XVII. MECHANICAL TRANSLATION*

V. H. Yngve
Carol M. Bosche
Elinor K. Charney
Ann Congleton
J. L. Darlington
Muriel S. Kannel
E. S. Klima
E. S. Lowry
J. D. McCawley
T. More, Jr.
W. K. Percival
A. C. Satterthwait
K. G. Sellin
J. S. Wright

A. LOGICAL EVALUATION OF ARGUMENTS STATED IN "FORMAT Q"

The COMIT program for logical translation and evaluation (Quarterly Progress Reports No. 68 (pages 174-175) and No. 69 (pages 165-168)) has been developed to the point at which one may submit for evaluation an entire argument written in a quasi-logical notation, "format Q." The program translates the argument into a strictly logical functional calculus notation, "format L," and then proceeds to test its validity by using the Davis-Putnam proof-procedure algorithim.[1] The following excerpts from the machine output resulting from the translation and evaluation of a sample argument may be presented and briefly discussed. The sample argument, as it originally occurred in a logic textbook,[2] is

> "Whoever belongs to the Country Club is wealthier than any member of the Elks Lodge. Not all who belong to the Country Club are wealthier than all who do not belong. Therefore not everyone belongs either to the Country Club or the Elks Lodge."

This argument was translated by hand into the following "format Q" representation; in this form it was submitted to the machine, which then proceeded to translate it into "format L," test it, and find it to be valid, in the time of 0.7 minute, exclusive of compilation.

THE INPUT ARGUMENT IS ALL + X/A + SUCH + THAT + X/A + BELONGS + TO + THE + COUNTRY + CLUB + IS + WEALTHIER + THAN + ALL + X/B + SUCH + THAT + X/B + BELONGS + TO + THE + ELKS + LODGE + . + SOME + X/C + SUCH + THAT + X/C + BELONGS + TO + THE + COUNTRY + CLUB + IS + NOT + WEALTHIER + THAN + SOME + X/D + SUCH + THAT + X/D + BELONGS + NOT + TO + THE + COUNTRY + CLUB + . + THEREFORE + SOME + X/E + SUCH + THAT + X/E + BELONGS + NOT + TO + THE + COUNTRY + CLUB + IS + AN + X/E + SUCH + THAT + X/E + BELONGS + NOT + TO + THE + ELKS + LODGE + .

The sample argument consists of three sentences, the first two of which are premises and the third of which is the conclusion, since its first word is 'therefore'. The program proceeds to parse each sentence individually, in accordance with the grammar

*This work was supported in part by the National Science Foundation (Grant G-24047).

described in Quarterly Progress Report No. 69 (pages 165-168). In the course of parsing, each word or sequence of words constituting what logicians call a "predicate," either simple or relational, is labelled with 'P' and given a numerical subscript. In our example, 'belongs to the Country Club' is labelled with 'P/.37', 'belongs to the Elks Lodge' with 'P/.38', and 'is wealthier than' with 'P/.225'. For the negative predicates, like 'belongs not to the Country Club', the subscript '/NOT' is added to the appropriate 'P'.

The structure of each parsed sentence corresponds to an equivalent structure in format L. The program contains a list of these equivalences and uses them to translate each sentence into format L. For all but the most simple sentences, the translation into format L involves several applications of these structural equivalences, since the first application usually results in a formula containing parts that are not yet in format L. These parts are then translated in turn into format L, and the results of these translations may themselves contain parts that need further translation. This translative "loop" is repeatedly executed until each sentence is entirely in format L. Any resulting formula that is not already in prenex normal form, with all the quantifiers on the left, is run through a subroutine that puts it into this form.

The three prenex formulae, corresponding to the three sentences of the argument, were printed out by the machine as follows:

(1) THE PRENEX NORMAL FORM IS Q/A, ALL + Q/B, ALL + *(+ *(+ *(+ P/.37, A + *) + AND/C + *(+ P/.38, B + *) + *) + IMPLIES/C + *(+ P/.225 + *(+ X/A + , + X/B + *) + *) + *) +

(2) THE PRENEX NORMAL FORM IS Q/SOME, C + Q/SOME, D + *(+ *(+ *(+ P/.37, C + *) + AND/C + *(+ P/.37, NOT, D + *) + *) + AND/C + *(+ P/.225, NOT + *(+ X/C + , + X/D + *) + *) + *) +

