │
6425 LOCATE R+14,C:PRINT "└─────┴─┴─┴─┴─┴─┴─┴─┴─┴─┴─┘
6470 LOCATE R+15,C:PRINT "____Press F4 to continue___";
6490 SWF4%=1:LOCATE IR%,IC%:COLOR 7,0:RETURN
6500 ' ╔═══════════════════════════════════════════════════════════════════╗
6502 ' ║                      CHARACTER CHART [2]                          ║
6504 ' ╚═══════════════════════════════════════════════════════════════════╝
6510 LOCATE R   ,C:PRINT "                            ";
6520 LOCATE R+1 ,C:PRINT "        ┌─────┬───┐         ";
6525 LOCATE R+2 ,C:PRINT "        │Alt A│ æ │         ";
6530 LOCATE R+3 ,C:PRINT "        ├─────┼───┤         ";
6535 LOCATE R+4 ,C:PRINT "        │Alt C│ ç │         ";
6540 LOCATE R+5 ,C:PRINT "        ├─────┼───┤         ";
```

```
6545 LOCATE R+6 ,C:PRINT "          │Alt N│ Ñ │          ";
6550 LOCATE R+7 ,C:PRINT "          ├─────┼───┤          ";
6555 LOCATE R+8 ,C:PRINT "          │Ctr A│ Æ │          ";
6560 LOCATE R+9 ,C:PRINT "          ├─────┼───┤          ";
6565 LOCATE R+10,C:PRINT "          │Ctr C│ Ç │          ";
6570 LOCATE R+11,C:PRINT "          ├─────┼───┤          ";
6585 LOCATE R+12,C:PRINT "          │Ctr N│ ñ │          ";
6590 LOCATE R+13,C:PRINT "          └─────┴───┘          ";
6592 LOCATE R+14,C:PRINT "                               ";
6595 LOCATE IR%,IC%:COLOR 7,0:RETURN
6900 ' ╔══════════════════════════════════════════════════════════╗
6902 ' ║                  WORK AREA WINDOW DEFINITION             ║
6904 ' ╚══════════════════════════════════════════════════════════╝
6905 COLOR 0,7  :LOCATE 6,2 : PRINT "LIN";
6910 COLOR 7,0
6930 LOCATE 2,2 :PRINT "_______WORK  AREA______";
6940 LOCATE 4,2 :PRINT "RULE     :           ";
6950 LOCATE 6,5 :PRINT " Pho D In S X A R T";
6960 RETURN
7000 ' ╔══════════════════════════════════════════════════════════╗
7002 ' ║                       CURSOR   UP                        ║
7004 ' ╚══════════════════════════════════════════════════════════╝
7010 IF CSRLIN-1 < IPR% OR SW.LIN%(CSRLIN-6)=0 THEN 7030
7020 GOSUB 2960:IR%=CSRLIN-1: LIN%=IR%-6 :LOCATE IR%,POS(0)
7030 RETURN
7050 ' ╔══════════════════════════════════════════════════════════╗
7052 ' ║                       CURSOR  DOWN                       ║
7054 ' ╚══════════════════════════════════════════════════════════╝
7060 IF CSRLIN+1 > 21 OR SW.LIN%(LIN%+1)=0 OR SW.LIN%(CSRLIN-6)=0 THEN 7080
7070 GOSUB 2960:IR%=CSRLIN+1: LIN%=IR%-6 :LOCATE IR%,POS(0)
7080 RETURN
7100 ' ╔══════════════════════════════════════════════════════════╗
7102 ' ║                         DELETE                           ║
7104 ' ╚══════════════════════════════════════════════════════════╝
7110 IF SW.LIN%(1)=0 THEN PLAY "L8N10" : GOTO 7270       :'EMPTY LINE
7111 CR%=CSRLIN
7112 IF SW.LIN%(CSRLIN-6)=1 AND SW.LIN%(CSRLIN-5)=0 THEN 7250
7120 FOR I%=CSRLIN TO 21                        :'NUMBER OF PERMISIBLES LINES
7125 LN%=I%-6 : L%=I%-5
7130 IF SW.LIN%(L%)=0 THEN 7200
7150 COLOR 0,7 : LOCATE I%,IC%-4 : PRINT USING "###";LN%;
7155 P1$=PHON$(L%,1)+PHON$(L%,2)+PHON$(L%,3)+" "+PHON$(L%,4)+" "+PHON$(L%,5)
7160 COLOR 7,0 : LOCATE I%,IC%:    PRINT P1$+PHON$(L%,6)+" "+PHON$(L%,7)+" "+
     PHON$(L%,8)+" "+PHON$(L%,9)+" "+PHON$(L%,10)+" "+PHON$(L%,11);
7170 PHON$(LN%,1)=PHON$(L%,1)  : PHON$(LN%,2)=PHON$(L%,2)
7175 PHON$(LN%,3)=PHON$(L%,3)  : PHON$(LN%,4)=PHON$(L%,4)
7180 PHON$(LN%,5)=PHON$(L%,5)  : PHON$(LN%,6)=PHON$(L%,6)
7185 PHON$(LN%,7)=PHON$(L%,7)  : PHON$(LN%,8)=PHON$(L%,8)
7190 PHON$(LN%,9)=PHON$(L%,9)  : PHON$(LN%,10)=PHON$(L%,10)
7195 PHON$(LN%,11)=PHON$(L%,11) : SW.LIN%(LN%)=1 :R1%(LN%)=R1%(L%)
7198 NEXT I%
7200 L%=LN% : SW.LIN%(L%)=0 : GOSUB 7500
7205 COLOR 0,7 : LOCATE I%,IC%-4 : PRINT SPACE$(3);
```

```
7210 COLOR 7,0 : LOCATE I%,IC%   : PRINT SPACE$(18);
7215 IR%=CR%:LOCATE IR%,IC% :LIN%=CSRLIN-6
7220 RETURN
7250 L%=CSRLIN-6:SW.LIN%(L%)=0 : GOSUB 7500
7255 COLOR 0,7 : LOCATE CSRLIN,IC%-4 : PRINT SPACE$(3);
7257 COLOR 7,0 : LOCATE CSRLIN,IC%   : PRINT SPACE$(18);
7260 IF CSRLIN=IPR% THEN LOCATE IPR%,IC% :IR%=IPR%:CPOS%=32:GOTO 7270
7265 IR%=CSRLIN-1:LOCATE IR%,IC% : CPOS% =SCREEN(IR%,IC%)
7266 LOCATE IR%,IC% : LIN%=CSRLIN-6
7270 RETURN
7300 '
7302 '                          INSERT
7304 '
7305 IF SW.LIN%(CSRLIN-6)=0 OR CSRLIN=21 OR SW.LIN%(15)=1 THEN PLAY "L8N10" :
       GOTO 7390
7308 CR%=CSRLIN : L%=CR%-5
7310 WHILE SW.LIN%(L%) :L%=L%+1 : WEND :SW.LIN%(L%)=1
7317 COLOR 0,7 : LOCATE CR%,IC%-4 : PRINT USING "###";CR%-6;
7318 COLOR 7,0 : LOCATE CR%,IC%   : PRINT SPACE$(18);
7320 FOR I%=L% TO CR%-5 STEP -1                  :'NUMBER OF PERMISIBLES LINES
7325 LN%=I%-1 : IR%=I%+6
7330 COLOR 0,7 : LOCATE IR%,IC%-4 : PRINT USING "###";I%;
7335 P1$=PHON$(LN%,1)+PHON$(LN%,2)+PHON$(LN%,3)+" "+PHON$(LN%,4)+" "
        +PHON$(LN%,5)
7340 COLOR 7,0 : LOCATE IR%,IC%:   PRINT P1$+PHON$(LN%,6)+" "+PHON$(LN%,7)+" "
     +PHON$(LN%,8)+" "+PHON$(LN%,9)+" "+PHON$(LN%,10)+" "+PHON$(LN%,11);
7350 PHON$(I%,1)=PHON$(LN%,1) : PHON$(I%,2)=PHON$(LN%,2)
7355 PHON$(I%,3)=PHON$(LN%,3) : PHON$(I%,4)=PHON$(LN%,4)
7360 PHON$(I%,5)=PHON$(LN%,5) : PHON$(I%,6)=PHON$(LN%,6)
7365 PHON$(I%,7)=PHON$(LN%,7) : PHON$(I%,8)=PHON$(LN%,8)
7370 PHON$(I%,9)=PHON$(LN%,9) : PHON$(I%,10)=PHON$(LN%,10)
7375 PHON$(I%,11)=PHON$(LN%,11) : R1%(I%)=R1%(LN%)
7380 NEXT I%
7385 LOCATE CR%,IC% : LIN%=CSRLIN-6 : IR%=CSRLIN
7387 SW.LIN%(LIN%)=0:L%=LIN%: GOSUB 7500
7390 RETURN
7500 '
7502 '                   CLEANNING PHON$ (ARRAY)
7504 '
7510 PHON$(L%,1)=" "          : PHON$(L%,2)=" "
7515 PHON$(L%,3)=" "          : PHON$(L%,4)=" "
7520 PHON$(L%,5)=" "          : PHON$(L%,6)=" "
7525 PHON$(L%,7)=" "          : PHON$(L%,8)=" "
7530 PHON$(L%,9)=" "          : PHON$(L%,10)=" "
7535 PHON$(L%,11)=" "
7540 RETURN
8000 '
8002 '                      REGISTER VALUES
8004 '                           SPEAK
8010 '
8020 FOR I%=1 TO 15
8030 IF SW.LIN%(I%)=0 THEN 8145
8040 FOR J%=15 TO 0 STEP -1 : HX$=HEX$(J%)
```

```
8050 IF HX$=PHON$(I%,6)  THEN P6% =J%
8060 IF HX$=PHON$(I%,8)  THEN P8% =J%
8070 IF HX$=PHON$(I%,9)  THEN P9% =J%
8080 IF HX$=PHON$(I%,10) THEN P10%=J%
8090 NEXT J%
8095 RD1%(I%)=R1%(I%)+(4-VAL(PHON$(I%,4)))*64
8100 R2%(I%)=(VAL(PHON$(I%,5))*16+P6%)*8+VAL(PHON$(I%,7))
8110 R3%(I%)=P8%+P10%*16
8120 R4%(I%)=P9%+VAL(PHON$(I%,11))*16
8140 NEXT I%
8145 RETURN
8150 '
8152 '                        SPEAK
8154 '
8155 GOSUB 8000
8157 FOR I%=1 TO 15 : IF SW.LIN%(I%) = 0 THEN 8180
8160 'CALL SSI263.BUFFER% (RD1%(I%),R2%(I%),R3%(I%),R4%(I%),R5%)
8165 CALL ABSOLUTE (RD1%(I%),R2%(I%),R3%(I%),R4%(I%),R5%,SSI263.BUFFER%)
8170 NEXT I%
8180 RETURN
8200 '
8204 '                     SAVE REGISTERS
8208 '
8210 RULES$(RULE.NUMBER%)=CHR$(LEN(TEXT$))+TEXT$
8250 FOR I%=1 TO 16
8260 IF SW.LIN%(I%)=0 THEN 8290
8270 RULES$(RULE.NUMBER%)=LEFT$(RULES$(RULE.NUMBER%),LEN(RULES$(RULE.NUMBER%)))+
     CHR$(RD1%(I%))+CHR$(R2%(I%))+CHR$(R3%(I%))+CHR$(R4%(I%))
8280 NEXT I%
8290 RULES$(RULE.NUMBER%)=CHR$(I%-1)+RULES$(RULE.NUMBER%)
8295 RETURN
8300 '
8302 '                     LOAD REGISTERS
8306 '
8310 TNC%=ASC(MID$(RULES$(RULE.NUMBER%),2,1))
8315 TEXT$=MID$(RULES$(RULE.NUMBER%),3,TNC%)
8320 RNC%=ASC(MID$(RULES$(RULE.NUMBER%),1,1))
8325 IF RNC%=0 THEN GOSUB 9110:GOTO 8590              :'NO PHONEMS (CLEAN)
8330 REGISTERS$=MID$(RULES$(RULE.NUMBER%),TNC%+3,RNC%*4)
8340 FOR I%=1 TO RNC%
8350 RD1%(I%)=ASC(MID$(REGISTERS$,(I%-1)*4+1,1))
8360 R2%(I%)=ASC(MID$(REGISTERS$,(I%-1)*4+2,1))
8370 R3%(I%)=ASC(MID$(REGISTERS$,(I%-1)*4+3,1))
8380 R4%(I%)=ASC(MID$(REGISTERS$,(I%-1)*4+4,1))
8390 SW.LIN%(I%)=1
8400 R1=RD1%(I%)/64 : R1%=(R1-FIX(R1))*64
8410 PHON$(I%,4)=RIGHT$(STR$(4-FIX(R1)),1)
8420 R1%(I%)=R1%                                      :'PHONEME NUMBER
8430 PHON$(I%,1)=MID$(PHONEMES$,(R1%+1)*3-2,1)
8440 PHON$(I%,2)=MID$(PHONEMES$,(R1%+1)*3-1,1)
8445 PHON$(I%,3)=MID$(PHONEMES$,(R1%+1)*3,1)
8460 R2=R2%(I%)/8 : R2%=(R2-FIX(R2))*8 :R2=FIX(R2)
8470 PHON$(I%,7)=RIGHT$(STR$(R2%),1)
```

