

by

F. Curado J. A. Valente +\* Dept. Ciencia da Computacao, UNICAMP, Campinas - SP, Brazil

#### 1. Introduction

Making the computer available to children and other naive users requires attention to the structure of the computer language. Such attention has been given to the design of the language LOGO which attempts to use syntactic forms which feel natural to an English speaker. This paper discusses problems which arise when we attempt to transport into another linguistic environment the design of LOGO and experience accumulated with its use. Specifically we are concerned with making a version of LOGO for use in Brazil.

The only similar enterprize we know is the translation of LOGO into French by Guy Montpetit in Montreal [2]. However our approach is very different. Montpetit made an early decision to stay close to the English usage, especially the usage of the MIT ll-LOGO. This decision was dictated partly by practical consideration: he needed a "French LOGO" immediately, i.e. without first having to do research on this aspect of the problem. But also, the decision reflected strategy of first doing the simplest thing in order to gain experience and appreciate the problems. Since the Quebec group has now provided the experience we do not have to repeat it and have engaged in a study of the more subtle issues involved.

- + Presently on leave of absence at MIT-LOGO Group
- \* The authors acknowledge the help of Seymour Papert, MIT LOGO Group

We note that the implications of our work are not confined to LOGO alone. Although we frame all our discussion in terms of LOGO, what we say is relevant to the general problem of designing a computer language intended for non-professional users.

In this paper we intend to discuss some of the languagedependent features of the English version of LOGO (ENG-LOGO) and to show that the translation to some other language may not allow the full use of the power embedded in the English version.

We first cover the essence of LOGO principles. After an analysis of ENG-LOGO we go on to discussing some specific translation problems which we foresee as we translate LOGO into Protuguese.

#### 2. Language and Learnability in LOGO

To create a frame of reference, we first go over some ideas that should be quite familiar to the LOGO community. We feel that we will not be simply translating words: we will be translating the words that translate ideas. Hence we ought to talk about these ideas.

The design of LOGO is related to pedagogical principles in two directions which one could call: getting into it and getting something out of it [1].

The first is concerned with the initial steps toward programming. LOGO is designed to bring naive users into it in a smooth, easy, pedagogically sound way; and many features of its structure reflect attempts to do this. The second direction focuses on using computer work as a means to improve the learners ability to learn in other domains. This is also reflected in the structure of LOGO.

The two directions actually coincide insofar as the experience of learning LOGO should be a model for what good learning should be.

We concentrate in this paper on some aspect of the design of the language which involves these pre-occupations and which may be undermined by a literal translation into another language.

We should emphasize another aspect of LOGO philosophy: This is that learning something, including LOGO, must always be seen in the broader context of a mini-world with several properties:

--there must be a pool of existing knowledge; --there must be the possibility of active intellectual exploration; --the new knowledge should emerge as an amalgam of the learners' previous knowledge of the domain, his self-knowledge and powerful problem solving techniques;

The LOGO-Turtle world is a model example of such a mini-world. Focusing on it, we view ENG-LOGO as one component of a system which interrelates anthropomorphism, body syntonics, heuristics, debugging and similar concepts to create descriptions of events, things and processes. By the proper manipulation of these descriptions, learning becomes what it always should have been: a natural process of self-fulfillment.

It is very possible that a different mini-world leads to a different view of what is important in a LOGO language design. Thus we see that the problem of translating LOGO is related broadly to the expectation of how LOGO is to be used, (see the discussions of TO - DEFINE usage in 3.3 and 4.1).