(3) THE PRENEX NORMAL FORM IS Q/SOME, E + *(+ *(+ P/. 37, NOT, E + *) + AND/C + *(+ P/.38, NOT, E + *) + *) +

These formulae are then combined into a single formula, of implicational form, in which the conjunction of the premises is taken to imply the conclusion. The single formula is then put into prenex normal form and undergoes some added changes in format. All of the information contained in the subscripts, numerical or otherwise, is incorporated into the symbols themselves and the subscripts are eliminated. The principal reason for these changes in format is that the subsequent proof-procedure program, which was actually written before the translation program, was written without using any subscripts. This conversion of format has been greatly facilitated by the recent addition to the COMIT system of a provision for elevating any subscript on a symbol into a symbol in its own right (it may also work the other way around – a symbol may be turned into a subscript). The resulting formula, representing the input argument, was printed out by the machine as follows:

ENTERING PROOF PROCEDURE THE FORMULA IS (EA)(EB)(AC)(AD)(EE)(((((P37A) AND(P38B))IMPLIES(P225AB))AND(((P37C)AND(NOT(P37D)))AND(NOT(P225CD)))) IMPLIES((NOT(P37E))AND(NOT(P38E))))

The proof-procedure program, based on the Davis-Putnam algorithm, operates by reductio ad absurdum, that is, by negating the formula and attempting to derive a contradiction. For the sample argument, the negated formula consists of a sequence of quantifiers,

'(AA)(AB)(EC)(ED)(AE)',

followed by a matrix in conjunctive normal form,

'P37C AND NP37D AND NP225CD AND NP37A NP38B P225AB AND P37E P38E AND'.

The existentially quantified variables, 'C' and 'D', are replaced in the matrix by 'PAB' and 'QAB', respectively, which are distinct functions of 'A' and 'B', the universally quantified variables that precede 'C' and 'D' in the sequence of quantifiers. This gives the matrix

'P37PAB AND NP37QAB AND NP225PABQAB AND NP37A NP38B P225AB AND P37E P38E AND'.

In evaluating the sample argument, the program generated a sequence of $3^3 = 27$ "quantifier-free lines" on the basis of this matrix, by substituting the terms 'A', 'PAA', and 'QAA' for the variables 'A', 'B', and 'E' in all possible combinations. These 27 lines were found to contain a contradiction; thus the original formula is valid.

In the immediate future, it is hoped that the program described can be improved in some or all of the following ways:

(1) By mechanizing the translation from ordinary language into format Q, or at least from a restricted ordinary language into format Q.

(2) By expanding the list of quantifier-words, at present restricted to 'all', 'some', 'no', 'only', and 'the', so as to allow for numerical propositions. The program already permits 'at most n', where 'n' is a whole number equal to or less than 20, to be used as a quantifying expression. We next plan to program 'at least n' and 'exactly n', the latter of which will be treated as the conjunction of 'at most n' and 'at least n'.

(3) By expanding the grammar so as to admit a greater variety of sentence-types, at present restricted to sentences in which two "NPs" (noun phrases) are connected by a form of the verb 'to be', or by a "binary relational predicate," such as 'is wealthier than'. It eventually will be desired to handle relational predicates of greater degree, such as the ternary predicate illustrated by the construction 'A gives B to C'.

J. L. Darlington

References

1. M. Davis and H. Putnam, A computing procedure for quantification theory, J. Assoc. Computing Machinery 7, 201-215 (1960).

2. I. M. Copi, Symbolic Logic (Macmillan and Company, New York, 1958), p. 140, no. 2.

B. CONSONANT MUTATION IN FINNISH

1. Introduction

In all Finnish words except those to be noted in section 3, a noninitial stop at the beginning of a short closed syllable undergoes various changes that depend on the surrounding segments, and thus produce alternations when an affix such as the genitive ending -n is added which closes a syllable that had hitherto been open. The changes are as follows:

(a) single /p, t, k/ become /v, d, zero/, respectively, except that /k/ becomes /j/ after /h/ or between a liquid and /i/ or /e/, /k/ becomes /v/ between two /u/'s or /y/'s,[1] and there is no mutation if the stop is preceded by /s/ (also sometimes if preceded by /h/) or in the combination /tk/.

	'help'	'worm'	'river'		'straw'	'calf (of leg)'	'dress'
Nominative	apu	mato	joki	but	olki	pohjeh	puku
Genitive	avun	madon	joen		oljen	pohkeen	puvun

(b) A geminate stop is shortened:

	'priest'	'carpet'	'flower'
Nominative	pappi	matto	kukka
Genitive	papin	maton	kukan

(c) Nasal plus single stop becomes geminate nasal:

	'coast'	'warmth'	'thread'
Nominative	ranta	lämpö	lanka
Genitive	rannan	lämmön	laŋŋan (written langan)

(d) Liquid plus /t/ becomes geminate liquid:

	'evening'	'stream'
Nominative	ilta	virta
Genitive	illan	virran

2. Mutation Rules

I shall endeavor here to state a maximally simple set of rules to generate correctly all forms with consonant mutation. Note first that if the above-given statement of the facts is translated directly into distinctive-feature terms, the resulting rules would be extremely cumbersome. A much simpler set of rules can be obtained by describing the process in a somewhat different fashion, namely, by saying that all stops undergo some particular change in the environment[2] $\_\_\_\_\_ \, C\begin{Bmatrix} C \\ \# \end{Bmatrix}$ and that the resulting forms are converted into the correct forms by other rules that perform assimilations or deletions.

