[*International Conference on Machine Translation of Languages and Applied Language Analysis*, National Physical Laboratory, Teddington, UK, 5-8 September 1961]

RANDOM GENERATION OF ENGLISH SENTENCES*

by

VICTOR H. YNGVE

Department of Modern Languages

Center for Communication Sciences, R. L. E.

THIS paper reports the results of writing and running a program which constructs English sentences. The sentences are chosen at random by the program from among those English sentences that conform to certain rules of sentence structure. This work is a continuation of a line of research begun several years ago.

IN the paper "A Framework for Syntactic Translation",[1] it was proposed that a translation routine could be divided into six logically separate parts. There was a horizontal division into three steps: sentence analysis, transfer of structure, and sentence synthesis; and there was a vertical division into the operational parts, or routines proper, and the parts that contained all the necessary knowledge of the structures of the languages involved and their interrelation. It was hoped that to divide is to conquer.

In the work reported here we are concerned with just two of the six parts of a translation routine - the sentence-synthesis routine and the grammar, which is eventually to contain as complete a set of rules for English sentence structure as possible.

Of the various possible forms for writing a grammar, the generative[2] form seems to have the most to recommend it. A generative grammar is a grammar written in a manner analogous to a deductive system. Its main advantage is that it offers a relatively easy method of dealing with the difficult problems posed by the multiple function of words and constructions. It also seems plausible that a generative type of grammar is not necessarily confined in its use to a sentence synthesis routine, but could be used equally well with a sentence recognition routine.[3]

A number of forms of generative grammar have been explored. The original transformational type of grammar has been abandoned because it cannot be mechanized by a finite device, because of the difficulty of assigning a phrase structure to the result of a transformation, and for several other lesser technical reasons.

---

*This work was supported in part by the National Science Foundation and in part by the U.S. Army (Signal Corps), the U.S. Air Force (Office of Scientific Research, Air Research and Development Command), and the U.S. Navy (Office of Naval Research).

The type of grammar and sentence synthesis mechanism finally chosen have been described in detail elsewhere.[4] The grammar consists of a finite set of phrase-structure rules that can be applied one at a time by the sentence synthesis mechanism. The rules of the grammar form an unordered set. The order in which they are applied is determined by what is needed next as the words of the unfolding sentence are produced in their natural order, that is left-to-right according to the English orthographic convention.

At present there are three types of rules allowed in the grammar. They are applied by the mechanism as follows.

Rules of the type A = B + C means that a construction or form A is to be replaced by its functional parts or constituents B and C. An example would be

SENTENCE = SUBJECT + PREDICATE.

When A is replaced by B + C, B, the first element on the right, will be treated as the element on the left when the next rule is applied, but C, the second element on the right, is placed in a temporary storage organized on a last in - first out principle.

Rules of the Type A = B
= C
= D
.
.
.

indicate that the element on the left, A, is to be replaced by any one of the elements on the right. The choice of element on the right is determined by the input to the sentence production routine. In the present program this is a random number generator so that sentences are produced at random. In a complete translation system, the choices would come from the structure transfer routine and would be effectively determined by the input sentences. The chosen symbol on the right then serves as the symbol on the left for the next rule. But if the symbol on the right is a word, it is sent to the output as part of the sentence being produced, and the symbol next in line in the temporary memory is used as the symbol on the left for the next rule. Examples would be

SUBJECT = NOUN PHRASE
= NOUN CLAUSE
= PRONOUN

or NOUN = MAN
= BOY
= WOMAN
= GIRL
= CHILD

The third type of rule can be symbolized as

A = B+....+C

It is applied in the same way as the first kind of rule except that the constituent C is placed in the temporary storage in a way that violates the

previous principle of last in - first out. The C is placed in a position immediately behind the constituent that would be next out so that for rules of this third type the principle is that the last in is given second priority. An example would be

VERB-PHRASE = VERB+ ... +ADVERB

as in "He *called* her *up.*"

Even if the grammar consists of a finite set of rules, the mechanism can in general produce sentences from an infinite set, that is the grammar might impose no limit on the length of sentences. This would be the case if there were recursive loops in the grammar in such a way that certain rules could be reapplied an unlimited number of times. There appears in fact to be no limit to the length of English sentences.