#### 3. An Analysis of ENG-LOGO

3.1 The Grammar of Verbs in LOGO and English

A powerful feature of LOGO for beginners is the easy and "natural" way in which the beginners can define a new command. Programming appears in the metaphor of teaching the computer. It benefits from the child's existing model of teaching (illustrating the getting in) and in turn enriches his concept, for example, of definition (illustrating the getting out). A typical case in which LOGO resembles natural English is the definition of simple commands to produce turtle action. Consider the dialog:

? FORWARD 100	/turtle moves forward
? CIRCLE	/This will produce an error message
YOU HAVEN'T TOLD ME HOW TO CIRCLE	/This is the computer error message

The child's response is to "tell the computer how to CIRCLE". Thus:

TO CIRCLE 1 FORWARD 1 2 RIGHT 1 3 GO TO LINE 1 END

The smoothness of this process reflects aspects of English syntax. In particular the fact that the imperative used in the command

? CIRCLE

is lexically identical with the infinitive in

"YOU HAVEN'T TOLD ME HOW TO CIRCLE"

and the mixed imperative/infinite connotation of

? TO CIRCLE

In an inflected language such as Portuguese the process is less smooth. So this is one situation we shall have to worry about. Several possibilities are suggested in section 4.1.

A related exploitation of a special property of English is the use of English nouns as LOGO verbs. Thus a program to draw a house is written on the model:

TO HOUSE

-----END

موجع پوچه کری کری کری کوچه جست فرید محد شده محد شهر مرحه کری می محد به محد به محد محد محد محد محد مح	na ana ma ma ma ma ma ma ma ma					
REPORT:	TRANSLATING	LOGO	TO	Α	ROMANCE	LANGUAGE
و محکود موجد موجد به محمد منه محمد بخش بدخت مرجد مرجد ومحل خشت مرجل محمد مرجد مرجد مرجد مرجد موجود موجو						

It is not unnatural in English to use the word HOUSE as a command. We do not say it is grammatical. The sense in which it is natural is a research topic in psycholinguistics. We cannot fully appreciate the subtle meaning differences (if any exist), between the verbs TO HOUSE or TO FLAG and the use of procedures with these names which draw a HOUSE or a FLAG. The question is further complicated when we notice that TO POLYGON or TO PETAL do not exist as verbs.

The problem illustrates the complex, interdisciplinary nature of the task of designing a language. But in the meantime it is clear that there is a kind of "resonance" with the linguistic intuition of a native English speaker which is not evoked from a Brazilian ear by the literal translation. We see from this also how the multidisciplinary effort needs a "poetic" component as well as scientific component.

#### 3.2 Commands vs. Operands in ENG-LOGO

We look next at an example which illustrates yet another feature of English which is exploited by ENG-LOGO but which also shows some areas which ENG-LOGO has ignored. In making a Brazilian translation we should not see our goal merely as achieving what has been done by ENG-LOGO. A comparative linguistic study might open the door to solving problems left unsolved by the designers of ENG-LOGO.

In natural language, verbs are used to represent actions and the imperative of verbs are used to command. In the general idea of creating an "everyday life" environment, one of the features introduced in ENG-LOGO was to associate commands with the imperative forms of verbs.

This association has been done in two ways; first by choosing for primitives English terms which resemble, as much as possible, imperative forms. Appendix B presents the etymology of the commands of Appendix A. As it can be seen, the correspondence imperative-form/command can not always be followed.

The second association has been implemented by choosing the word TO to be the switch that turns on editing mode for the definition of procedures. To define a procedure is analogous to creating a new verb for the computer language, and this is used to teach the computer to do something. Since imperative forms of verbs in English are formed by deleting the word TO from the infinitive form, to use this new

ACM SIGCSE BULLETIN

OCTOBER 1978 PAGE 15

command is analogous to commanding (imperative) the computer to do what the word is meant to mean.

Operations are not commands [3] and therefore the English terms chosen to name operation primitives did not have to be imperative forms of English verbs as the list of Appendix C indicates.

This analysis shows that built into ENG-LOGO is the concept that commands are verbs and that operations may not be verbs, and a new command is created by defining a new verb.