It is necessary to assume the following set of morphophonemically distinct consonants for the base forms of Finnish morphemes: p, t, k, v, s, h, m, n, l, r.[3] Note that the feature of voicing is nondistinctive, that is, predictable: the features of obstruence, continuence, nasality, and those features relating to place of articulation (/p, k/ are grave and /t/ nongrave; /k/ is compact and /p, t/ noncompact) are sufficient to distinguish between these segments. The segments that are voiced are the vowels, the resonants (liquids and nasals), and /v/. Voicing can thus be predicted by the following rules:

$$[\quad] \rightarrow [-\text{voice}]$$

$$\begin{Bmatrix} [-\text{obs}] \\ \begin{bmatrix} +\text{obs} \\ +\text{cnt} \\ +\text{grv} \\ -\text{cmp} \end{bmatrix} \end{Bmatrix} \rightarrow [+\text{voice}]$$

that is, everything becomes voiceless except nonobstruents (vowels, liquids, and nasals) and /v/, which become voiced.

Since the result of consonant mutation is a voiced segment except in only those cases in which a segment is deleted entirely, I propose having the consonant mutation rule make all stops voiced in the given environment and then have a set of assimilation and deletion rules that apply to voiced stops. This will necessitate, of course, that the rules that predict voicing occur earlier in the grammar than the consonant-mutation rules. I thus state the rule:

1. $[-\text{cnt}] \rightarrow [+\text{voice}] \quad \text{in env} \; \_\_\_\_\_ \, V\,C\begin{Bmatrix} C \\ \# \end{Bmatrix}.$

This, among other things, converts the geminate stops pp, tt, kk into pb, td, kg. Since geminate stops become single stops under consonant mutation, a rule will be needed to delete a voiced stop preceded by a homorganic voiceless stop:

2. $\begin{bmatrix} +\text{obs} \\ +\text{voice} \\ \alpha\,\text{grv} \end{bmatrix} \rightarrow \emptyset \quad \text{in env} \quad \begin{bmatrix} +\text{obs} \\ -\text{cnt} \\ \alpha\,\text{grv} \end{bmatrix} \_\_\_\_\_ \, .$

The specification of $[\alpha\ \text{grv}]$[4] ensures that the rule will apply only to homorganic-stop sequences. Note that /tk/, the only nonhomorganic-stop sequence, does not undergo consonant mutation.

The combinations sp, st, sk, and tk are unaffected by consonant mutation. That means that the sb, sd, sg, and tg into which rule 1 would convert them must be made voiceless again. This is accomplished by the following rule:

3. $$[+\text{obs}] \rightarrow [-\text{voice}] \quad \text{in env} \quad \begin{bmatrix}+\text{obs}\\-\text{grv}\end{bmatrix} \underline{\qquad} .$$

There next follows a set of assimilation rules:

4. $$\begin{bmatrix}+\text{obs}\\+\text{voice}\end{bmatrix} \rightarrow \begin{bmatrix}+\text{nas}\\-\text{obs}\end{bmatrix} \quad \text{in env} \quad [+\text{nas}] \underline{\qquad} .$$

5. $$\begin{bmatrix}+\text{obs}\\+\text{voice}\\+\text{grv}\end{bmatrix} \rightarrow \left\{ \begin{array}{lll} [-\text{cmp}] & \text{in env} & \begin{bmatrix}-\text{cns}\\+\text{flt}\\+\text{dif}\end{bmatrix} \underline{\qquad} \begin{bmatrix}-\text{cns}\\+\text{flt}\\+\text{dif}\end{bmatrix} \\ \begin{bmatrix}-\text{cns}\\-\text{obs}\\-\text{grv}\end{bmatrix} & \text{in env} & \left\{ \begin{array}{l} \begin{bmatrix}+\text{cns}\\+\text{voc}\end{bmatrix} \underline{\qquad} \begin{bmatrix}-\text{cns}\\-\text{flt}\\-\text{cmp}\end{bmatrix} \\ \begin{bmatrix}-\text{cns}\\-\text{voc}\\+\text{cmp}\end{bmatrix} \underline{\qquad} \end{array} \right\} \end{array} \right\} .$$