Similarly the grammar might impose no limit on the number of symbols stored at any time in the temporary memory. If such were the case, the mechanism would not be physically realizable because it would need an infinite temporary memory. An examination of English sentences appeared to show that a small temporary memory, capable of holding no more than about seven symbols, was adequate. This led to the hypothesis[4] that English and probably all languages possess grammars that impose the limitation that no more than about seven items need ever be stored in the temporary memory. A phrase-structure grammar with this restriction is equivalent to a finite-state device. Many of the complications of English syntax can be understood as the means for imposing such a restriction.

A need has thus developed for a relatively complete grammar of English. It is needed for use in the translation routine. It is needed in order to test more carefully on English the hypothesis that a grammar with the predicted restriction is adequate. It is also needed in order to explore further certain additional questions about the structure of grammars.

The grammar of a language cannot be written down immediately in its final form. It must be discovered. And as the various parts of it are discovered and written down, they must be tested. The testing of a grammar for adequacy is not easy. The aid of a computer seems indispenslble.

A set of rules that purports to represent the grammar of a language, or part of it, partitions the set of all strings of characters into two mutually exclusive subsets, the subset containing those strings that it can generate, and the subset containing those strings that it cannot generate. If the set of rules is adequate as the grammar of a language, then the set of strings that the grammar can produce will all be recognized by native speakers as belonging to the language, and the set of strings that the grammar cannot produce will be recognised by native speakers as not belonging to the language.

It is, of course, recognized that in many border-line cases, native speakers are unsure, and disagree as to whether a string is part of the language or not. But even if native speakers were always sure, and always agreed among themselves, a complete validation of a grammar would be impossible for the simple reason that the sets of strings to be tested are infinite sets. We are forced to fall back on a sampling procedure. A random sample of the set of strings generated by the grammar can be pro-

duced by a computer program and examined by native speakers. In addition, sentences that are found to occur naturally can be checked to see if they are produced by the grammar. At a later stage, part of this process can be mechanized by the use of a recognition routine.

For the first stage of writing and validating a grammar of English, it was decided to start with the simple, straightforward language of a carefully selected children's book,[5] and write the rules necessary to generate its 161 sentences. More complicated material would be turned to later. It was soon evident, however, that it would be too difficult to write the rules for the whole book before testing any of them. Attention was therefore directed to the first ten sentences, which provide a surprisingly wide linguistic diversity. These ten sentences are as follows:

> Engineer Small has a little train. The engine is black and shiny. He keeps it oiled and polished. Engineer Small is proud of his little engine. The engine has a bell and a whistle. It has a sand-dome. It has a headlight and a smokestack. It has four big driving wheels. It has a firebox under its boiler. When the water in the boiler is heated, it makes steam.

A set of tentative rules were written down that could produce these ten sentences and many other similar sentences. There are 77 rules in all. 24 are of the type A = B + C. 5 are of the type A = B+... +C. There are 13 rules of the type A = B
= C
5 of the similar type that offers 3 alternatives, and 1 each that offer 5, 7, 8, and 11 alternatives. In addition there are 26 rules of the type A = B, many of which are not strictly necessary in this grammar, but were included because it was recognized that they represented potential choices that would be necessary in an expanded grammar.

The set of sentences that can be produced is infinite because of several types of recursions. One type is:

> "The engine is black, oiled,..., and shiny." Another type is:
> "He has a black, oiled .., and shiny engine." Another is:
> "It has a headlight, a smokestack,..., and a bell." Another is:
> "the wheels under the firebox under the boiler under ..."

If each of these recursions were limited to one repetition, there would still be over 10^{20} different sentences generated by the grammar without exceeding the vocabulary of 38 words, one suffix, and three marks of punctuation.

All constructions were examined carefully to see if they were potentially recursive, and the several types of recursion included in this first attempt are typical. The coordinated adjectives are handled in the following way

A = ADJ + B	B = C
C = + E	= D
D = AND + ADJ	E = A

The rule E = A is not strictly necessary, but it serves to held keep form and function separated, a great aid to clear thinking about syntax. Care

was taken not to include rules that would lead to ungrammatical recursions. Such recursions can always be prevented by relabeling, or any of the other devices discussed in the previous paper.[4]

The five discontinuous constituent rules take care of the following five constructions. An asterisk* is used to indicate capitalization of the next letter

1) *IT *HAS* A FIREBOX *UNDER ITS BOILER*.
2) *HE *KEEPS* IT *OILED AND POLISHED.*
   *HE *MAKES* IT *BLACK.*
3) *THE* WATER *IN THE BOILER*
4) **WHEN* IT IS HEATED, IT MAKES STEAM.
5) *IT HAS A SAND-DOME.