```
8480 IF R2 < 16 THEN PHON$(I%,5)="0" : PHON$(I%,6)=HEX$(R2):GOTO 8500
8490 PHON$(I%,5)="1" : PHON$(I%,6)=RIGHT$(HEX$(R2),1)
8500 R3=R3%(I%)/16 : R3%=(R3-FIX(R3))*16
8510 PHON$(I%,10)=HEX$(FIX(R3))
8520 PHON$(I%,8)=HEX$(R3%)
8530 R4=R4%(I%)/16 : R4%=(R4-FIX(R4))*16
8540 PHON$(I%,11)=RIGHT$(STR$(FIX(R4)),1)
8550 PHON$(I%,9)=RIGHT$(HEX$(R4%),1)
8560 NEXT I%
8570 IF I%=17 THEN 8590
8580 FOR J%=I% TO 15 :SW.LIN%(J%)=0:NEXT J%
8590 RETURN
8800 '
8804 '                              NEXT
8808 '
8810 IF RULE.NUMBER%+1 > 999 THEN PLAY "L8N10" :GOTO 8860
8820 GOSUB 8000 : GOSUB 8200                    :'SAVE REGISTERS
8830 RULE.NUMBER%=RULE.NUMBER%+1
8835 LOCATE 4,7:PRINT USING "###";RULE.NUMBER%;:
     LOCATE IPR%,IC%
8840 IF RULE.NUMBER% =< NUMBER.OF.RULES% THEN
        GOSUB 8300: GOSUB 9000 : GOTO 8860      :'LOAD & PRINT REGISTERS
8850 NUMBER.OF.RULES%=NUMBER.OF.RULES%+1 :
     GOSUB 9100                                 :'CLEAN SCREEN & PHON ARRAY
8860 RETURN
8900 '
8902 '                              PRIOR
8904 '
8910 IF RULE.NUMBER%-1 < 1 THEN PLAY "L8N10" :GOTO 8940
8920 GOSUB 8000 : GOSUB 8200                    :'SAVE REGISTERS
8930 RULE.NUMBER%=RULE.NUMBER%-1 :
        GOSUB 8300: GOSUB 9000                  :'LOAD & PRINT REGISTERS
8935 LOCATE 4,7:PRINT USING "###";RULE.NUMBER%;:
     LOCATE IPR%,IC%
8940 RETURN
9000 '
9002 '                    PRINTING PHONEMES
9004 '
9007 LOCATE  4,11:
     PRINT STRING$(12,32);:
     LOCATE  4,11
9008 PRINT TEXT$;
9010 FOR I%= 1 TO 15
9020 IF SW.LIN%(I%)=0 THEN 9060
9030 COLOR 0,7 : LOCATE I%+6,IC%-4 : PRINT USING "###";I%;
9035 COLOR 7,0 : LOCATE I%+6,IC%
9040 PRINT PHON$(I%,1);PHON$(I%,2);PHON$(I%,3);CHR$(32);PHON$(I%,4);CHR$(32);
9045 PRINT PHON$(I%,5);PHON$(I%,6);CHR$(32);PHON$(I%,7);CHR$(32);PHON$(I%,8);
9050 PRINT CHR$(32);PHON$(I%,9);CHR$(32);PHON$(I%,10);CHR$(32);PHON$(I%,11);
9055 GOTO 9070
9060 COLOR 0,7 : LOCATE I%+6,IC%-4 : PRINT "   ";
9065 COLOR 7,0 : PRINT STRING$(19,32);
9070 NEXT I%
```

```
9080 IR%=IPR% : LIN%=1 : LOCATE IR%,IC%
9090 RETURN
9100 '╔══════════════════════════════════════════════════════════╗
9102 '║                  CLEANNING PHONEMES ARRAY                ║
9104 '╚══════════════════════════════════════════════════════════╝
9105 LOCATE  4,11:
     PRINT STRING$(12,32);:
     TEXT$=" "
9110 FOR I%= 1 TO 15
9120 IF SW.LIN%(I%)=0 THEN 9145
9125 SW.LIN%(I%)=0 : L%=I% : GOSUB 7500
9130 COLOR 0,7 : LOCATE I%+6,IC%-4 : PRINT "    ";
9135 COLOR 7,0 : PRINT STRING$(19,32);
9140 NEXT I%
9145 IR%=IPR% : LIN%=1 : LOCATE IR%,IC%
9150 RETURN
10000 '╔═════════════════════════════════════════════════════════╗
10002 '║                                                         ║
10004 '║                  ERROR TRAPPING ROUTINE                 ║
10008 '║                                                         ║
10010 '╚═════════════════════════════════════════════════════════╝
10020 LOCATE 23,3 : PRINT STRING$(75,32)
10025 LOCATE 23,3 : PLAY "L8N10"
10030 PRINT " ERROR -->  ";: COLOR 20,7
10040 IF ERR=7 THEN PRINT "OUT OF MEMORY";:GOTO 10200
10045 IF ERR=14 THEN PRINT "OUT OF STRING SPACE";:GOTO 10200
10050 IF ERR=24 THEN PRINT "DEVICE TIME OUT";:GOTO 10200
10055 IF ERR=25 THEN PRINT "DEVICE FAULT";:GOTO 10200
10060 IF ERR=27 THEN PRINT "OUT OF PAPER";:GOTO 10200
10065 IF ERR=51 THEN PRINT "[ERROR 51] INTERNAL ERROR ";:GOTO 10200
10070 IF ERR=53 THEN PRINT "KNOWLEDGE NOT FOUND";:GOTO 10200
10075 IF ERR=57 THEN PRINT "DEVICE I/O ERROR ";:GOTO 10200
10080 IF ERR=61 THEN PRINT "DISK FULL";:GOTO 10200
10085 IF ERR=64 THEN PRINT "BAD FILE NAME ";:GOTO 10200
10090 IF ERR=67 THEN PRINT "[ERROR 67] TOO MANY FILES ";:GOTO 10200
10095 IF ERR=68 THEN PRINT "DEVICE UNAVAILABLE ";:GOTO 10200
10100 IF ERR=70 THEN PRINT "DISK WRITE PROTECT ";:GOTO 10200
10105 IF ERR=71 THEN PRINT "DISK NOT READY ";:GOTO 10200
10190 PRINT "ERROR-STATE REBOOT THE SYSTEM";
10200 COLOR 7,0:PRINT " Push "+CHR$(17)+"──┘  (Enter) To continue..";
10210 IP$=INKEY$ : IF IP$="" THEN 10210          :'READING FROM KEYBOARD
10220 IF ASC(IP$) <> 13 THEN 10210
10230 LOCATE 23,3:PRINT STRING$(75,32);
10232 IF ERR=53 THEN RESUME 11810
10234 IF ERR=64 AND ERL=11630 THEN RESUME 11500
10236 IF ERR=64 AND ERL=11930 THEN RESUME 11800
10250 RESUME 2000
11000 '╔═════════════════════════════════════════════════════════╗
11002 '║                                                         ║
11004 '║                   USER DEFINED SYMBOLS                  ║
11006 '║                                                         ║
11008 '╚═════════════════════════════════════════════════════════╝
11015 IF SKEY%=60 THEN 11025    : 'IF UDS (F2) N-OR-MORE
```

```
11020 LOCATE 23,3 : PRINT " =   OF  --->";:LOCATE 23,3:GOTO 11030
11025 LOCATE 23,3 : PRINT " =  OR MORE->";:LOCATE 23,3
11030 IP$=INKEY$:IF IP$="" THEN PRINT CHR$(219);:LOCATE CSRLIN,POS(0)-1:
      FOR I%=1 TO SL%:NEXT I%:PRINT CHR$(32);:LOCATE CSRLIN,POS(0)-1:
      FOR I%=1 TO SL%:NEXT I%:GOTO 11030
11032 IF ASC(IP$)=27 THEN 11092 : 'Esc ESCAPE TO MAIN MENU
11035 FOR I%=1 TO LEN(DC$)
11040 IF MID$(DC$,I%,1)=IP$ THEN 11045
11042 NEXT I%
11044 PLAY "L8N10": GOTO 11030
11045 PRINT USING "!";IP$;:
      LOCATE 23,5 : NUDS%=I%
11047 IP$=INKEY$:IF IP$="" THEN PRINT CHR$(219);:LOCATE CSRLIN,POS(0)-1:
      FOR I%=1 TO SL%:NEXT I%:PRINT CHR$(32);:LOCATE CSRLIN,POS(0)-1:
      FOR I%=1 TO SL%:NEXT I%:GOTO 11047
11048 IP%=ASC(IP$)
11050 IF (IP% > 50) OR (IP% <48) THEN PLAY "L8N10":GOTO 11047
11052 PRINT USING "!";IP$;
11055 IF SKEY%=60 THEN UDS$(NUDS%)=IP$+"OR" ELSE UDS$(NUDS%)=IP$+"OF"
11058 COLOR 7,0                   : 'WHITE,BLACK
11059 OLD.TEXT$=TEXT$             : 'SAVING TEXT$
11060 NUM.CHAR=0 :TEXT$=""        : 'NUMBER OF CHARACTERS IN THE UDS TEXT
11065 BUF.LENGHT=60               : 'TEXT BUFFER LENGHT
11070 TR=23: TC=16                : 'TR (TITLE ROW)  : TC (TITLE COLUMN)
11080 LOCATE TR,TC:
      PRINT STRING$(62,32);:
      LOCATE TR,TC
11085 GOSUB 4030                  : 'INPUT UDS TEXT
11090 UDS$(NUDS%)=UDS$(NUDS%)+TEXT$
11092 LOCATE 23,3 : PRINT STRING$(75,32)
11095 LOCATE 1,1:TEXT$=OLD.TEXT$:RETURN
11200 '╔══════════════════════════════════════════════════════╗
11202 '║                      DELETE RULE                     ║
11204 '╚══════════════════════════════════════════════════════╝
11210 IF NUMBER.OF.RULES%-1 < 2 THEN PLAY "L8N10" : GOTO 11295
11220 FOR I%=RULE.NUMBER% TO NUMBER.OF.RULES%-1
11225 RULES$(I%)=RULES$(I%+1)
11230 NEXT I%
11240 RULES$(NUMBER.OF.RULES%)=" "
11250 NUMBER.OF.RULES%=NUMBER.OF.RULES%-1
11255 IF RULE.NUMBER% > NUMBER.OF.RULES% THEN
         RULE.NUMBER%=RULE.NUMBER%-1
11260 GOSUB 8300 : GOSUB 9000               :'LOAD & PRINT REGISTERS
11265 LOCATE 4,7:PRINT USING "###";RULE.NUMBER%;:
      LOCATE IPR%,IC%
11295 RETURN
11300 '╔══════════════════════════════════════════════════════╗
11302 '║                      INSERT RULE                     ║
11304 '╚══════════════════════════════════════════════════════╝
11310 IF NUMBER.OF.RULES%+1 > 999 THEN PLAY "L8N10" : GOTO 11395
11315 GOSUB 8000 : GOSUB 8200               : 'SAVE REGISTERS
11320 FOR I%=NUMBER.OF.RULES%+1 TO RULE.NUMBER% STEP -1
11325 RULES$(I%)=RULES$(I%-1)
```