However, in some instances we may need to teach the computer a word that is not a command. Example [3]:

TO REPLACE :LET :L1 :L2 10 IF :LET = :L1 OUTPUT :L2 ELSE OUTPUT :LET END

TO LISP :W 10 IF :W = "OUTPUT " 20 OUTPUT WORD REPLACE FIRST :W "S "TH LISP BUTFIRST :W END

TO MULTILISP :S 10 IF :S = [ ] OUTPUT [ ] 20 OUTPUT SENTENCE LISP FIRST :S MULTILISP BUT.FIRST :S END

PRINT MULTILISP[ THIS IS A RECURSIVE PROCESS ] THITH ITH A RECURTHIVE PROCETHTH

In the above example, a sequence of three recursive procedures are used to scan a sentence and to replace, in each word of the sentence, the letter S by TH. The procedures are defined as operations. They are named as verbs: REPLACE, LISP, MULTLISP. Anyway we look at it, some confusion may arise. If we stick to the imperative-infinitive schema, then we are not being consistent with the fact that operations are not commands. If we decide that operations are not verbs, an inverse argument can be built.

The example also reinforces the power of ENG-LOGO to allow quite a freedom in the choice of names one can give to procedures. MULTLISP

OCTOBER 1978 PAGE 16

is not even a word in English, but as it has been constructed as a generalization of the word <u>lisp</u>, it is natural to understand it in ENG-LOGO as a verb of the <u>lisp</u> family. It is perfectly valid for a space buff to call a procedure SPW37 or ARTOO, regardless of its suitable classification as a command, operation, switch or anything else.

One could argue that the distinction between these types of LOGO-words is not natural or necessary. As we see, it may be useful to keep the distinction as a pedagogical tool to make the point that words and concepts have different uses. On the other hand, if there were only commands, the distinction found in the present implementation would be non-existent.

#### 3.3 Mnemonics

Mnemonics is an issue that focus many of the problems faced in the design and analysis of the LOGO language. For various reasons computer languages primitives compress more information in them than the English word used does. Questions of how much information is embedded, whether or not is possible to follow a general pattern of compressing information naturally arise.

ENG-LOGO uses mnemonics in three contexts. First to represent a larger idea: FORWARD means to move forward. Secondly, to indicate unambiguously the action or operation being performed: HIDETURTLE, BUTFIRST. Finally, to allow the use of shorthand notation (abbreviations) for some frequently used primitives: FD, BK, HT, BF. One key feature of the use of abbreviations is that they are allowed to be typed on the keyboard as inputs, but are echoed in the screen by their full name.

Not much can be done in the first two cases. It is more convenient to express LOGO words as short expressions and not always can one word express the whole idea embedded in a primitive. In one case or another the only real improvement is to find a better term to describe a certain primitive. For instance, the Ediburgh LOGO Group has substituted DEFINE for TO, because they prefer to avoid the ambiguities that TO brings to LOGO.

With respect to abbreviations the situation is a little more complex. If we want to keep the "everyday life" environment then to use abbreviations is a departure from this principle. Moreover, why bother to devise terms like HIDETURTLE, which are full of meaning, if they can be as effectively substituted by cipher. Undoubtedly they are convenient for the expert programmer, but one should wonder what impact they have among beginners. The experience of the Edinburgh Group resulted in abandoning them.

A translation project must also consider all these aspects. A decision will have to be made whether to follow the present structure of ENG-LOGO or to pursue a more independent mode of action.

In summary, what this analysis hopes to have shown is that the natural language English is an integral part of ENG-LOGO. Consequently, ENG-LOGO has to reflect an English speaking culture and one should wonder how much of ENG-LOGO should be taken to another language or culture.

Finally, the analysis also indicates the difficulties in transfering concepts from a highly idiosynchratic means of communication (spoken English) into a highly structured one (ENG-LOGO), when a goal is to keep both languages quite close.

#### 4. Work Strategy for the Translation Project

The translation of a LOGO computer language to other non English contexts must be done is such a way that the translation of the primitives preserves the original LOGO principles.

Hypothetically it would be possible to develop a "new" computer language for the foreign setting by ignoring the work accumulated around the existing LOGO versions. It would have the advantage of avoiding problems of cultural transfers but it would not capitalize on the experience already available.

If we start from ENG-LOGO and try to translate it into a foreign language we can use available technology but we must analyse the points where ENG-LOGO is dependent upon cultural aspects that do not simply "translate".