Discontinuous constituents are quite prevalent in English and appear to be connected with depth conservation. Any grammar that does not have a simple and direct way of handling them will probably be overly complicated and unwieldy if not completely impractical.

Among the trivial shortcoming of the grammar, it must be pointed out that the article A is not changed to AN before a vowel, and the plural S is not changed to ES after FIRE-BOX. Routines to do these things are straightforward and well understood. They will be added later.

The sentence producing routine and the grammar were coded in COMIT[6]. A copy is appended. The full power of the subscript operations available in COMIT was not used in this program but will be needed for the next. The first run after the initial program check-out revealed an error in one of the grammar rules. This was corrected and a run of 100 sentences produced an output deck of cards which were then used to print the appended set of random sentences. The output sentences were for the most part quite grammatical, though of course nonsensical.

An examination of the output sentences reveals a number of interesting points for further investigation. Most of these involve the coordination structures. In several of the sentences, the same item appears more than once in a series. There may be grammatical restrictions here. Also, it appears difficult to coordinate such diverse types of singular noun phrases as *ENGINEER *SMALL, WATER, THE BOILER, BOILERS, *SMALL AND IT. These items already represent different constructions in the grammar, but are shown together for purposes of coordination.

This raises a delicate point as where to draw the line between what is grammatical and what is not grammatical, a question that is further pointed up by such sentences as *WHEN HE IS OILED, HE IS POLISHED. In the original sentence, OILED and POLISHED refer to the engine, and are used in their literal sense. In the above sentence, they would normally be construed in a different way. This fact argues in favor of setting up a classification of animate and inanimate, which could be used to restrict adjectives and nouns. BIG and LITTLE could apply to both groups of nouns, but OILED and POLISHED would have to be entered twice with different meanings. The trouble with all this is that the restriction is really semantic and not grammatical. The sentence can be construed in its literal sense, although this is admittedly a bit far-fetched. It does not seem realistic to handle semantic restrictions as if they were

grammatical restrictions of the type that exclude *WHEN HE ARE OILED, THEY IS POLISHED.

A number of the longer more complicated sentences are either awkward or confusingly ambiguous. Most of these sentences involve co-ordination, and point to areas which should be investigated further. Many of these sentences involve rules in the grammar associated with the most serious error yet found. The grammar is ill-behaved, as can be seen from the following steps in the production of a noun phrase (NP)

OBJECT
COORDINATED NP
SINGULAR NP + AND + SINGULAR NP
SINGULAR NP
THE + NOUN + LOCATIVE
THE + NOUN + IN + OBJECT

We can thus return to OBJECT at a depth greater by two. The object of the preposition should probably not be allowed to lead to a coordinated NP. The result is ambiguous anyway: THE WATER IN THE BOILER AND THE SMOKESTACK in the spoken language, at least one of the meanings can be rendered unambiguous by the proper intonation pattern.

The grammar as it stands already rules out coordination directly within coordination, as (A AND B), C AND D, but it does not rule out coordinated adjectives before the nouns of a coordinated NP, for example.