```
11330 NEXT I%
11340 NUMBER.OF.RULES%=NUMBER.OF.RULES%+1
11350 GOSUB 9100                                : 'CLEAN SCREEN & PHON ARRAY
11365 LOCATE 4,7:PRINT USING "###";RULE.NUMBER%;:
      LOCATE IPR%,IC%
11395 RETURN
11500 '╔═══════════════════════════════════════════════════════════════════════╗
11502 '║                                                                       ║
11504 '║                              SAVE  RULES                              ║
11506 '║                                                                       ║
11508 '╚═══════════════════════════════════════════════════════════════════════╝
11510 COLOR 7,0
11520 LOCATE 23,3 : PRINT "[SAVE] FILE-NAME->";:
11530 NUM.CHAR=0:BUF.LENGHT=8:FILE$=""
11535 TR=23 : TC=21
11540 LOCATE TR,TC:
      PRINT STRING$(12,32);:
      LOCATE TR,TC
11550 SW.INPUT=1                 : 'SWITCHE FOR INPUT DATA (GET IN THE WHILE)
11560 WHILE SW.INPUT
11565 PRINT CHR$(219);:LOCATE TR,POS(0)-1
11570 T$=INKEY$:IF T$="" THEN 11570
11575 T=ASC(T$)
11576 IF T=27 THEN 11680        : 'Esc ESCAPE
11580 IF T<>8 THEN 11590                                :'IF NOT BACKSPACE
11585 IF NUM.CHAR>0 THEN LOCATE TR,POS(0)-1:
       PRINT USING "\ \";"   ";: LOCATE TR,POS(0)-3:
       NUM.CHAR=NUM.CHAR-1: GOTO 11565
11590 IF T=13 THEN SW.INPUT=0:PRINT CHR$(32);:T$="":GOTO 11610
11592 IF NUM.CHAR>BUF.LENGHT THEN PLAY "L12GB-B-4L2" :
       GOTO 11565
11595 IF (T>64 AND T<91) OR (T>96 AND T<123) OR
       (T>45 AND T<59) THEN 11600 ELSE PLAY "L8N10":GOTO 11620
11600 PRINT USING "!";T$;
11610 TEXT$=LEFT$(TEXT$,NUM.CHAR)+T$ : NUM.CHAR=NUM.CHAR+1
11620 WEND
11630 OPEN TEXT$ FOR OUTPUT AS #1
11635 WRITE #1,DC$ : WRITE #1,NUMBER.OF.RULES%
11640 FOR I%=1 TO LEN(DC$)
11650 WRITE #1,UDS$(I%)
11655 NEXT I%
11660 FOR I%= 1 TO NUMBER.OF.RULES%
11664 TNC%=ASC(MID$(RULES$(I%),2,1))                    :'TEXT NUMBER OF CHAR.
11665 TEXT$=MID$(RULES$(I%),3,TNC%)                     :'GETTING THE TEXT
11666 RNC%=ASC(LEFT$(RULES$(I%),1))                     :'RULES NUMBER OF CHAR.
11668 WRITE #1,RNC%,TNC%,TEXT$                          :'LENGHTS & TEXT
11669 IF RNC%=0 THEN 11678
11670 REGISTERS$=MID$(RULES$(I%),TNC%+3,RNC%*4)
11672 FOR J%=1 TO LEN(REGISTERS$)
11674 A%=ASC(MID$(REGISTERS$,J%,1)) : WRITE #1, A%       :'SAVE REGISTERS AS %
11676 NEXT J%
11678 NEXT I% : CLOSE
11680 LOCATE 23,3 : PRINT STRING$(75,32)
```

```
11690 LOCATE 1,1:TEXT$="":RETURN
11800 '╔═══════════════════════════════════════════════════════════════════╗
11802 '║                            LOAD  RULES                            ║
11804 '╚═══════════════════════════════════════════════════════════════════╝
11810 COLOR 7,0
11820 LOCATE 23,3 : PRINT "[LOAD] FILE-NAME->";:
11830 NUM.CHAR=0:BUF.LENGHT=8:TEXT$=""
11835 TR=23 : TC=21
11840 LOCATE TR,TC:
      PRINT STRING$(12,32);:
      LOCATE TR,TC
11850 SW.INPUT=1                    : 'SWITCHE FOR INPUT DATA (GET IN THE WHILE)
11860 WHILE SW.INPUT
11865 PRINT CHR$(219);:LOCATE TR,POS(0)-1
11870 T$=INKEY$:IF T$="" THEN 11870
11875 T=ASC(T$)
11876 IF T=27 THEN 11980           : 'Esc ESCAPE
11880 IF T<>8 THEN 11890                                   :'IF NOT BACKSPACE
11885 IF NUM.CHAR>O THEN LOCATE TR,POS(0)-1:
         PRINT USING "\ \";"   ";: LOCATE TR,POS(0)-3:
         NUM.CHAR=NUM.CHAR-1: GOTO 11865
11890 IF T=13 THEN SW.INPUT=0:PRINT CHR$(32);:T$="":GOTO 11910
11892 IF NUM.CHAR>BUF.LENGHT THEN PLAY "L12GB-B-4L2" :
         GOTO 11865
11895 IF (T>64 AND T<91) OR (T>96 AND T<123) OR
         (T>45 AND T<59) THEN 11900 ELSE 11920
11900 PRINT USING "!";T$;
11910 TEXT$=LEFT$(TEXT$,NUM.CHAR)+T$ : NUM.CHAR=NUM.CHAR+1
11920 WEND
11930 OPEN TEXT$ FOR INPUT AS #1
11935 INPUT #1,DC$ : INPUT #1,NUMBER.OF.RULES%
11940 FOR I%=1 TO LEN(DC$)
11950 INPUT #1,UDS$(I%)                                    :'SAVING USER DEF. SYM.
11955 NEXT I%
11960 FOR I%= 1 TO NUMBER.OF.RULES%
11961 INPUT #1,RNC%,TNC%,TEXT$
11963 RULES$(I%)=CHR$(RNC%)+CHR$(TNC%)+TEXT$
11965 IF RNC%=0 THEN 11978
11968 FOR J%=1 TO RNC%*4
11970 INPUT #1,A%                                          :'LOAD REGISTERS AS %
11972 RULES$(I%)=RULES$(I%)+CHR$(A%)
11974 NEXT J%
11978 NEXT I% : CLOSE
11980 LOCATE 23,3 : PRINT STRING$(75,32)
11990 TEXT$="": RULE.NUMBER%=1 :GOSUB 8300 : GOSUB 9000 :'PRINT FIRST RULE
11995 RETURN
12000 '╔═══════════════════════════════════════════════════════════════════╗
12002 '║                DISPLAY USER DEFINED SYMBOLS (UDS)                 ║
12004 '╚═══════════════════════════════════════════════════════════════════╝
12010 COLOR 7,1 :C=52 :R=2
12020 LOCATE 2,   C:PRINT "USER DEFINED SYMBOLS┌────┐";
12055 LOCATE R+1 ,C:PRINT "                    │ #  │";
12060 LOCATE R+2 ,C:PRINT "                    ├────┤";
```

```
12070 LOCATE R+3 ,C:PRINT "                   │  $  │";
12075 LOCATE R+4 ,C:PRINT "                   ├─────┤";
12080 LOCATE R+5 ,C:PRINT "                   │  %  │";
12085 LOCATE R+6 ,C:PRINT "                   ├─────┤";
12090 LOCATE R+7 ,C:PRINT "                   │  &  │";
12095 LOCATE R+8 ,C:PRINT "                   ├─────┤";
12100 LOCATE R+9 ,C:PRINT "                   │  *  │";
12105 LOCATE R+10,C:PRINT "                   ├─────┤";
12110 LOCATE R+11,C:PRINT "                   │  +  │";
12115 LOCATE R+12,C:PRINT "                   ├─────┤";
12120 LOCATE R+13,C:PRINT "                   │  ^  │";
12125 LOCATE R+14,C:PRINT "                   ├─────┤";
12130 LOCATE R+15,C:PRINT "                   │  :  │";
12135 LOCATE R+16,C:PRINT "                   ├─────┤";
12140 LOCATE R+17,C:PRINT "                   │  @  │";
12145 LOCATE R+18,C:PRINT "  Esc -for-> Exit   └─────┘";
12150 CRF%=R+1 :COLOR 1,7
12160 COLOR 4,7:LOCATE CRF%,C:
      PRINT "   DISPLAY -UDS-->";:
      COLOR 7,1: PRINT CHR$(16);
12200 IP$=INKEY$ : IF IP$="" THEN 12200          :'READING FROM KEYBOARD
12210 IP%=ASC(IP$)                               :'ASCII IP$
12220 IF ASC(MID$(IP$,1,1))<>0 THEN 12260        :'COMMON ASCII VALUE
12230 SKEY%=ASC(MID$(IP$,2,1))                   :'SPECIAL KEY COMBINATIONS
12240 IF SKEY%=80 THEN GOSUB 12300:GOTO 12200    :'CURSOR DOWN
12250 IF SKEY%=72 THEN GOSUB 12350:GOTO 12200
         ELSE GOTO 12200                         :'CURSOR UP
12260 IF IP% = 13 THEN GOSUB 12400               :'CARRIAGE RETURN
12270 IF IP%<> 27 THEN 12200                     :'MUST BE Escape
12272 LOCATE CRF%,C:COLOR 7,1
12274 PRINT STRING$(19,32)
12275 COLOR 7,0:LOCATE 23,3:PRINT STRING$(75,32):'CLEAN WORK BOX AREA
12280 LOCATE 1,1:RETURN
12300 LOCATE CRF%,C:COLOR 7,1
12310 PRINT STRING$(19,32)
12320 CRF%=CRF%+2
12325 IF CRF%>R+17 THEN CRF%=R+1
12330 COLOR 4,7:LOCATE CRF%,C:
      PRINT "   DISPLAY -UDS-->";:
      COLOR 7,1: PRINT CHR$(16);
12340 RETURN
12350 LOCATE CRF%,C:COLOR 7,1
12360 PRINT STRING$(19,32)
12370 CRF%=CRF%-2
12375 IF CRF%<R+1 THEN CRF%=R+17
12380 COLOR 4,7:LOCATE CRF%,C:
      PRINT "   DISPLAY -UDS-->";:
      COLOR 7,1: PRINT CHR$(16);
12390 RETURN
12400 LOCATE 23,3:COLOR 7,0:PRINT STRING$(75,32);
12405 CUDS%=(CRF%-R+1)/2                          :'CURRENT UDS GIVEN CRF%
12410 LOCATE 23,3: PRINT MID$(DC$,CUDS%,1);
12412 IF UDS$(CUDS%)="" THEN PRINT " --> THIS UDS IS EMPTY ";:GOTO 12440
```

```
12415 PRINT "="+LEFT$(UDS$(CUDS%),1);
12420 IF MID$(UDS$(CUDS%),2,2)="OR"
          THEN PRINT " OR MORE->";
          ELSE PRINT "  OF  --->";
12430 PRINT RIGHT$(UDS$(CUDS%),LEN(UDS$(CUDS%))-3)
12440 RETURN
12500 '╔══════════════════════════════════════════════════════════════════════╗
12502 '║                                NUMBER                                ║
12505 '╚══════════════════════════════════════════════════════════════════════╝
12510 COLOR 7,0
12520 LOCATE 23,3 : PRINT "[RULE] NUMBER --> ";:
12530 NUM.CHAR=0:BUF.LENGHT=3:NUMT$=""
12535 TR=23 : TC=21
12540 LOCATE TR,TC:
      PRINT STRING$(12,32);:
      LOCATE TR,TC
12550 SW.INPUT=1                   : 'SWITCHE FOR INPUT DATA (GET IN THE WHILE)
12560 WHILE SW.INPUT
12565 PRINT CHR$(219);:LOCATE TR,POS(0)-1
12570 T$=INKEY$:IF T$="" THEN 12570
12575 T=ASC(T$)
12576 IF T=27 THEN 12785         : 'Esc ESCAPE
12580 IF T<>8 THEN 12590                                   :'IF NOT BACKSPACE
12585 IF NUM.CHAR>0 THEN LOCATE TR,POS(0)-1:
          PRINT USING "\ \";"   ";: LOCATE TR,POS(0)-3:
          NUM.CHAR=NUM.CHAR-1: GOTO 12565
12590 IF T=13 THEN SW.INPUT=0:PRINT CHR$(32);:T$="":GOTO 12610
12592 IF NUM.CHAR>BUF.LENGHT THEN PLAY "L12GB-B-4L2" :
          GOTO 12565
12595 IF (T>47 AND T<58)
          THEN 12600 ELSE PLAY "L8N10":GOTO 12620
12600 PRINT USING "!";T$;
12610 NUMT$=LEFT$(NUMT$,NUM.CHAR)+T$ : NUM.CHAR=NUM.CHAR+1
12620 WEND:NUM%=0
12650 FOR I%=1 TO LEN(NUMT$)
12660      NUM%=NUM%+VAL(RIGHT$(NUMT$,1))*10^(I%-1)
12665      NUMT$=LEFT$(NUMT$,LEN(NUMT$)-1)
12670 NEXT I%
12680 IF NUM%=<NUMBER.OF.RULES% THEN 12750
12690 LOCATE 23,3 : PLAY "L8N10"
12700 PRINT " ERROR -->  ";: COLOR 20,7
12710 PRINT USING "RULE NUMBER: ### DOES NOT EXIST";NUM%;
12720 COLOR 7,0:PRINT " Push "+CHR$(17)+"──┘ (Enter) To continue..";
12730 IP$=INKEY$ : IF IP$="" THEN 12730            :'READING FROM KEYBOARD
12740 IF ASC(IP$) <> 13 THEN 12730
12745 LOCATE 23,3:PRINT STRING$(75,32);:GOTO 12500
12750 '╔══════════════════════════════════════════════════════════════════════╗
12752 '║                            RULE (NUMBER)                             ║
12755 '╚══════════════════════════════════════════════════════════════════════╝
12760 GOSUB 8000 : GOSUB 8200                       :'SAVE REGISTERS
12770 RULE.NUMBER%=NUM%
12775 LOCATE 4,7:PRINT USING "###";RULE.NUMBER%;:
      LOCATE IPR%,IC%
```