6. $$\begin{bmatrix}+\text{obs}\\+\text{voice}\\-\text{grv}\end{bmatrix} \rightarrow \begin{bmatrix}-\text{obs}\\ \alpha\,\text{cnt}\end{bmatrix} \quad \text{in env} \quad \begin{bmatrix}+\text{cns}\\+\text{voc}\\ \alpha\,\text{cnt}\end{bmatrix} \underline{\qquad} .$$

Rule 4 converts /b, d, g/ into nasals when they are preceded by nasals, and thus has the total effect of creating geminate nasals out of nasal-stop sequences. The first part of rule 5 changes /g/ into /b/ (which later becomes /v/) when it occurs between two /u/'s or /y/'s, and the second part converts it into a /j/ when it occurs in the environment Liquid $\underline{\qquad} \left\{ \begin{matrix} \text{i} \\ \text{e} \end{matrix} \right\}$ or /h/$\underline{\qquad}$. Rule 6 assimilates /d/ to a preceding liquid.

It remains only to delete all remaining /g/'s (in accordance with the fact that consonant mutation converts /k/ into zero in all environments except those discussed above) and to convert all remaining /b/'s into /v/'s. This is accomplished by the rules:

7. $$\begin{bmatrix}+\text{obs}\\+\text{cmp}\\+\text{voice}\end{bmatrix} \rightarrow \emptyset$$

8. $$\begin{bmatrix}+\text{obs}\\+\text{grv}\\+\text{voice}\end{bmatrix} \rightarrow [+\text{cnt}].$$

Rules 1-8 are part of a larger set of rules that have been tested by means of a computer program (written in COMIT) which executes the rules on the base forms of a given set of words and prints out the results both in the form of a matrix of distinctive-feature specifications and in a phonemic orthography. The results produced by the machine are correct for all words that I have tried thus far.

3. Environment in which Consonant Mutation Operates

I stated above that consonant mutation occurs at the beginning of a short closed syllable and gave the formula $\_\_\_\_ \; V\, C \begin{Bmatrix} C \\ \# \end{Bmatrix}$. I must now make more precise just what is to be understood by "short closed syllable," which will amount to stating what the C's are to include. Consonant mutation occurs before syllables closed by a glide; for example, the partitive plural of silakka 'herring' is silakojta (written silakoita in standard orthography). The features of consonantalness and vocalicness distinguish between true consonants, liquids, glides, and vowels as follows:

	true consonants	liquids	glides	vowels
cns	+	+	−	−
voc	−	+	−	+

The first C of the formula given thus can be any segment that is either [+ cns] or [- voc], that is, anything except a vowel. There is dialect variation as to whether the second C may be a glide or whether it can only be a liquid or true consonant. In the former type of dialect the illative plural of silakka is silakojhin, and the rule has $\begin{Bmatrix} [+\text{ cns}] \\ [-\text{ voc}] \end{Bmatrix}$ for the second C; in the latter type of dialect the illative plural is silakkojhin, and the rule has [+ cns]. The reformulation of rule 1 which is given below is for the latter type of dialect. Both silakojhin and silakkojhin seem to be equally frequent in educated Finnish speech. Finally, consonant mutation does not apply to an initial stop; for example, pappi does not become *vappi. This means that the rule must require that the stop in question be preceded by at least one segment. The full form of the rule, thus, is

$$1.\quad [-\text{ cnt}] \rightarrow [+\text{ voice}] \text{ in env } [\quad]\; \_\_\_\_ \, V \begin{Bmatrix} [+\text{ cns}] \\ [-\text{ voc}] \end{Bmatrix} \begin{Bmatrix} [+\text{ cns}] \\ \# \end{Bmatrix}$$

With respect to my assertion that consonant mutation occurs before short syllables closed by /j/, it may be objected that there are words in which a nonmutated consonant occurs before a VjC sequence; for example, the partitive plural of hammas 'tooth' is hampaita and not *hammaita. However, all of the words in which this happens have underlying forms in which there is a consonant between the vowel and the /j/. For example, hampaita is derived from the underlying form hampas + i + ta. Thus it suffices to assume that the rule for deleting the final consonant in these words occurs later in the grammar than the consonant-mutation rule. Then, at the time the consonant-mutation rule applies, the stops in question will not be at the beginning of a short closed syllable, so that the rule will have no effect.

4. Words That Do Not Undergo Consonant Mutation

There are a large number of proper names and incompletely assimilated loanwords that do not undergo consonant mutation. These words are also peculiar in many other respects and simply will have to be treated as a different class of words, to which certain of the morphophonemic rules will not apply.