The results of this experiment lead to several conclusions. No shortcomings have appeared in the type of program used. No reason has yet appeared for doubting that it would be possible to use this same formalism for a complete grammar of English. Even with such a small and simple corpus, a lot has been learned about English syntax in the effort to generalize from the structures found. It is also interesting that some of the original vocabulary words seem to change meaning drastically when they are embedded in different, though similar contexts. We thus appear to have in this type of program a fruitful method of examining the relation of the meaning of words to their context, one of the central problems in mechanical translation.

```
                COM M476 YNGVE ENGLISH 10 PUNCH
* $ = -THE-DATE-IS-+X+*.//*RSL2, *WAM1 2 3          *
* $ = -M*4*7*6-PUNCH-OUTPUT*. // *WAA1 *
* $ = -THE-TIME-IS-+X+$.  //*RAL2, *WAM1 2 3        *
* $ = *( + *0 + *0 + *0 + *) // *Q1 1 2 3 4 5   CT
START $ = *1/1 + E // PRINT NO $
S $1 + $1 = 1 + *A + 2 //  *E1 *
* $ + *A - $1 - // *Q3 1, *N2 2, *N2 3 SENTQ
PP $ + *A+$1+$1 // *Q3 4 1,  *N2 2 , *N2 3 SENTQ
PRINT YES $1 = 1 + A + *. // *A3 2, *WAA2 3, PRINT NO PP
      NO                                              PP
P $1 + 2 = 1 + *A + 2 + - // *E1 PRINT
ENTR $1 = $
E $2 + $2 = // *S2 2 $
D $2 + $2 = 1 A + A + 2 // *N2 2, *N2 3, *S2 4 3 2 $
* $ = *.-GRAMMAR-ERROR- + 1 + *.// *WAM1 2 3 FINAL
SENTQ $2 = // *X3 LINEQ
FINISH $ = A + *.// *A3 1, *WAA1 2 CT
LINEQ $50 = // PRINT YES, *X3 ENTR
* $ = // *X3 ENTR
(GRAMMAR OF 10 ENGLISH SENTENCES)
1 A  $2 = *2/2+E                          P
2    $2 = *3/3+S+*7/7+S                   E
3 A  $2 = *4/4+D                          P
4    $2 = *5/5+S+*6/6+S                   D
5 A  $2 = ** + W/.2                       P
6 A  $2 = . + W/.1  S
7 A  $2 = *8/8+E                          P
  B     = *1*7/17+E                       P
8    $2 = *9/9+S+*1*6/16+S                E
9 A  $2 = *1*0/10+E                       P
10   $2 = *1*1/11+S+*1*6/16+S             E
11A  $2 = *1*2/12+D                       P
12   $2 = *1*3/13+S+*1*4/14+S             D
13A  $2 = WHEN + W                        P
14A  $2 = , + W/.1  S
16A  $2 = *1*7/17+E                       P
17   $2 = *6*0/60+S+*1*8/18+S             E
18A  $2 = *2*1/21+E                       P
  B     = *2*1.*5/21.5+E                  P
  C     = *2*2/22+E                       P
21   $2 = *2*3/23+S+*3*8/38+S             E
21.5 $2 = *2*3/23+S+*3*7/37+S             E
22   $2 = *2*4/24+S+*6*2/62+S             E
23A  $2 = IS + W                          P
24A  $2 = HAS + W                         P
  B     = KEEPS + W                       P
  C     = MAKES + W                       P
  D     = *2*8/28+E                       P
  E     = *3*5/35+E                       P
28   $2 = *2*9/29 +S+*3*1/31+S            D
29A  $2 = HAS + W                         P
```

```
31 A  $2 = *3*2/32+E              P
32    $2 = *3*3/33+S+*6*2/62+S    E
33 A  $2 = IN + W                 P
   B     = UNDER + W              P
35    $2 = *3*6/36+S+*3*8/38+S    E
36 A  $2 = KEEPS + W              P
   B     = MAKES + W              P
37 A  $2 = HEATED +W              P
   B     = OILED + W              P
   C     = POLISHED + W           P
38 A  $2 = *5*5/55+E              P
   B     = *4*0/40+E              P
   C     = *4*7/47+S              P
40    $2 = *4*7/47+S +*4*1/41+S   E
41 A  $2 = *4*5/45+E              P
   B     = *4*3/43+E              P
43    $2 = *4*4/44+S+*4*6.*1/46.1+S    E
44 A  $2 = , + W/.1  S
45    $2 = *4*6/46+S+*4*7/47+S    E
46 A  $2 = AND + W                P
46.1A $2 = *4*0/40+E              P
47 A  $2 = BLACK + W              P
   B     = SHINY + W              P
   C     = OILED + W              P
   D     = POLISHED + W           P