```
12780 GOSUB 8300 : GOSUB 9000                   :'LOAD & PRINT REGISTERS
12785 LOCATE 23,3:PRINT STRING$(75,32);
12790 LOCATE IPR%,IC%:RETURN
20000 PRINT "       "+CHR$(201)+STRING$(70,CHR$(205))+CHR$(187)
20010 PRINT "       "+CHR$(186)+STRING$(70,CHR$(32 ))+CHR$(186)
20020 PRINT "       "+CHR$(186)+STRING$(70,CHR$(32 ))+CHR$(186)
20030 PRINT "       "+CHR$(186)+STRING$(70,CHR$(32 ))+CHR$(186)
20040 PRINT "       "+CHR$(200)+STRING$(70,CHR$(205))+CHR$(188)
```

APPENDIX D
THE PARSER PROGRAM

Language: BASIC
Compiler: IBM BASIC Compiler Version 2.00

```
2 '
4 '
6 '                            PARSER (RULE-APPLIER)
8 '
10 '
11 ON ERROR  GOTO 12000                                        :'ERROR TRAP. ROUTINE
12 KEY OFF
14 FOR I%=1 TO 10:KEY I%,"":NEXT I%                            :'DISABLES FUN KEYS
15 CLS :SCREEN 1,0
100 DIM UDS$(9,30),R.PREF$(500),R.BODY$(500),R.SFIX$(500),R.PHON$(500),
        INDEX.RULES%(255),R1%(525),R2%(525),R3%(525),R4%(525)
120 DIM PICTURE%(525)
130 DIM A1%(250),A2%(250),A3%(250),A4%(250),A5%(250),A6%(250),A7%(250),
        A8%(250),A9%(250),A10%(250),A11%(250),A12%(250),A13%(250),A14%(250),
        A15%(250),A16%(250),A17%(250),A18% (250)
140 DIM T1%(852),T2%(852),T3%(852),T4%(852),T5%(852),T6%(852),T7%(852),
        T8%(852),T9%(852),T10%(852),T11%(852),T12%(852),T13%(852),T14%(852),
        T15%(852),T16%(852),T17%(852),T18% (852)
200 '
202 '                              PARAMETERS
204 '
205 GOSUB 13000                           : 'LOAD FACE AND RULES
210 R5%=232
216 XT=185:YT=101                         : 'POSITION OF THE MOUTH (PUT STATEMENT
218 ERA$=STRING$(38," ")                  : 'ERASER
300 '
302 '
304 '                          LOCATE SONIX KERNEL
306 '
308 '
350  X%=&H180                                        :'START OF USER INTERUPT VECTOR
360  WHILE X%>0 AND X%<&H1A0
370    DEF SEG= 0: SONIX.SEG#= PEEK(X%+2) + PEEK(X%+3)*256
380    X%=X%+4
390    DEF SEG= SONIX.SEG#
400 '
402 '                      CHECK FOR SONIX'S SIGNATURE
404 '
410    IF PEEK(&H100)=83 AND PEEK(&H101)=79 AND PEEK(&H105)=170 THEN X%=0
420  WEND
430  WHILE NOT X%=0: 'If can't find SONIX then exit
440    PRINT "CANNOT LOCATE THE SONIX KERNEL!"
450    PRINT "RETURN TO DOS AND LOAD SONIX BEFORE CONTINUING": END
460  WEND
470 '
472 '                     DEFINE SONIX-88 ENTRY POINTS
```

```
474 ' ╚══════════════════════════════════════════════════════════════════════╝
510 'SONIX Function                   Form of usage..........................
540  SSI263.BUFFER%= 306        : 'CALL SSI263.BUFFER (R1%,R2%,R3%,R4%,R5%)
550  SSI263.DIRECT%= 309        : 'CALL SSI263.DIRECT (R1%,R2%,R3%,R4%,R5%)
570  SET.GLOBALS%=   318        : 'CALL SET.GLOBALS (MR%,MA%,MI%,MF%)
590  INIT.SPEECH%=   327        : 'CALL INIT.SPEECH
600  QUIET%=         330        : 'CALL QUIET
610  WAIT.EMPTY%=    333        : 'CALL WAIT.EMPTY
630  DEF SEG= SONIX.SEG#
1000 ' ╔══════════════════════════════════════════════════════════════════════╗
1002 ' ║                           MAIN PROCEDURE                             ║
1004 ' ╚══════════════════════════════════════════════════════════════════════╝
1020 GOSUB 9000          : 'Input text by keyboard
1030 GOSUB 3000          : 'Parse "Text to speech"
1040 GOTO 1020
1300 ' ╔══════════════════════════════════════════════════════════════════════╗
1302 ' ║                       VOCAL TRACT ANIMATION                          ║
1304 ' ╚══════════════════════════════════════════════════════════════════════╝
2400 ' ╔══════════════════════════════════════════════════════════════════════╗
2402 ' ║                               MOUTHS                                 ║
2404 ' ╚══════════════════════════════════════════════════════════════════════╝
2410 GET (  0,  0)-( 59, 30),A1%
2415 GET ( 60,  0)-(119, 30),A2%
2420 GET (120,  0)-(179, 30),A3%
2425 GET (180,  0)-(239, 30),A4%
2430 GET (240,  0)-(299, 30),A5%
2435 GET (  0, 30)-( 59, 60),A6%
2440 GET ( 60, 30)-(119, 60),A7%
2445 GET (120, 30)-(179, 60),A8%
2450 GET (180, 30)-(239, 60),A9%
2455 GET (240, 30)-(299, 60),A10%
2460 GET (  0, 60)-( 59, 90),A11%
2465 GET ( 60, 60)-(119, 90),A12%
2470 GET (120, 60)-(179, 90),A13%
2475 GET (180, 60)-(239, 90),A14%
2480 GET (240, 60)-(299, 90),A15%
2485 GET (  0, 90)-( 59,120),A16%
2490 GET ( 60, 90)-(119,120),A17%
2495 GET (120, 90)-(179,120),A18% : RETURN
2500 ' ╔══════════════════════════════════════════════════════════════════════╗
2502 ' ║                              TONGUES                                 ║
2504 ' ╚══════════════════════════════════════════════════════════════════════╝
2510 GET ( 20,  0)-(120, 32),T1%
2515 GET (170,  0)-(270, 32),T2%
2520 GET ( 20, 40)-(120, 72),T3%
2525 GET (170, 40)-(270, 72),T4%
2530 GET ( 20, 80)-(120,112),T5%
2535 GET (170, 80)-(270,112),T6%
2540 GET ( 20,120)-(120,152),T7%
2545 GET (170,120)-(270,152),T8%
2550 GET ( 20,160)-(120,192),T9%
2555 GET (170,160)-(270,192),T10% : RETURN
2560 GET ( 20,  0)-(120, 32),T11%
```

```
2565 GET (170,  0)-(270, 32),T12%
2570 GET ( 20, 40)-(120, 72),T13%
2575 GET (170, 40)-(270, 72),T14%
2580 GET ( 20, 80)-(120,112),T15%
2585 GET (170, 80)-(270,112),T16%
2590 GET ( 20,120)-(120,152),T17%
2595 GET (170,120)-(270,152),T18% : RETURN
3000 '
3002 '
3004 '                          TEXT-TO-SPEECH
3006 '
3008 '
3100 L.TEXT%=LEN(TEXT$) : IF L.TEXT%=0 THEN 3750
3102 TEXT$=" "+TEXT$+" " : N%=0 :L.TEXT%=L.TEXT%+2
3105 SW.SPEAK%=1 :I%=1
3110 WHILE SW.SPEAK%
3115 LETTER$=MID$(TEXT$,I%,1)
3200 KOR%=INDEX.RULES%(ASC(LETTER$))                       :'KIND OF RULE TO APPLY
3201 IF KOR%>255 THEN 3220
3202 '
3204 '
3205 '
3210 IF R.BODY$(KOR%)=MID$(TEXT$,I%,LEN(R.BODY$(KOR%)))
       THEN 3250
3220 KOR%=KOR%+1 : IF KOR%>255 THEN I%=I%+1: GOTO 3345
                   ELSE GOTO 3210 :'VALIDATE NON MATCH
3250 '
3252 '
3254 '
3260 FOR J%= 1 TO LEN(R.PHON$(KOR%)) STEP 4
3270      N%=N%+1:
          R1%(N%)=ASC(MID$(R.PHON$(KOR%),J%,1)):
          R1=R1%(N%)/64
3272      PICTURE%(N%)=(R1-FIX(R1))*64
3280      R2%(N%)=ASC(MID$(R.PHON$(KOR%),J%+1,1))
3290      R3%(N%)=ASC(MID$(R.PHON$(KOR%),J%+2,1))
3300      R4%(N%)=ASC(MID$(R.PHON$(KOR%),J%+3,1))
3310 NEXT J%
3340 I%=I%+LEN(R.BODY$(KOR%))
3345 IF I%>L.TEXT% THEN SW.SPEAK%=0
3350 WEND
3500 '
3502 '                          ANIMATION
3504 '
3510 FOR I%=1 TO N%
3520 'CALL SSI263.BUFFER% (R1%(I%),R2%(I%),R3%(I%),R4%(I%),R5%)
3530 CALL ABSOLUTE (R1%(I%),R2%(I%),R3%(I%),R4%(I%),R5%,SSI263.BUFFER%)
3550 'IF PICTURE%(I%)=0  THEN CALL WAIT.EMPTY% : GOTO 3600
3552 IF PICTURE%(I%)=0  THEN CALL ABSOLUTE (WAIT.EMPTY%): GOTO 3600
3560 IF PICTURE%(I%)>32 THEN PICTURE%=PICTURE%(I%)-32: GOTO 3580
3565 '02-,-03-,-04-,-05-,-06-,-07-,-08-,-09-,-10-,-11-,-12-,-13-,-14-,-15-,-16-
3570 ON PICTURE%(I%) GOSUB    4030,
     4050,4030,4070,4090,4030,4050,4050,4110,4050,4050,4130,4130,4150,4150,4170,
     4170,4170,4190,4190,4190,4190,4190,4150,4150,4150,4150,4210,4210,4210,4210,
     4230
3572 '17-,-18-,-19-,-20-,-21-,-22-,-23-,-24-,-25-,-26-,-27-,-28-,-29-,-30-,-31-
3575 GOTO 3600
3576 '33-,-34-,-35-,-36-,-37-,-38-,-39-,-40-,-41-,-42-,-43-,-44-,-45-,-46-,-47-
3580 ON PICTURE% GOSUB
     4230,4230,4250,4270,4290,4310,4270,4290,4330,4500,4310,4350,4330,4500,4370,
     4370,4390,4390,4410,4410,4370,4370,4270,4290,4430,4500,4500,4500,4500,4500,
     4500
3582 '48-,-49-,-50-,-51-,-52-,-53-,-54-,-55-,-56-,-57-,-58-,-59-,-60-,-61-,-62-
3600 '
3610 IF SWF1%=0 THEN 3700
3620 A$=INKEY$: IF A$="" THEN 3620
3621 IF ASC(MID$(A$,1,1))<>0 THEN 3700
3622 SKEY%=ASC(MID$(A$,2,1))
3623 IF SKEY%=59 AND SWF1%=1 THEN SWF1%=0 : LOCATE 23,39:PRINT CHR$(32);:
        GOTO 3610
3624 IF SKEY%=59 AND SWF1%=0 THEN SWF1%=1 : LOCATE 23,39:PRINT CHR$(95);:
        GOTO 3610
3700 NEXT I% :GOSUB 4010
3710 A$=INKEY$: IF A$="" THEN 3710
3720 IF ASC(MID$(A$,1,1))<>0 THEN 3740
3725 SKEY%=ASC(MID$(A$,2,1))
3730 IF SKEY%=59 AND SWF1%=1 THEN SWF1%=0 : LOCATE 23,39:PRINT CHR$(32);:
        GOTO 3710
3735 IF SKEY%=59 AND SWF1%=0 THEN SWF1%=1 : LOCATE 23,39:PRINT CHR$(95);:
        GOTO 3710
3740 IF ASC(A$)<>13 THEN 3500
3750 RETURN
4000 ' ╔═══════════════════════════════════════════════════════════════════════╗
4002 ' ║              ASSIGNATION OF MOUTHS AND TONGUES TO PHONEMES            ║
4005 ' ╚═══════════════════════════════════════════════════════════════════════╝
4010 PUT (XN,YN),A1%,PSET                               :' PA
4020 PUT (XT,YT),T1%,PSET : RETURN
4030 PUT (XN,YN),A6%,PSET                               :'  E,  Y, IE
4040 PUT (XT,YT),T10%,PSET: RETURN
4050 PUT (XN,YN),A10%,PSET                              :' E1,  I,  A, EH,EH1
4060 PUT (XT,YT),T6%,PSET : RETURN
4070 PUT (XN,YN),A6%,PSET                               :' Y1 *
4080 PUT (XT,YT),T10%,PSET: RETURN                      :' PROV. TONGUE
4090 PUT (XN,YN),A10%,PSET                              :' AY *
4100 PUT (XT,YT),T10%,PSET: RETURN                      :' PROV. TONGUE
4110 PUT (XN,YN),A9%,PSET                               :' A1
4120 PUT (XT,YT),T6%,PSET : RETURN
4130 PUT (XN,YN),A4%,PSET                               :' AE,AE1 *
4140 PUT (XT,YT),T2%,PSET : RETURN
4150 PUT (XN,YN),A2%,PSET                               :' AH,AH1, UH,UH1,UH2,UH3
4160 PUT (XT,YT),T2%,PSET : RETURN
4170 PUT (XN,YN),A14%,PSET                              :' AW,  O, OU
4180 PUT (XT,YT),T9%,PSET : RETURN
4190 PUT (XN,YN),A18%,PSET                              :' OO, IU,  U, U1
4200 PUT (XT,YT),T4%,PSET : RETURN
4210 PUT (XN,YN),A13%,PSET                              :' ER,  R, R1, R2
```

