LEE HARTMAN

Southern Illinois University

Carbondale, IL, U.S.A.

Lhartman@siu.edu

Phono (Version 4.0): Software for Modeling Regular Historical Sound Change

(In* Actas [del] VIII Simposio Internacional de Comunicación Social, Santiago de Cuba, 20-24 de enero del *2003 [Santiago de Cuba, 2003], I.606-609)

1. What Is Phono?

Phono is a software tool for developing and testing models of regular historical sound change. A model consists essentially of an ordered set of sound-change rules. The user types ancestor words on the keyboard, and the program displays on the screen the successive stages of development and the final descendant form. The program was developed in conjunction with a model for Latin-to-Spanish (based mainly on Otero 1971 and Hartman 1974; see also Hartman 1985), but it is designed to operate models for any natural language. For the Spanish model, the ancestor words are mainly documented (in Latin), and the purpose of Phono is to test hypotheses about sound-change rules: their contents, their relative chronology, and their cumulative effects. For models of languages with undocumented proto-forms, the program can test not only the contents and order of proposed sound-change rules, but also the proposed forms of reconstructed ancestor words. This was the case when bin Muzaffar (1996a and 1996b) used Phono to model the development of Shawnee from Proto-Algonquian.

The prospect of modeling language evolution often raises questions that make it necessary to clarify what Phono will *not* do. Phono deals only with *regular* change, although the derivation of some exceptional words can be simulated by temporarily masking a single rule in the chronological sequence (see Hartman 1986). The program deals with just one line of descendancy at a time (rather than deriving several sister-language forms simultaneously). It simulates only “downstream” derivation (that is, Phono cannot put a sound-change model “into reverse” and project upstream to generate ancestor words). Thus Phono does not carry out directly the process of comparative reconstruction of one ancestor word from several descendants in different daughter languages, although it may indirectly help with that enterprise by testing hypotheses.

2. History.

The development of Phono began in the early 1980s. Version 1 was written in PL/I for an IBM 360 mainframe, with the Spanish model hard-coded in the program. Beginning with that first version and continuing to the present, the procedure has been to receive the etymon (ancestor word), as a character string; to translate it to a set of binary feature values (+syllabic, -high, etc.); to perform the historical derivation in terms of these feature values; and finally to retranslate the resulting values back to the form of

a character string for output. For Version 1, the font available for input and output was limited in effect to the uppercase letters of the Roman alphabet. For input, the phonetic values of these letters could be refined and specified by means of a set of “adjustment rules”, applied prior to the sound-change derivation. Latin ancestor words could be entered almost entirely in Latin orthography, with the help of adjustment rules that served, for example, to reinterpret the letter <X> as a phonetic sequence of [ks], to assign stress to words, and so on. And in output, the uppercase alphabet’s lack of precision was remedied by means of a system of so-called “feature-based diacritics” (derived from Burton-Hunter 1976), whereby each segment of the character string could be accompanied by a listing of the feature values in which the segment differed from the default values of the character in the alphabet. The velar nasal “eng”, for example, would appear as uppercase <N> (the nearest equivalent character in the alphabet), qualified as [-coronal, +high, +back], and so on. This notation required an elaborate two-dimensional display for each stage of the derivation, but it did insure that no phonetic detail would be lost (see Hartman 1981).

From the beginning, it was clear that Phono needed the capability to perform “batch testing”. Each time a model is altered during its development, it needs to be retested in the Batch mode, to insure that the new alterations—made in order to account for one group of words—do not cause erroneous results for some other, unforeseen group of words. For the purpose of this kind of testing, Phono reads some large number of pairs of words—etymon and reflex (that is, ancestor and descendant forms)—from a data file. For each pair, Phono performs its derivation on the etymon and compares the result with the known reflex, marking each pair as either a “good” or a “bad” match. The bad matches serve to signal where the model has been impaired or needs improvement. In this process, the input of the known reflex form, like the input of the etymon, requires its own set of adjustment rules to provide phonetic detail sufficient for the matching operation. Versions 1, 2, and 3 used a single alphabet for etymon input, for output of the derivation, and—in the Batch mode—for input of the known reflex for comparison.