   E     = HEATED + W             P
   F     = BIG + W                P
   G     = LITTLE + W             P
   H     = PROUD + W              P
55    $2 = *5*6/56+S+*5*7/57+S    E
56 A  $2 = PROUD + W              P
57 A  $2 = *5*8/58+E              P
58    $2 = *5*9/59+S+*6*2/62+S    E
59 A  $2 = OF + W                 P
60 A  $2 = HE + W                 P
   B     = *7*1/71+S              P
62 A  $2 = *6*9/69+S              P
   B     = *6*4/64+E              P
64    $2 = *6*9/69+S+*6*5/65+S    E
65 A  $2 = *6*7/67+E              P
   B     = *6*8/68+E              P
67    $2 = *4*4/44+S+*6*8.*1/68.1+S    E
68    $2 = *4*6/46+S+*6*9/69+S    E
68.1A $2 = *6*4/64+E              P
69 A  $2 = *7*1/71+S              P
   B     = *9*3/93+S              P
71 A  $2 = *7*9/79+S              P
   B     = *7*7/77+S              P
   C     = IT + W                 P
   D     = *8*2/82+E              P
   E     = *8*0/80+S              P
   F     = *8*3/83+E              P
   G     = *8*4/84+E              P
```

```
77     $2 = *7*8/78+S+*7*9/79+S                E
78 A   $2 = **ENGINEER + W                     P
79 A   $2 = **SMALL + W                        P
80 A   $2 = WATER + W                          P
   B      = STEAM + W                          P
82     $2 = *8*5/85+S+*8*0/80+S                E
83     $2 = *8*5/85+S+*1*0*3/103+S             E
84     $2 = *8*9/89+S+*1*0*3/103+S             E
85 A   $2 = *8*6/88+S                          P
   B      = *9*2/92+D                          P
   C      = *9*0/90+S                          P
88 A   $2 = THE + W                            P
89 A   $2 = A + W                              P
90 A   $2 = HIS + W                            P
   B      = ITS + W                            P
92     $2 = *8*8/88+S+*3*1/31+S                D
93 A   $2 = *9*5/95+E                          P
   B      = *1*0*0/100+E                       P
   C      = *9*8/98+E                          P
95     $2 = *8*5/85+S+*9*6/96+S          (  D  )  E
96 A   $2 = *1*0*0/100+E                       P
   B      = *9*8/98+E                          P
98     $2 = *1*0*2/102+S+*9*9/99+S             E
99 A   $2 = *1*0*0/100+E                       P
100    $2 = *1*0*3/103+S+*1*0*1/101+S          E
101A   $2 = S + W/.1       S
102A   $2 = FOUR  + W                          P
103A   $2 = *1*0*8/108+S                       P
   B      = *1*0*5/105+E                       P
105    $2 = *1*0*6/106+S+*1*0*8/108+S          E
106A   $2 = *4*7/47+S                          P
   B      = *4*0/40+E                          P
108A   $2 = DRIVING-WHEEL +  W                 P
   B      = TRAIN + W                          P
   C      = ENGINE + W                         P
   D      = BELL + W                           P
   E      = WHISTLE + W                        P
   F      = SAND*-DOME + W                     P
   G      = HEADLIGHT + W                      P
   H      = SMOKESTACK + W                     P
   I      = WHEEL + W                          P
   J      = FIRE*-BOX + W                      P
   K      = BOILER + W                         P

(END OF GRAMMAR)
CT $=A + *N // *A1 1*
* * ( +*1+*0+*0+*)+*N = FINAL (CHECK COUNT)
COUNT *0+*) = *1+2       PC      (INCREMENT)
*     *1+*) = *2+2       PC
*     *2+*) = *3+2       PC
*     *3+*) = *4+2       PC
*     *4+*) = *5+2       PC
```

```
*       *5+*) = *6+2          PC
*       *6+*) = *7+2          PC
*       *7+*) = *8+2          PC
*       *8+*) = *9+2          PC
*       *9+*) = 2+*0          COUNT
PC      *)+$+*N = 2+1+3       *   (PRINT COUNT OF NEXT S)
*$   + *N = *0 + 1 + 1 // *Q3 1 2, *Q1 3 START
FINAL $ = -*.-*.-THE-TIME-IS +X+*.//*RAL2,*WAM1 2
                                                    3 *
*       $ = -END-OF-COMIT-RUN*. //*WAM1 *

END
```