```
4220 PUT (XT,YT),T15%,PSET: RETURN
4230 PUT (XN,YN),A12%,PSET                     :'  L, L1, LF
4240 PUT (XT,YT),T12%,PSET: RETURN
4250 PUT (XN,YN),A18%,PSET                     :'  W
4260 PUT (XT,YT),T8%,PSET : RETURN
4270 PUT (XN,YN),A3%,PSET                      :'  B,  P,  M
4280 PUT (XT,YT),T13%,PSET: RETURN
4290 PUT (XN,YN),A5%,PSET                      :'  D,  T,  N * (DENT ALV)
4300 PUT (XT,YT),T12%,PSET: RETURN             :'  PROV. TONGUE
4310 PUT (XN,YN),A8%,PSET                      :' KV,HVC
4320 PUT (XT,YT),T8%,PSET : RETURN
4330 PUT (XN,YN),A4%,PSET                      :'  K,HFCT (*)
4340 PUT (XT,YT),T4%,PSET : RETURN
4350 PUT (XN,YN),A9%,PSET                      :' HF
4360 PUT (XT,YT),T9%,PSET : RETURN
4370 PUT (XN,YN),A17%,PSET                     :'  Z,  S,THV, TH
4380 PUT (XT,YT),T18%,PSET: RETURN
4390 PUT (XN,YN),A16%,PSET                     :'  J,SCH * PROV. TONGUE
4400 PUT (XT,YT),T16%,PSET: RETURN
4410 PUT (XN,YN),A7%,PSET                      :'  V,  F
4420 PUT (XT,YT),T17%,PSET: RETURN
4430 PUT (XN,YN),A11%,PSET                     :' NG
4440 PUT (XT,YT),T4%,PSET : RETURN             :' PROV. TONGUE &  OUTH
4500 'UNASSIGNED
4510 RETURN                                    :' HV, HN, :A,:OH, :U,:UH,E2,LB
9000 '
9010 '
9020 '                          READ TEXT FROM KEYBOARD
9030 '
9040 '
9060 NUM.CHAR=0 :TEXT$=""       : 'NUMBER OF CHARACTERS IN THE TITLE TEXT
9070 BUF.LENGHT=35              : 'TEXT BUFFER LENGHT
9080 TR=23: TC=2                : 'TR (TITLE ROW)  : TC (TITLE COLUMN)
9090 LOCATE TR,TC:
     PRINT STRING$(BUF.LENGHT,32);:
     LOCATE TR,TC
9100 GOSUB 9120                 : 'INPUT TITLE TEXT
9110 LOCATE 1,1     : RETURN    : 'END OF TEXT INPUT PROCEDURE
9120 '
9130 '
9140 '  KEYBOARD SELECTION AND ASSIGNATION OF SPECIAL SYMBOLS AND ACCENTS
9150 '
9160 '
9170 SW.INPUT=1                 : 'SWITCHE FOR INPUT DATA (GET IN THE WHILE)
9180 WHILE SW.INPUT
9190 PRINT CHR$(219);:LOCATE TR,POS(0)-1
9200 T$=INKEY$:IF T$="" THEN 9200
9210 T=ASC(T$)
9220 IF T<>8 THEN 9240                                 :'IF NOT BACKSPACE
9230 IF NUM.CHAR>0 THEN LOCATE TR,POS(0)-1:
        PRINT USING "\ \";"   ";: LOCATE TR,POS(0)-3:
        NUM.CHAR=NUM.CHAR-1: GOTO 9190
9240 IF T=13 THEN SW.INPUT=0:PRINT CHR$(32);:T$="":GOTO 10170
```

```
9250 IF NUM.CHAR>=BUF.LENGHT THEN PLAY "L12GB-B-4L2" :
     GOTO 9190
9255 IF T=27 THEN WIDTH 80: SCREEN 0:CLS:END
9260 IF T>64 AND T<91 THEN T=T+32 : T$=CHR$(T)          :'ASSURE LOWER CASE
9270 '
9280 '    ACCENT    :
9290 '
9300 IF ACCENT1 = 0 THEN 9350
9310    ACCENT1=0
9320    IF T=97  THEN T$=CHR$(166) : SW=1 : GOTO 9340
9330    IF T=111 THEN T$=CHR$(167) : SW=1
9340 IF SW=0 THEN 9920 ELSE SW=0: GOTO 10130
9350 '
9360 '    ACCENT    :
9370 '
9380 IF ACCENT2 = 0 THEN 9430
9390    ACCENT2=0
9400    IF T=97  THEN T$=CHR$(134) : SW=1 : GOTO 9420
9410    IF T=65  THEN T$=CHR$(143) : SW=1
9420 IF SW=0 THEN 9920 ELSE SW=0: GOTO 10130
9430 '
9440 '    ACCENT    :
9450 '
9460 IF ACCENT3 = 0 THEN 9540
9470    ACCENT3=0
9480    IF T=97  THEN T$=CHR$(131) : SW=1 : GOTO 9530
9490    IF T=101 THEN T$=CHR$(136) : SW=1 : GOTO 9530
9500    IF T=105 THEN T$=CHR$(140) : SW=1 : GOTO 9530
9510    IF T=111 THEN T$=CHR$(147) : SW=1 : GOTO 9530
9520    IF T=117 THEN T$=CHR$(150) : SW=1
9530 IF SW=0 THEN 9920 ELSE SW=0: GOTO 10130
9540 '
9550 '    ACCENT    :
9560 '
9570 IF ACCENT4 = 0 THEN 9650
9580    ACCENT4=0
9590    IF T=97  THEN T$=CHR$(133) : SW=1 : GOTO 9530
9600    IF T=101 THEN T$=CHR$(138) : SW=1 : GOTO 9530
9610    IF T=105 THEN T$=CHR$(141) : SW=1 : GOTO 9530
9620    IF T=111 THEN T$=CHR$(149) : SW=1 : GOTO 9530
9630    IF T=117 THEN T$=CHR$(151) : SW=1
9640 IF SW=0 THEN 9920 ELSE SW=0: GOTO 10130
9650 '
9660 '    ACCENT    :
9670 '
9680 IF ACCENT5= 0 THEN 9800
9690    ACCENT5=0
9700    IF T=97  THEN T$=CHR$(132) : SW=1 : GOTO 9790
9710    IF T=101 THEN T$=CHR$(137) : SW=1 : GOTO 9790
9720    IF T=105 THEN T$=CHR$(139) : SW=1 : GOTO 9790
9730    IF T=111 THEN T$=CHR$(148) : SW=1 : GOTO 9790
9740    IF T=117 THEN T$=CHR$(129) : SW=1 : GOTO 9790
9750    IF T=121 THEN T$=CHR$(152) : SW=1 : GOTO 9790
```

```
9760    IF T=65  THEN T$=CHR$(142) : SW=1 : GOTO 9790
9770    IF T=79  THEN T$=CHR$(153) : SW=1 : GOTO 9790
9780    IF T=85  THEN T$=CHR$(154) : SW=1
9790 IF SW=0 THEN 9920 ELSE SW=0: GOTO 10130
9800 '
9810 '   ACCENT   :
9820 '
9830 IF ACCENT6 = 0 THEN 9920
9840    ACCENT6=0
9850    IF T=97  THEN T$=CHR$(160) : SW=1 : GOTO 9910
9860    IF T=101 THEN T$=CHR$(130) : SW=1 : GOTO 9910
9870    IF T=105 THEN T$=CHR$(161) : SW=1 : GOTO 9910
9880    IF T=111 THEN T$=CHR$(162) : SW=1 : GOTO 9910
9890    IF T=117 THEN T$=CHR$(163) : SW=1 : GOTO 9910
9900    IF T=69  THEN T$=CHR$(144) : SW=1 : GOTO 9910
9910 IF SW=0 THEN 9920 ELSE SW=0: GOTO 10130
9920 IF (T>34 AND T<123) OR T=32 OR T=33 THEN 10130
9930 IF ASC(MID$(T$,1,1))<>0 THEN GOTO 10060
9940 SPECIAL.KEY=ASC(MID$(T$,2,1))                        :'EXTENDED CODES
9950 IF SPECIAL.KEY<126 OR SPECIAL.KEY>131 THEN 10030
9960 ON SPECIAL.KEY-125 GOTO 9970,9980,9990,10000,10010,10020
9970 ACCENT1=1 : GOTO 10180
9980 ACCENT2=1 : GOTO 10180
9990 ACCENT3=1 : GOTO 10180
10000 ACCENT4=1 : GOTO 10180
10010 ACCENT5=1 : GOTO 10180
10020 ACCENT6=1 : GOTO 10180
10030 IF SPECIAL.KEY=49 THEN T$=CHR$(164) : GOTO 10130    :' ñ . Alt N
10040 IF SPECIAL.KEY=46 THEN T$=CHR$(135) : GOTO 10130    :' ç . Alt C
10050 IF SPECIAL.KEY=30 THEN T$=CHR$(145) : GOTO 10130    :' æ . Alt A
10060 IF T=14 THEN T$=CHR$(165) : GOTO 10130              :' Ñ . Ctrl N
10070 IF T=3  THEN T$=CHR$(128) : GOTO 10130              :' Ç . Ctrl C
10080 IF T=1  THEN T$=CHR$(146) : GOTO 10130              :' Æ . Ctrl A
10090 '
10100 '             CONCLUDE SELECTION
10110 '
10120 GOTO 10180                                          :' WEND
10130 '
10140 '        PRINT CHARACTER & UPDATE COUNTERS
10150 '
10160 PRINT USING "!";T$;
10170 TEXT$=LEFT$(TEXT$,NUM.CHAR)+T$ : NUM.CHAR=NUM.CHAR+1
10180 WEND
10190 RETURN
11000 '
11010 '                          LOAD  RULES
11020 '
11040 LOCATE 23,3 : PRINT "[LOAD] KNOWLEDGE :";:
11050 NUM.CHAR=0:BUF.LENGHT=8:TEXT$=""
11060 TR=23 : TC=21
11070 LOCATE TR,TC:
      PRINT STRING$(12,32);:
      LOCATE TR,TC
```