The second major version (around 1988) was written in Pascal for the personal computer (DOS operating system), with the Spanish model (and, in principle, any other model) to be read as data and interpreted by the program. The output font was composed of symbols from the ASCII character set, including some ad hoc conventions such as the “$” sign to represent the palatal sibilant. In this version the output display, with its feature-based diacritics, still occupied an entire screen for each stage of the derivation.

Version 3.0 (1993, Pascal) brought four innovations: (1) an internal editor for the alphabet, making it possible to customize the alphabet’s feature values to the user’s taste, with little risk of typographical error; (2) a display of the whole derivation in a single screen; (3) “Word-Trace” and “Rule-Trace” procedures to bring together a list of all the words affected by a particular rule, or of all the rules that affect a particular word; and (4) a system of on-line Help screens.

Version 3.1 (1994, Pascal) added to the alphabet-editor a set of internal editors for rule makeup and rule order, further reducing the risk of typographical errors. This version was made available on the Web for downloading.

The newest version, number 4, released in 2002, is written in Visual Basic to run on the Windows operating system. It is available for downloading from

http://mypage.siu.edu/lhartman. Its most notable advance over Version 3 is the display of output in a standard phonetic font: the alphabet of the International Phonetic Association. This IPA font is a copyrighted product of SIL International (www.sil.org), 1993, used with permission, and bundled with the program. While previous versions used a single alphabet for etymon input, for output of the derivation, and for input of known reflex forms in the Batch mode, Version 4 handles these three functions with three separate alphabets. The phonetic alphabet for output is “read-only”, while the two alphabets for input—of the etymon and of the known reflex respectively—are subject to editing by the user. Adjustment rules are still necessary to supply phonetic detail to the keyboard notation of input, but the role of these rules is reduced by the use of independent alphabets for etymon and reflex. Thanks to the regularity of the Spanish spelling system (unlike English, for example), reflex words in this language can be supplied in their orthographic form, leaving to the adjustment rules such tasks as silencing the letter <h>, or interpreting the letter <c> as [k], or as an [s] or theta sound, or as the palatal affricate “che”, according to the following letter.

Like its predecessors, Version 4 has the potential to enhance the precision of the output strings by means of feature-based diacritics. In the feature matrix of the word ready for output, if a segment matches the values of a symbol in the Phonetic Alphabet in every one of its 20 features, then that symbol appears on the screen in black. But if any of the feature values of the segment differ from those of the “nearest equivalent” symbol in the Phonetic Alphabet, then the symbol is displayed in blue, and the user can touch it with the mouse-pointer to see the names and signs of the features whose values differ. In Version 4, this capability is much less needed than before, given the improved precision of the IPA font.

Like previous versions, Version 4 offers the possibility of marking any rule as “persistent” in the chronological sequence. Persistent rules (defined by Chafe 1968) are those that reapply throughout the derivation whenever their conditions occur. Each time an ordinary rule (a so-called “transient” rule) brings about a change in the word, Phono traverses the entire list of persistent rules automatically. For example, a rule of assimilation within consonant clusters can be made persistent to apply to new clusters whenever these are formed by the loss of a vowel.

3. Computing Challenges

Given the decision to carry out the historical derivation at the level of binary feature values (rather than, say, through a process of whole-segment replacement), the main challenges of the Phono project have come down to questions of notation: (1) what keyboard conventions to use for the input of ancestor words and, in the Batch mode, for the input of known descendant words; (2) what screen conventions and font to use for the output display of derivations; and (3) how to represent the rules in a form that would be sufficiently versatile to accommodate any rule that might occur, and yet most easily learnable for the user who wishes to edit the rules (see Hartman 1993a and 1993b).

The question of notation for word input was solved fairly simply through the use of input adjustment rules (one set for the etymon words, another set for the known reflex words of the Batch mode). In the new version, the need for these rules has been reduced, but not eliminated, by the use of separate alphabets for etymon and reflex.

The question of output notation (the main problem cited by Becker 1996 in his review

of Version 3.2) has been largely solved in Version 4 through the use of the IPA phonetic font. Feature-based diacritics are still available for signaling unexpected discrepancies in derivations, but the need for them has likewise been greatly reduced by the precision of the IPA alphabet.