M476 PUNCH OUTPUT

(001) * WHEN *ENGINEER *SMALL KEEPS *SMALL AND FOUR FIRE-BOXS, HE KEEPS DRIVING WHEELS, HIS STEAM, IT AND FOUR BLACK AND OILED FIRE-BOXS.

(002) * HE HAS FOUR POLISHED SAND-DOMES.

(003) * HE HAS FOUR PROUD, LITTLE, POLISHED, POLISHED AND PROUD BOILERS UNDER PROUD BELLS, STEAM AND THE WHISTLES IN FOUR WHISTLES.

(004) * WHEN STEAM IS PROUD OF THE FOUR FIRE-BOXS AND FOUR ENGINES, THE TRAIN IS SHINY.

(005) * WHEN *ENGINEER *SMALL IS PROUD, HE HAS *SMALL UNDER A LITTLE AND PROUD SMOKESTACK

(006) * WHEN HE IS PROUD AND OILED, *ENGINEER *SMALL IS POLISHED.

(007) * WATER IS BIG.

(008) * WHEN HE IS OILED, THE SHINY SMOKESTACK IS PROUD OF FOUR ENGINES.

(009) * A HEADLIGHT IS HEATED.

(010) * WHEN HE IS HEATED, *ENGINEER *SMALL IS POLISHED.

(011) * WHEN HE IS HEATED AND HEATED, STEAM IS POLISHED.

(012) * A BLACK SMOKESTACK IS PROUD AND POLISHED.

(013) * ITS STEAM IS PROUD OF WHEELS.

(014) * THE STEAM MAKES ITS STEAM AND A WHISTLE BIG AND PROUD.

(015) * THE WATER UNDER THE FOUR SHINY WHEELS HAS A BLACK BELL AND FOUR BLACK AND POLISHED TRAINS.

(016) * WHEN *ENGINEER *SMALL IS OILED, THE WATER IN THE BELLS IS HEATED.

(017) * A LITTLE DRIVING WHEEL HAS SMOKESTACKS IN *ENGINEER *SMALL.

(018) * HE IS POLISHED.

(019) * WHEN *ENGINEER *SMALL IS OILED, HIS WHISTLE MAKES BLACK AND LITTLE FIRE-BOXS.

(020) * HE IS OILED AND PROUD.

(021) * WHEN HE KEEPS SAND-DOMES, HE IS SHINY AND SHINY.

(022) * WHEN HE MAKES FOUR PROUD FIRE-BOXS, HE IS PROUD OF FOUR FIRE-BOXS AND *SMALL.

(023) * WHEN HE KEEPS FOUR BELLS AND THE SAND-DOMES UNDER STEAM, FOUR TRAINS AND FOUR SHINY AND BIG WHISTLES, *ENGINEER *SMALL HAS A POLISHED TRAIN AND BIG WHEELS IN HIS FOUR HEATED DRIVING WHEELS.

(024) * A FIRE-BOX IS PROUD OF *SMALL.

(025) * STEAM IS HEATED.

(026) * WHEN A HEATED SAND-DOME IS HEATED, IT KEEPS FOUR ENGINES, A POLISHED AND BIG FIRE-BOX AND A FIRE-BOX.

(027) * HE IS BIG, POLISHED, HEATED AND BIG.

(028) * WHEN WATER IS POLISHED, STEAM IS HEATED.

(029) * A POLISHED, SHINY AND SHINY DRIVING WHEEL MAKES A FIRE-BOX, FOUR WHEELS, HIS FOUR OILED, BLACK AND BLACK WHEELS AND A POLISHED BOILER POLISHED.

(030) * WHEN HE IS OILED, HE IS SHINY AND BIG.

(031) * WHEN HE IS PROUD OF WHEELS, HE HAS THE SMOKESTACK UNDER A BELL.

(032) * HE IS OILED.

(033) * ITS WATER MAKES THE WATER IN *SMALL.

(034) * WHEN WATER IS OILED, HE IS POLISHED.

(035) * *ENGINEER *SMALL MAKES IT AND LITTLE FIRE-BOXS.

(036) * WHEN A POLISHED, LITTLE AND OILED SMOKESTACK IS SHINY AND LITTLE, HE IS OILED.

(037) * WHEN HE MAKES FOUR OILED, PROUD, LITTLE, SHINY AND SHINY SAND-DOMES OILED, POLISHED, SHINY AND BLACK, *SMALL IS LITTLE AND BLACK.