```
11080 SW.INPUT=1                     : 'SWITCHE FOR INPUT DATA (GET IN THE WHILE)
11090 WHILE SW.INPUT
11100 PRINT CHR$(219);:LOCATE TR,POS(0)-1
11110 T$=INKEY$:IF T$="" THEN 11110
11120 T=ASC(T$)
11130 IF T=27 THEN 11480           : 'Esc ESCAPE
11140 IF T<>8 THEN 11160                                      :'IF NOT BACKSPACE
11150 IF NUM.CHAR>0 THEN LOCATE TR,POS(0)-1:
          PRINT USING "\ \";"   ";: LOCATE TR,POS(0)-3:
          NUM.CHAR=NUM.CHAR-1: GOTO 11100
11160 IF T=13 THEN SW.INPUT=0:PRINT CHR$(32);:T$="":GOTO 11200
11170 IF NUM.CHAR>BUF.LENGHT THEN PLAY "L12GB-B-4L2" :
          GOTO 11100
11180 IF (T>64 AND T<91) OR (T>96 AND T<123) OR
          (T>45 AND T<59) THEN 11190 ELSE 11210
11190 PRINT USING "!";T$;
11200 TEXT$=LEFT$(TEXT$,NUM.CHAR)+T$ : NUM.CHAR=NUM.CHAR+1
11210 WEND
11220 OPEN TEXT$ FOR INPUT AS #1
11225 LOCATE 23,3:PRINT STRING$(30,32);
11230 INPUT #1,DC$ : INPUT #1,NUMBER.OF.RULES%
11240 L.DC%=LEN(DC$):FOR I%=1 TO L.DC%
11250 INPUT #1,UDS$ : L%=1
11252 IF LEN(UDS$)=0 THEN 11262
11254 FOR J%=1 TO LEN(UDS$):IF L%>30 THEN 11262
11255 SCHR$=MID$(UDS$,J%,1)                                   :'SINGLE UDS CHARACTER
11256 IF J%<4 THEN UDS$(I%,0)=UDS$(I%,0)+SCHR$:GOTO 11260
11258 IF SCHR$ = "," THEN L%=L%+1 : GOTO 11260
11259 UDS$(I%,L%)=UDS$(I%,L%)+SCHR$
11260 NEXT J%
11261 UDS$(I%,0)=CHR$(L%)+UDS$(I%,0)
11262 NEXT I%
11270 FOR I%= 1 TO NUMBER.OF.RULES%
11272 INPUT #1,RNC%,TNC%,TEXT$
11275 IF TNC%=0 THEN 11305
11280 SWPREF%=0 : SWBODY%=0
11282 FOR J%=1 TO TNC%
11284 W%=ASC(MID$(TEXT$,J%,1))
11285 IF W%=91 THEN SWPREF%=1 : GOTO 11300
11286 IF W%=93 THEN SWBODY%=1 : GOTO 11300
11290 IF SWPREF%=0 THEN
          R.PREF$(I%)=R.PREF$(I%)+CHR$(W%):GOTO 11300
11292 IF SWBODY%=0 THEN
          R.BODY$(I%)=R.BODY$(I%)+CHR$(W%):GOTO 11300
11294 R.SFIX$(I%)=R.SFIX$(I%)+CHR$(W%)
11300 NEXT J%
11305 IF RNC%=0 THEN 11310
11306 FOR J%=1 TO RNC%*4
11307 INPUT #1,A%                                             :'LOAD REGISTERS AS %
11308 R.PHON$(I%)=R.PHON$(I%)+CHR$(A%)
11309 NEXT J%
11310 NEXT I% : CLOSE
11400 '╔══════════════════════════════════════════════════════════════════════════════╗
```

```
11410 '║                         LOAD  INDEX                         ║
11420 '╚═════════════════════════════════════════════════════════════╝
11430 FOR I%=1 TO NUMBER.OF.RULES%
11434 IF LEN(R.BODY$(I%)) = 0 THEN 11470
11435 A.R.BODY%=ASC(R.BODY$(I%))
11440 IF A.R.BODY% = CURR.INDEX% THEN 11470
11450 CURR.INDEX% = A.R.BODY%
11460 INDEX.RULES%(CURR.INDEX%)=I%
11470 NEXT I%
11475 FOR I%=1 TO 255 :IF INDEX.RULES%(I%)=0 THEN INDEX.RULES%(I%)=256
11478 NEXT I%
11480 RETURN
12000 '╔═════════════════════════════════════════════════════════════╗
12010 '║                                                             ║
12020 '║                   ERROR TRAPPING ROUTINE                    ║
12030 '║                                                             ║
12040 '╚═════════════════════════════════════════════════════════════╝
12045 IF ERR=53 AND ERL = 13040 THEN RESUME 13100
12050 LOCATE 23,3 : PRINT STRING$(35,32)
12060 LOCATE 23,3 : PLAY "L8N10"
12080 IF ERR=7 THEN PRINT  " OUT OF MEMORY";:GOTO 12230
12090 IF ERR=14 THEN PRINT " OUT OF STRING SPACE";:GOTO 12230
12100 IF ERR=24 THEN PRINT " DEVICE TIME OUT";:GOTO 12230
12110 IF ERR=25 THEN PRINT " DEVICE FAULT";:GOTO 12230
12120 IF ERR=27 THEN PRINT " OUT OF PAPER";:GOTO 12230
12130 IF ERR=51 THEN PRINT " INTERNAL ERROR ";:GOTO 12230
12140 IF ERR=53 THEN PRINT " KNOWLEDGE NOT FOUND";:GOTO 12230
12150 IF ERR=57 THEN PRINT " DEVICE I/O ERROR ";:GOTO 12230
12160 IF ERR=61 THEN PRINT " DISK FULL";:GOTO 12230
12170 IF ERR=64 THEN PRINT " BAD FILE NAME ";:GOTO 12230
12180 IF ERR=67 THEN PRINT " TOO MANY FILES ";:GOTO 12230
12190 IF ERR=68 THEN PRINT " DEVICE UNAVAILABLE ";:GOTO 12230
12200 IF ERR=70 THEN PRINT " DISK WRITE PROTECT ";:GOTO 12230
12210 IF ERR=71 THEN PRINT " DISK NOT READY ";:GOTO 12230
12230 PRINT " Push "+CHR$(17)+"---";
12240 IP$=INKEY$ : IF IP$="" THEN 12240          :'READING FROM KEYBOARD
12250 IF ASC(IP$) <> 13 THEN 12240
12260 LOCATE 23,3:PRINT STRING$(35,32);
12270 IF ERR=53 AND ERL = 11220 THEN RESUME 11040
12280 IF ERR=64 AND ERL = 11220 THEN RESUME 11040
12300 RESUME 11040
13000 '╔═════════════════════════════════════════════════════════════╗
13010 '║                   LOADING FACE FEATURES                     ║
13015 '╚═════════════════════════════════════════════════════════════╝
13016 DEF SEG=&HB800
13017 BLOAD "BABEL",0   : GOSUB 11000    : 'PRESENTATION (LOAD RULES)
13020 BLOAD "TONGS1",0  : GOSUB 2500
13030 BLOAD "TONGS2",0  : GOSUB 2560
13035 OPEN "FACENAME" FOR INPUT AS #1
13036 INPUT #1,F.NAME$ : CLOSE
13040 OPEN F.NAME$ FOR INPUT AS #1
13050 INPUT #1,XN,YN,MOUTH$,FACE$
13052 BLOAD MOUTH$,0 : GOSUB 2400                : 'LOAD MOUTH SCREEN
```