However, there are also two situations in which native Finnish words do not undergo consonant mutation. One is words in which the closed syllable is produced by a possessive affix, for example, tupa 'hut', tupamme 'our hut' (not *tuvamme). The other is words with the suffix -nen~-s. These words have an -s- throughout their paradigms, except in the nominative singular in which it is replaced by -nen. The -nen~-s- must be treated as a separate morpheme; it often functions as either an adjective-forming or a diminutive suffix, although there are many words such as hevonen 'horse' in which it does not have such a function. In words of this type there is no consonant mutation in the partitive, even though a short closed syllable is formed.

	'sparrow'	'preceding'
Nominative	varpunen	entinen
Genitive	varpusen	entisen
Partitive	varpusta (not *varvusta)	entistä (not *ennistä)

This amounts to saying that, for the purposes of the application of consonant mutation, possessive suffixes and -s- do not count as part of the word or, what is the same thing, for the purposes of consonant mutation the words in question are split into two parts, and the rule applies to these parts separately rather than to the entire word. This is, of course, the familiar phenomenon of "juncture," about whose precise nature there has been much controversy, but which, as the word is used by all authors with whom I am familiar, consists in the splitting of an utterance into chunks that serve as the domains on which the various morphophonemic processes operate. Authors such as Trager,[5] Hill,[6] and Joos[7] speak of juncture as an independent phonetic entity; but the places in which they observe this alleged element are simply places in which a segment in the middle of an utterance behaves in some respects as if it were at the beginning or end of the utterance, so that the phenomenon still can be regarded as one of the splitting of an utterance into chunks.[8] Thus, by marking the splits in the words in question by /, tupamme and entistä are represented as tupa/mme and enti/s + tä (+ denotes morpheme boundary).

Nouns in -nen~-s- have another peculiarity, namely the fact that the obstruent-deletion rule referred to earlier does not apply to them. The rule in question deletes an intervocalic obstruent if it occurs later than the second syllable and is preceded by a single vowel.[9] The rule is involved in the generation of the oblique case forms of nouns with stems ending in a single obstruent. For example, the genitive case hampaan of the noun hammas is generated as follows:

underlying form	hampas + e + n
obstruent deletion	hampa + e + n
vowel assimilation	hampa + a + n

However, entisen does not become *entiin. The assumption of a juncture / before the affix -nen~-s would give an underlying form enti / s + e + n for entisen. For the purposes of the obstruent-deletion rule the word again would be broken up into two chunks, enti and s + e + n, and the obstruent-deletion rule would not apply to either. Representing -nen~-s nouns with a / before that affix thus simultaneously accounts for the fact that -nen~-s nouns are the only words in the native vocabulary (outside of possessive forms) which do not undergo consonant mutation and the fact that they are also the only words that have intervocalic /s/'s beyond the second syllable.

J. D. McCawley

References

1. I write my examples in a slightly modified version of standard Finnish orthography. I denote final "aspiration" by h and write j for phonetic /j/ even where the standard orthography writes i. y and ä denote the sounds represented in the International Phonetic Alphabet by ü and æ.

2. # means word boundary. The actual environment is somewhat more complicated. I state exactly what it is in section 3.

3. d and ŋ, which are regarded as separate phonemes by most American linguists, only arise through constant mutation or assimilation and thus do not have to be recognized as separate morphophonemes.

4. Greek letters α, β, ... denote variables that assume the values + and -. Variables are useful in stating identities, as in rule 2, and in formulating assimilation and dissimilation rules. For example, $[+\text{cns}] \rightarrow [\alpha\ \text{voice}]$ in env _____ $\begin{bmatrix} +\text{cns} \\ \alpha\ \text{voice} \end{bmatrix}$ would be a rule for the regressive assimilation of voicing in consonant sequences.

5. G. L. Trager and H. L. Smith, Jr., *Outline of English Structure* (University of Oklahoma Press, Norman, 1951).

6. A. A. Hill, *Introduction to Linguistic Structures* (Harcourt, Brace and Company, New York, 1958).

7. M. Joos, *English Language and Linguistics* (Institute for Experimental Phonetics, Beograd, Yugoslavia, 1958).

8. It will be necessary, of course, to have several different junctural elements, since some morphological processes apply to larger domains than do others.

9. The present form of this rule was suggested to me by R. P. Kiparski. An earlier and incorrect version of the rule is given in J. D. McCawley, Finnish noun morphology, Quarterly Progress Report No. 68, Research Laboratory of Electronics, M.I.T., January 15, 1963, pp. 180-186.