(038) * HE MAKES BIG AND BLACK TRAINS, THE FOUR TRAINS, FIRE-BOXS, HIS

DRIVING WHEEL, THE OILED SAND-DOMES AND OILED, PROUD, POLISHED AND POLISHED ENGINES PROUD OF HEATED AND OILED DRIVING WHEELS AND FOUR POLISHED AND BIG BELLS.

(039) * WHEN *ENGINEER *SMALL KEEPS A BELL HEATED AND LITTLE, A PROUD SMOKESTACK HAS HIS FOUR POLISHED TRAINS.

(040) * WHEN A SAND-DOME KEEPS FOUR WHEELS PROUD OF DRIVING WHEELS AND THE WATER, THE STEAM HAS *ENGINEER *SMALL.

(041) * *SMALL IS PROUD OF WATER.

(042) * WHEN HE IS PROUD OF IT, *SMALL KEEPS HEATED FIRE-BOXS SHINY AND BLACK.

(043) * A HEADLIGHT MAKES THE TRAINS, ITS SAND-DOME, *SMALL, *SMALL AND *ENGINEER *SMALL.

(044) * THE STEAM IS OILED.

(045) * HE KEEPS IT, *SMALL AND *ENGINEER *SMALL.

(046) * WHEN *SMALL IS OILED, HE IS PROUD OF WATER AND STEAM.

(047) * HE IS HEATED.

(048) * WHEN THE STEAM KEEPS *SMALL AND *SMALL, ITS DRIVING WHEEL HAS FOUR SAND-DOMES AND THE LITTLE, POLISHED AND POLISHED WHISTLES IN *SMALL AND *SMALL.

(049) * WHEN HE IS HEATED, WATER IS HEATED.

(050) * WHEN HE MAKES ITS LITTLE AND POLISHED BELL, *SMALL IS PROUD OF THE FOUR BELLS UNDER HIS FOUR BLACK HEADLIGHTS.

(051) * A PROUD, LITTLE, PROUD AND HEATED HEADLIGHT IS LITTLE AND SHINY.

(052) * HE HAS A ENGINE AND A PROUD ENGINE.

(053) * HE IS POLISHED.

(054) * *ENGINEER *SMALL IS POLISHED.

(055) * WHEN ITS STEAM MAKES PROUD, PROUD AND OILED FIRE-BOXS, HE IS OILED.

(056) * HE HAS OILED BOILERS, A BELL, THE STEAM, *SMALL AND *ENGINEER *SMALL.

(057) * WHEN THE WATER UNDER TRAINS HAS OILED WHEELS, FOUR POLISHED, SHINY, LITTLE, OILED AND LITTLE FIRE-BOXS AND ITS BLACK, POLISHED, PROUD, BIG, BLACK AND OILED FIRE-BOXS, *ENGINEER *SMALL KEEPS THE FOUR WHEELS

AND FIRE-BOXS BLACK, BIG, LITTLE, OILED AND HEATED.

(058) * WHEN HE IS SHINY, HE IS OILED.

(059) * WHEN HE IS POLISHED, *SMALL IS SHINY AND HEATED.

(060) * *SMALL IS LITTLE.

(061) * WHEN A HEATED, POLISHED, BLACK, HEATED AND BLACK BELL IS HEATED, *SMALL IS POLISHED SHINY AND HEATED.

(062) * WHEN HE IS HEATED, HE IS PROUD OF BOILERS AND SAND-DOMES.

(063) * WATER IS POLISHED.

(064) * WHEN HE IS POLISHED, HE HAS *SMALL AND *SMALL.

(065) * A TRAIN MAKES FOUR SAND-DOMES, STEAM, IT, THE WATER UNDER FOUR BOILERS, SMOKESTACKS AND A OILED TRAIN AND IT SHINY.

(066) * HE KEEPS FOUR FIRE-BOXS, WATER, ITS FOUR SHINY, BIG, PROUD AND HEATED TRAINS, THE BOILER AND A OILED AND POLISHED WHEEL.

(067) * WHEN HE IS HEATED, THE POLISHED BELL IN *ENGINEER *SMALL HAS BLACK AND OILED WHEELS.