```
13054 BLOAD FACE$,0
13060 CLOSE
13070 RETURN
13090 '╔══════════════════════════════════════════════════════════════════════╗
13092 '║                         LOAD MOUTHS & TONGUES                        ║
13094 '╚══════════════════════════════════════════════════════════════════════╝
13100 XN=45 : YN=117                                  : 'DEFAULT PARAMETERS
13110 BLOAD "MOUTHS2",0 : GOSUB 2400                  : 'LOAD MOUTH SCREEN
13140 BLOAD "FINAL",0
13150 RETURN
```

APPENDIX E
THE CURRENT CONFIGURATION BETWEEN SPANISH LETTERS AND IMAGES

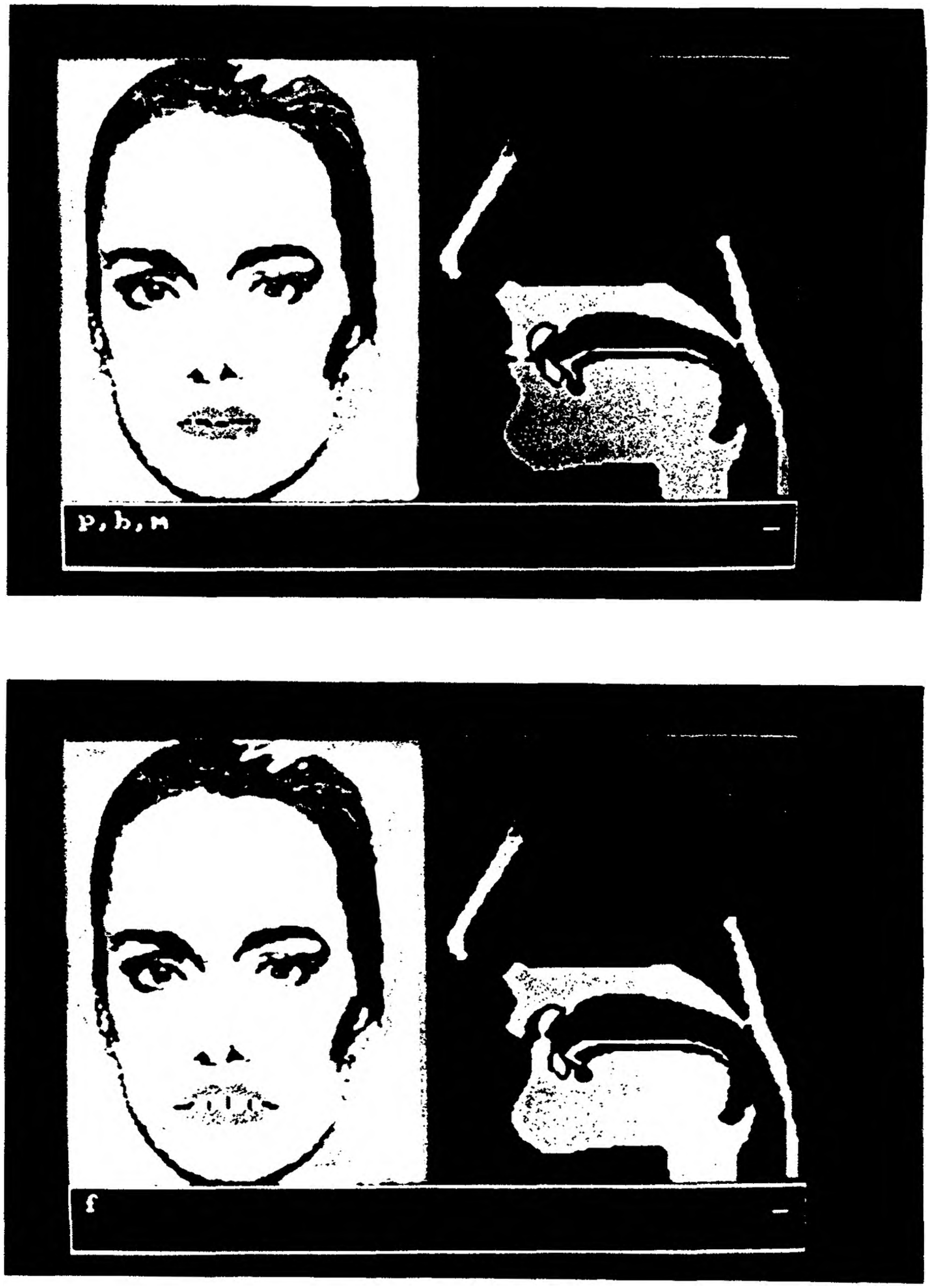


Figure 14: Correct mouth positions for pronunciation of p, b, m and f.

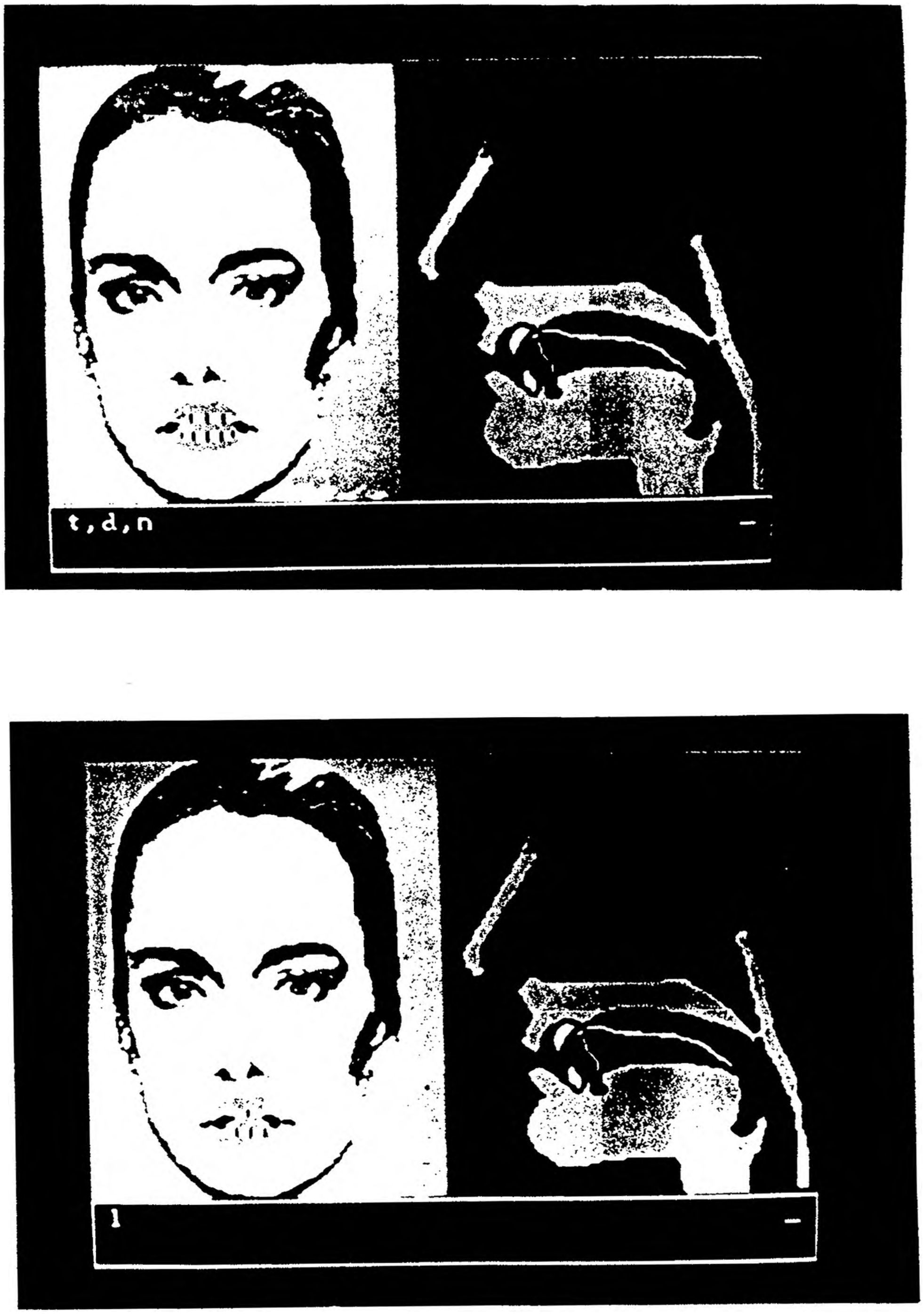