In the case of Portuguese we will develop the Brazilian LOGO (BRA-LOGO), trying to capitalize what has been done at BBN, MIT,

Edinburgh and in the French version used in Canada (QUE-LOGO). We also mention that we do not discuss the German version because it uses English words for its primitives.

#### 4.1 Teach-Order Metaphor

The teach-order metaphor is used when one defines a new procedure. There are three things involved in this definition: the switch that turns on editing mode, the infinitive form which names the procedure and the imperative form which is used to refer to the procedure. As it happens in ENG-LOGO, TO not only serves as a flag to turn the switch to editing mode but also is used to reflect the teach-order metaphor through the use of an infinitive-imperative structure.

For a Romance language (of which Portuguese is an instance) the first difficult point to circumvent is to maintain procedure generation by the infinitive-imperative analogy. The imperative of a verb in Portuguese is constructed keeping the radical, deleting the infinitive termination (AR, ER, IR, OR) and adding the corresponding imperative termination (E,A,A,ONHA). If the verb is irregular (and there are many such verbs), the imperative termination may vary. For example:

English	ſ	Portuguese				
Infinitive	Imperative	Infinitive	Imperative			
TO END	END	TERMIN.AR	TERMIN.E			
TO LIVE	LIVE	VIV.ER	VIV.A			
TO LEAVE	LEAVE	PART.IR	PART.A			
TO COMPOSE	COMPOSE	COMP.OR	COMP.ONHA			

Obviously, the computer implementation of the infinitive-imperative in this setting has difficult parsing problems, but the question of the infinitive-imperative structure does not stop here. We are not sure that the everyday use of imperative forms in Portuguese has the same meaning as in English. It seems that in popular conversation we form command orders using various formulations with subtle stress differences. Besides the regular form, we have found instances of using the infinitive and composite forms (VA APANHAR, literally GO TO PICK), to order. On the other hand, if from a noun we want to form a "verb" in English, we simply add TO to the noun (even if this does not mean anything). The equivalent way in Portuguese would be to change the noun by adding a suitable termination and this is quite awkward. As a result, we feel that a LOGO translation to Portuguese will have to be done with the help of a professional applied linguist to help us to understand how imperative forms are commonly used.

There are many ways of approaching the translation. Our first instinct is to keep the teach-order metaphor by simply translating the switch TO to a suitable word, maintaining the infinitive-imperative structure. In Portuguese this will not be simple to implement. In the search for alternative procedures, we found six possible ways of implementing procedure definitions in Portuguese. To help the understanding of this process we use a notation similar to the Backus Normal Form. In ENG-LOGO we have:

TO <word> ====> <word> where <word> is a name of variable

Semantically this means that the right hand side refers to the definition of a procedure which is invoked by using the left hand side.

I. PARA <word> =====> <word> where <word> is a name of variable.

PARA is a literal translation of TO. Using <word> as a name of any variable we leave the infinitive-imperative approach; although in the case where <word> is an infinitive form of a verb, this scheme is equivalent to the use of PARA <infinitive> as a way of teaching and of <infinitive> as an order. Sometimes it gives the proper ENG-LOGO emphasis.

This implementation would be equivalent to the Quebec version.

#### II. PARA <infinitive> ====> <imperative>

First, not all Portuguese words that finish with R are verbs. Hence the interpreter would first verify that infinitive is really a valid infinitive form, and then it would have to do one of many possible checks to see whether the corresponding imperative form is REPORT: TRANSLATING LOGO TO A ROMANCE LANGUAGE

being correctly used (a linguistic verification). The parsing problems previously mentioned would be present here.

Secondly, a lot of the freedom to name procedures is lost, creating a "cook-book" approach to parts of LOGO. It seems clear that this alternative should not even be tried.

III. <char.> <word/infinitive> =====> <word/infinitive>

A variation of the cases I and II, where the switch TO is translated into a simple ASCII character that does the switch.

The teach-order metaphor has to be brought "outside" LOGO. The instructor has to say that <char.> is the symbol which allows the computer to be taught.

If it is used with the construction of I, it may be acceptable.