(068) * WHEN HE IS PROUD OF DRIVING WHEELS, THE HEATED SAND-DOMES AND THE WATER, HE IS OILED.

(069) * IT IS POLISHED.

(070) * THE WATER UNDER THE WHEELS IN OILED WHISTLES AND ITS POLISHED, SHINY, BIG AND BIG TRAINS IS BLACK.

(071) * WHEN HE IS SHINY, IT KEEPS FOUR POLISHED ENGINES, FOUR BIG HEAD-LIGHTS AND THE TRAIN IN SHINY, HEATED AND HEATED HEADLIGHTS.

(072) * HE IS POLISHED.

(073) * WHEN HE IS POLISHED, HE IS PROUD OF SMOKESTACKS AND FIRE-BOX.

(074) * HE HAS STEAM.

(075) * HE HAS STEAM.

(076) * WHEN IT MAKES STEAM PROUD OF WATER, *SMALL IS HEATED.

(077) * WHEN WATER IS HEATED, BLACK, SHINY, SHINY, BLACK, HEATED AND BLACK, HE MAKES HEADLIGHTS.

(078) * WATER IS POLISHED.

(079) * WATER HAS LITTLE AND OILED SAND-DOMES IN A BELL.

(080) * *ENGINEER *SMALL IS OILED.

(081) * STEAM IS SHINY.

(082) * HE IS OILED.

(083) * WHEN HE IS POLISHED, *ENGINEER *SMALL IS BLACK

(084) * HE IS HEATED.

(085) * HE IS OILED.

(086) * WHEN HIS STEAM IS PROUD OF STEAM, *ENGINEER *SMALL IS HEATED.

(087) * WHEN IT IS LITTLE, POLISHED, PROUD AND HEATED, A SHINY WHEEL IS PROUD OF FOUR BIG AND BLACK WHEELS, ITS BLACK AND POLISHED DRIVING WHEELS, *ENGINEER *SMALL, ITS BOILERS AND FOUR TRAINS.

(088) * WHEN IT IS POLISHED, *ENGINEER *SMALL HAS *SMALL, A BIG SMOKE-STACK AND *SMALL IN ITS DRIVING WHEEL AND FOUR SHINY AND PROUD DRIVING WHEELS.

(089) * HE IS HEATED.

(090) * WHEN A FIRE-BOX HAS FOUR WHISTLES, HIS BELLS, *SMALL AND *ENGINEER *SMALL, HE HAS HEATED, LITTLE, LITTLE, OILED, OILED AND OILED WHISTLES.

(091) * WHEN HE IS OILED, ITS ENGINE IS BIG.

(092) * HE IS BLACK.

(093) * *ENGINEER *SMALL IS OILED.

(094) * HE IS OILED.

(095) * THE PROUD AND OILED BELL IS PROUD OF PROUD FIRE-BOXS.

(096) * HE IS PROUD OF OILED, BIG, BLACK, BLACK, SHINY, PROUD, POLISHED AND SHINY HEADLIGHTS.

(097) * WHEN THE WATER IN THE FOUR OILED AND SHINY TRAINS, LITTLE BELLS AND THE STEAM IS BLACK, IT IS SHINY.

(098) * HE IS OILED.

(099) * WHEN A BOILER KEEPS WATER, IT MAKES IT PROUD OF A BLACK WHEEL, THE STEAM AND HIS SHINY WHEELS.

(100) * HE MAKES FOUR LITTLE AND BLACK BOILERS AND THE WATER.

REFERENCES

[1] YNGVE, V. H. A Framework for Syntactic Translation, *Mechanical Translation,* 1959, 4, No. 3 59.

[2] CHOMSKY, N. *Syntactic Structures,* Mouton and Company, The Hague, 1957

[3] MATTHEWS, G. H., *Denumerability of the Sentences of Natural Languages* American Mathematical Society, New York, April 1960.

[4] YNGVE, V. H., "A Model and an Hypothesis for Language Structure," *Proc. Amer. Phil. Soc.* 1960 104, No. 5, 444

[5] LENSKI, L. *The Little Train,* Oxford University Press, New York, 1940

[6] YNGVE, V. H. "A Programming Language for Mechanical Translation," *Mechanical Translation,* 1958, 5, No. 1 p. 25.