Figure 15: Correct mouth positions for pronunciation of t, d, n and l.

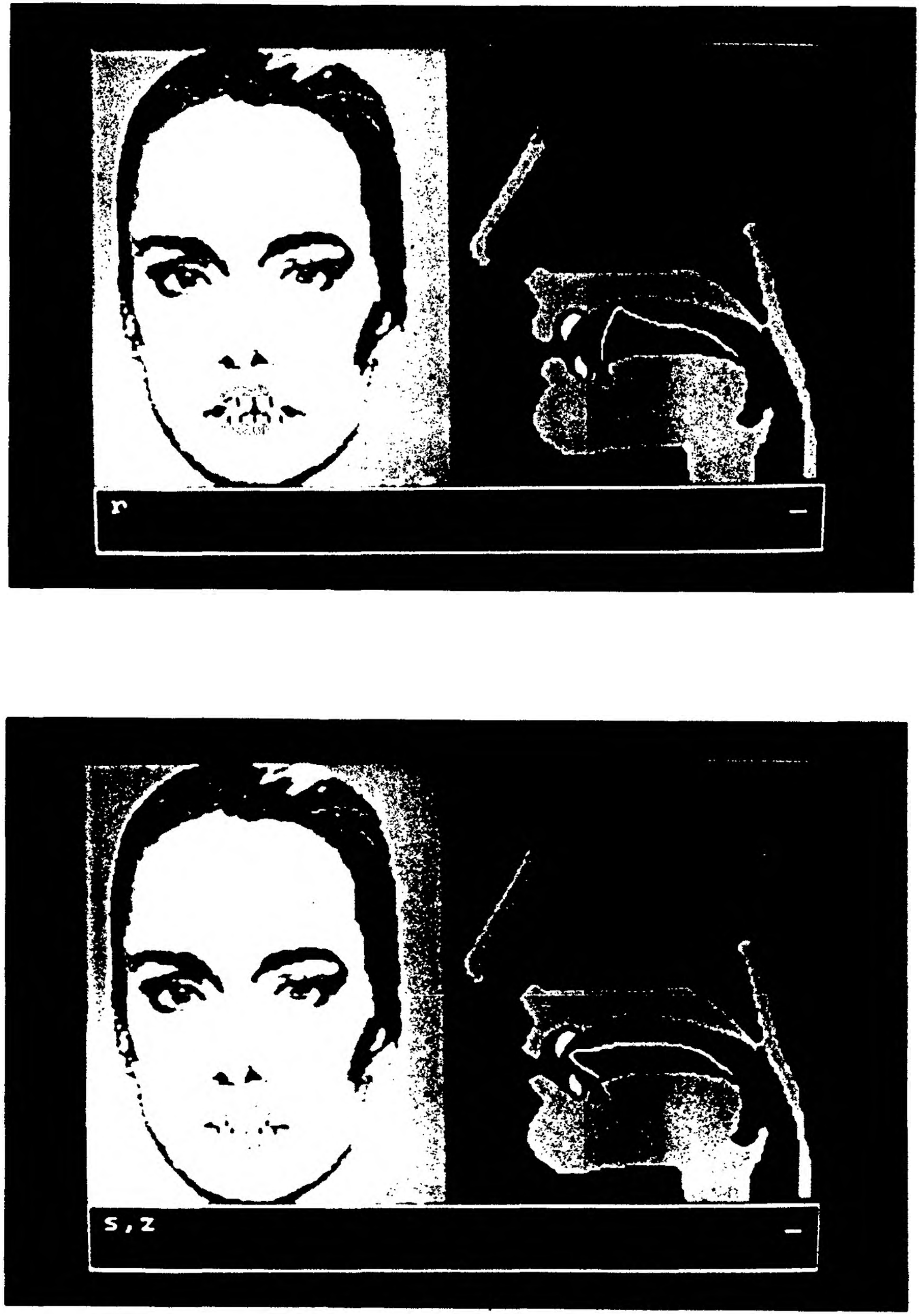


Figure 16: Correct mouth positions for pronunciation of r and s, z.

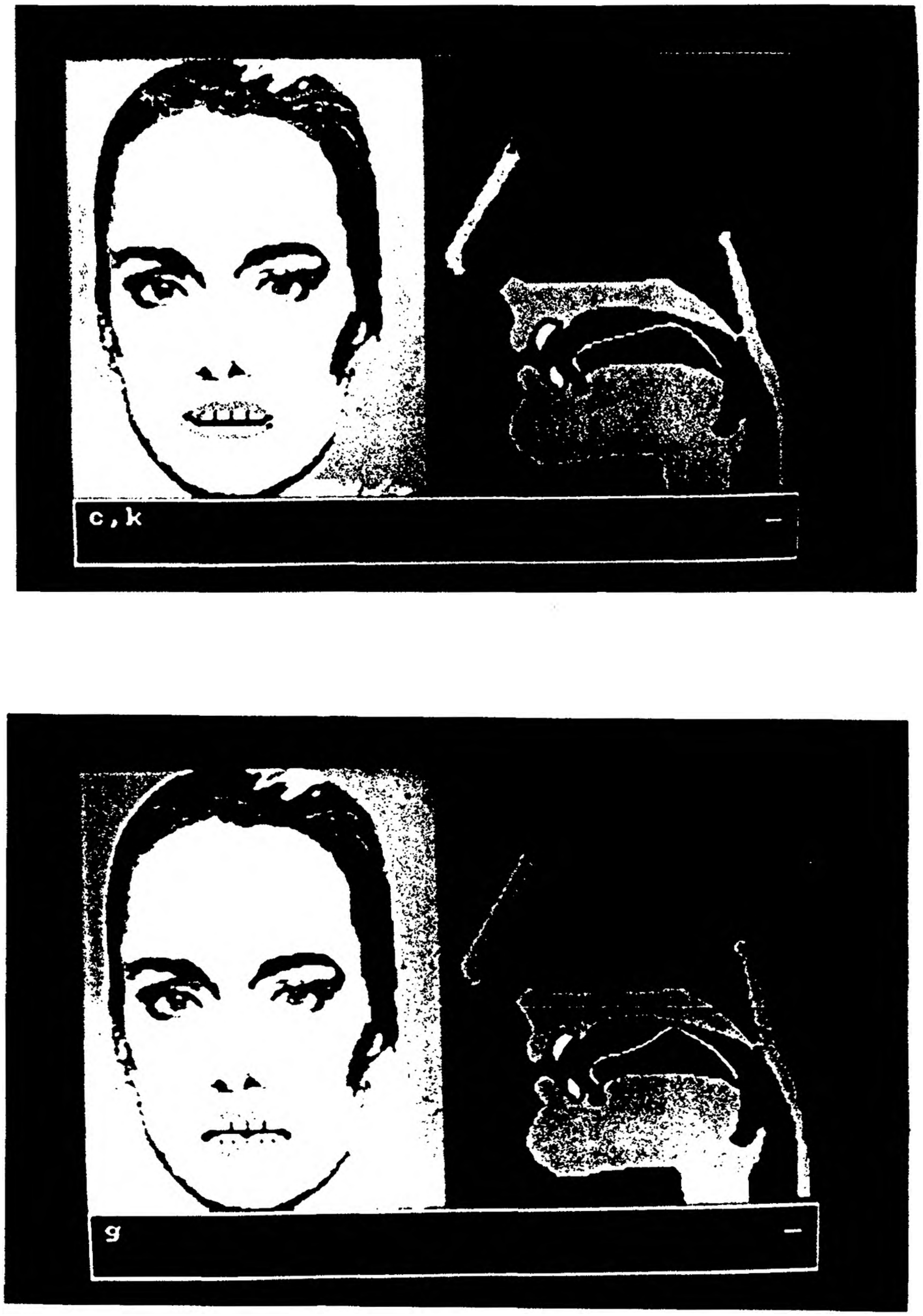


Figure 17: Correct mouth positions for pronunciation of c, k and g.

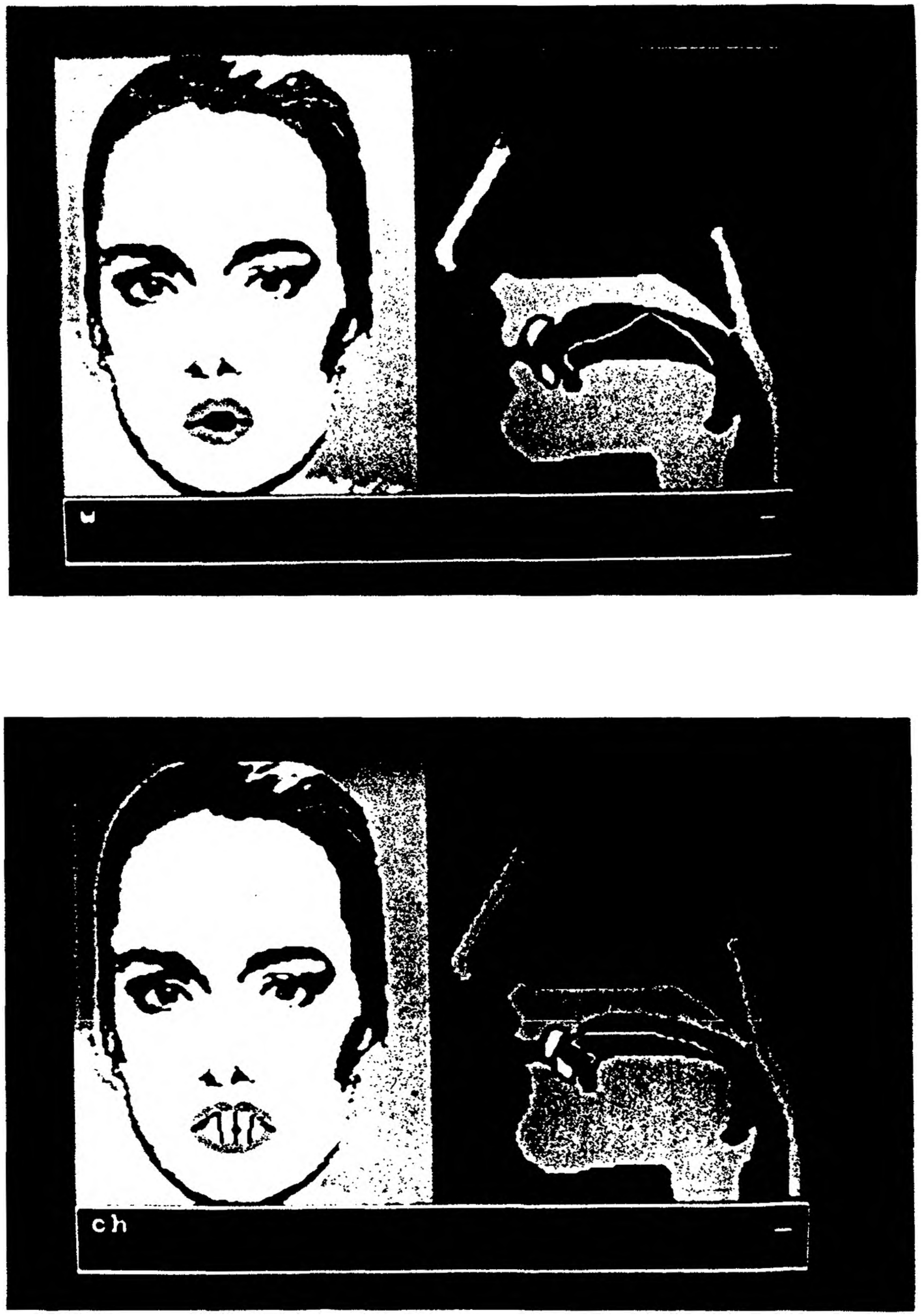


Figure 18: Correct mouth positions for pronunciation of _w_ and _ch_.

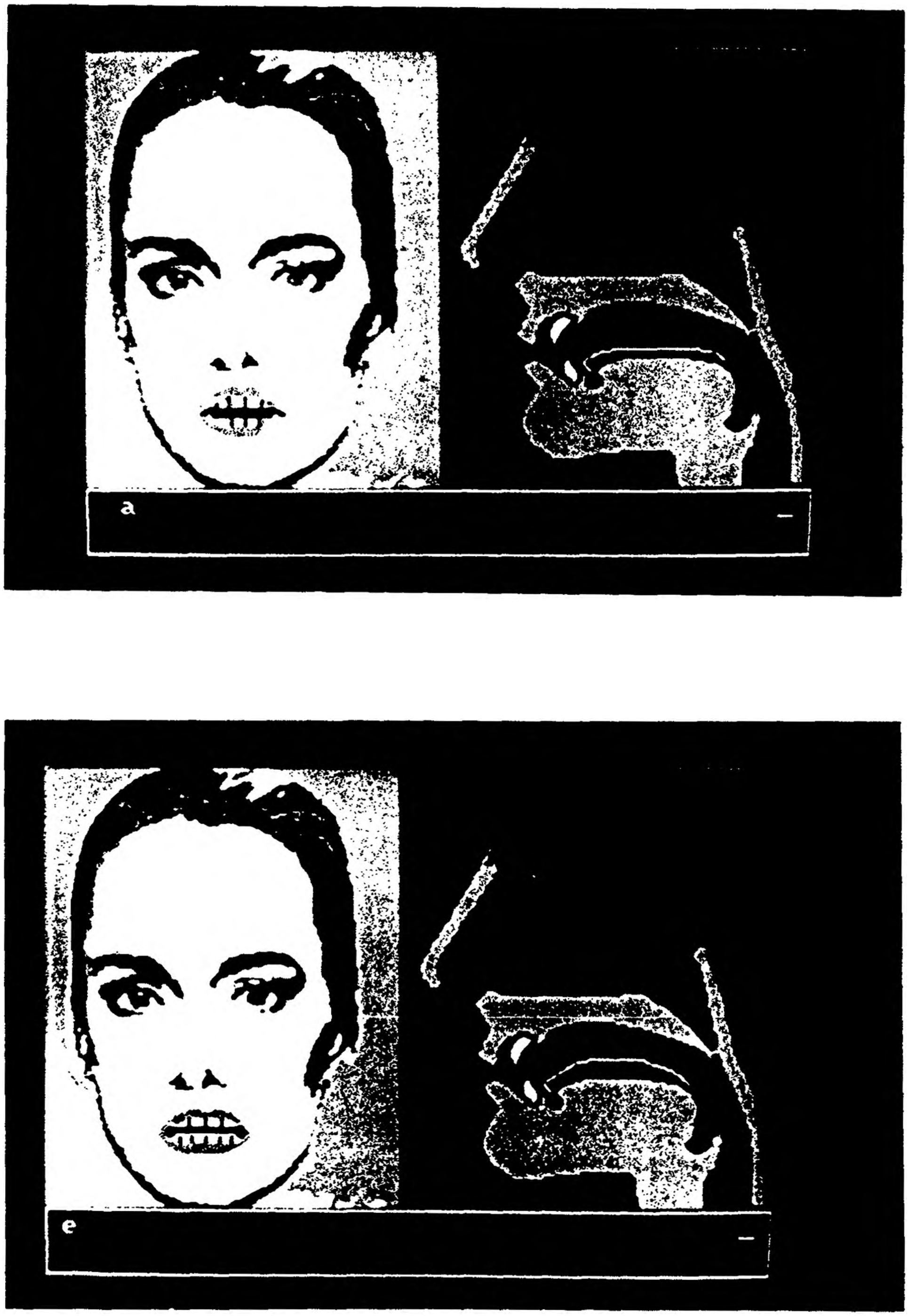


Figure 19: Correct mouth positions for pronunciation of a and e.

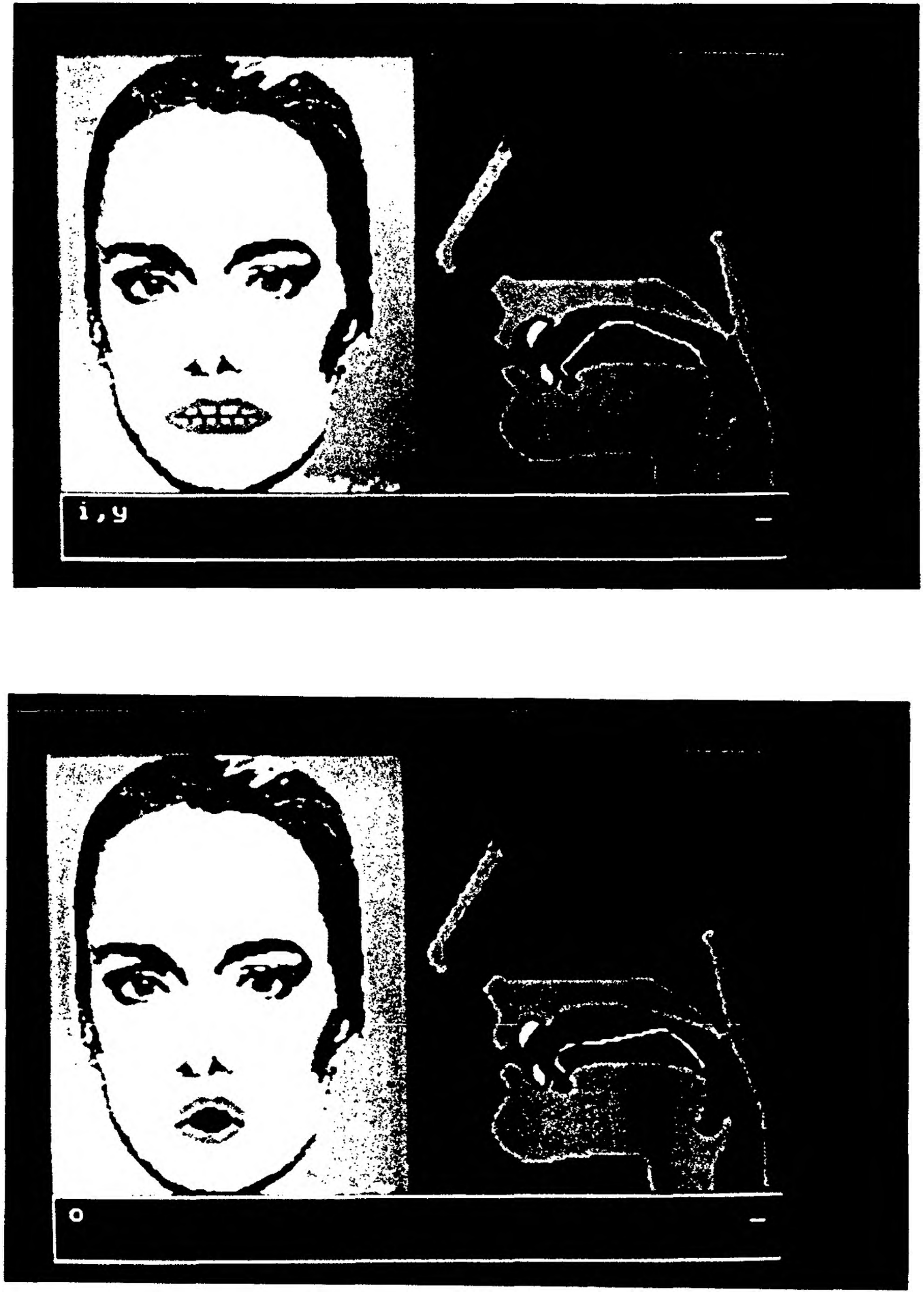


Figure 20: Correct mouth positions for pronunciation of i, y and o.

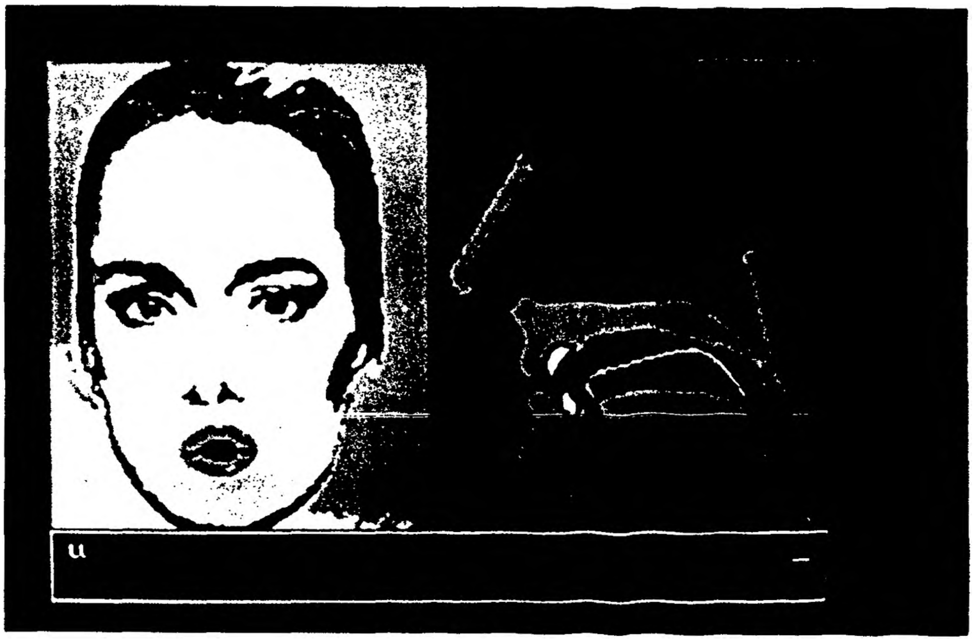


Figure 21: Position of the mouth for pronunciation of u.