The third notation problem, that of the rules themselves, has proven to be a harder nut to crack. Phonologists have a somewhat standard way of expressing sound rules in the format "A > B / C _ D", meaning that element A becomes element B in the environment following C and preceding D—or in other words, every instance of "CAD" becomes "CBD". But my experience with the Spanish model shows that this convention—even with the refinements of curly braces, angled brackets, parentheses, etc. that were codified by Chomsky and Halle (1968)—can be inadequate to express the complexity of some naturally occurring rules. As a result, Phono continues in Version 4 with essentially the same rule notation as previous versions: rules are expressed mainly in terms of binary feature values and locations in the word, using a hierarchy of *if*-clauses followed by a series of *then*-clauses. This system, although logical within itself, requires some learning effort on the part of the user. On the other hand, Version 4's Rule Editor is arguably more user-friendly than that of earlier versions: it operates almost entirely by movements and clicks of the mouse, and the user is encouraged to work with it exclusively, rather than edit the text of rules with a word processor and risk introducing typographical errors.

4. Future Developments.

Phono's Version 4 is complete in its essential components: the apparatus for derivations, in both the Interactive mode and the Batch mode for pair-testing; and the internal editors for the input alphabets and the contents and order of rules. At the time of this writing, some auxiliary features of Version 3 have yet to be transferred to Version 4: the so-called "singleton" batch mode, in which a list of ancestor words can be read and put through derivation with their outcomes written to a data file rather than to the screen; and the so-called "trace" procedures—Rule Trace and Word Trace—mentioned above. These features will be incorporated in the next few months. Likewise in the near future a system of on-line Help screens will be incorporated (at present, Version 4 is bundled with a "readme" file that serves as a user's manual). I hope that, in the future, I (or other programmers, since the source code is public) can find ways to make the rule notation more similar to the standard notation that is familiar to linguists. I encourage researchers to use Phono to test models for languages other than Spanish, from a variety of language families, and to provide feedback for further development of the program, in order to insure that it serves the needs of historical phonology universally.

References

Becker, Donald A. 1996. "Historical Linguistics as a Hacker's Paradise: Review of Phono 3.2". *Glot International*, 2:22.

Bin Muzaffar, Towhid. 1996a. "Computer Simulation of Shawnee Historical Phonology". M.A. thesis, Memorial University of Newfoundland.

_____. 1996b. "Computer Simulation of Shawnee Historical Phonology". In *Canadian Linguistic Association Annual Conference Proceedings* (Calgary: Calgary Working Papers in Linguistics), pp. 293-303.

Burton-Hunter, Sarah K. 1976. “Romance Etymology: A Computerized Model”. *Computers and the Humanities*, 10:217-220.

Chafe, Wallace. 1968. “The Ordering of Phonological Rules”. *International Journal of American Linguistics*, 34:115-136.

Chomsky, Noam, and Morris Halle. 1968. *The Sound Pattern of English*. New York: Harper & Row.

Hartman, Steven Lee. 1974. “An Outline of Spanish Historical Phonology”. *Papers in Linguistics*, 7:123-191.

_____. 1981. "A Universal Alphabet for Experiments in Comparative Phonology". *Computers and the Humanities*, 15:75-82.

_____. 1985. "A Computer Model of Spanish Historical Sound Change". In *Homenaje a Álvaro Galmés de Fuentes* (Madrid: Gredos), 2:89-98.

_____. 1986. "Learnèd Words, Popular Words, and 'First Offenders'". In Oswaldo Jaeggli and Carmen Silva-Corvalán (eds.),*Studies in Romance Linguistics* (Dordrecht: Foris), pp. 87-98.

_____. 1993a. “Three Problems of Notation in Modeling Sound Change”. Paper presented at Round Table on Computer Applications in Historical Linguistics, Brussels, Belgium, December 8.

_____. 1993b. "Writing Rules for a Computer Model of Sound Change." In *Southern Illinois Working Papers in Linguistics and Language Teaching*, 2:31-39.

Otero, Carlos-Peregrín. 1971. *Evolución y revolución en romance*. Barcelona: Seix Barral.