IV. <infinitive verb - root+R-termination> =====> <root+imperative
termination>

This is the infinitive-imperative ENG-LOGO analogy, implemented in Portuguese. The switch to invoke editing mode becomes the R-termination of the verb. Case II discusses the feasibility of this approach.

V. PARAQUESE <word> =====> <word> where <word> is a name of variable

In this case, TO is translated by a sequence of words (PARA QUE SE) which lead, in Portuguese, to the use of an imperative form as the name of <word>. The idea of commands being imperative is maintained but, as we are not checking the linguistic validity of <word> for its being imperative, we are back to the free interpretation of the analogy, as in case I. It has also the disadvantage of using three words to form a composite LOGO word.

VI. <reserved word> <word> ====> <word> where <reserved word> :

APRENDA (LEARN) , or ENSINE(TEACH), or DEFINA (DEFINE), etc. and <word> is a name of variable.

We try to keep the teach-order metaphor using a different name for the editing switch. The choice of the proper word needs more thinking and some experiments with possible candidates. In the case of LEARN or TEACH there are subtle emphasis-meaning problems. DEFINE seems to be more impersonal and more computer oriented. It has the advantage of avoiding awkward situations that are present even in ENG-LOGO (TO POLYGON). This approach also eliminates the commandoperation duality of certain user written procedures which only return values. Unless some serious reservations about this approach are raised, it seems that BRA-LOGO should go in this direction, specially because the teach-order metaphor is explicitly introduced if LEARN or TEACH is used.

#### 4.2 Translation of the LOGO Primitives

Under the point of view of translation, primitives present special features, which range from technological to metaphorical. The actual translation of words is generally a simple task, provided that the main general questions of language design and use have been understood.

The technological problem arises when the new natural language uses a set of characters different from the American one. In Portuguese this set is a little different and this difference already imposes heavy burdens. If we go to languages like Greek, Russian or Japenese the situation is much worse.

Portuguese has one different letter  $\zeta$  (the squiggle was added afterwards by hand!), and four characters used to modify the sound of letters ( $\prime$ ,  $\backslash$ ,  $\land$ ,  $\sim$ ). Character sets in Brazil come from foreign manufacturers who do not use these characters, and character sets appropriate for Portuguese are not very common.

Sometimes the use of these characters is fundamental and the case of translating MAKE shows this quite clearly. MAKE translates naturally into FAÇA. If Ç is not available we must use C instead, and the word becomes FACA. If FACA were not a Portuguese word, that might be adequate, but FACA means KNIFE. The consequences are obvious.

Adult users normally accept these discrepancies, although even for them the lack of special characters may introduce ambiguities that are hard to solve (as in the case of E - AND and E - IS). For the younger user these problems should be totally avoided and one must develop the proper technology to solve them: modification of keyboards, character codes, etc..

Software and metaphoric problems are present when we look at the etymology of LOGO words, at the use of their abbreviations and at the usage of noise words. Many actions that should be performed by LOGO commands or operations can be well expressed by only one word. Although FORWARD really means move forward, one easily gets by without move. The same does not happen with HIDETURTLE, BUTFIRST, etc.. In ENG-LOGO composite words are formed either using full words or abbreviating one of them (BTOUCH, XCOR, EMPTYP, etc.).

This seems to indicate that the heuristics used for LOGO words are something like:

- a) find a natural language word which represents the meaning of the action
- b) if not possible, find two or more words to do it: if the resulting composed word is to long, abbreviate one of them.
- c) test it for usability.
  - cl) if acceptable, end
  - c2) return to a)

Two problems arise. First the choice of the proper word(s) to convey the idea. Example: Does BUTFIRST really express the idea of all the elements of a list except the first one? Second when the composite word is too long we must decide which way to abbreviate it. Example: BACKTOUCH is too long and BTOUCH can be made to mean the same thing, but maybe BACKT would be acceptable, as in EMPTYP.

It is then clear that the dependence of the LOGO language on a natural language is important in order to retain the "everyday life" objectives of the environment in which LOGO is expected to operate. REPORT: TRANSLATING LOGO TO A ROMANCE LANGUAGE

4.3 Abbreviations and Noise Words

As we have discussed previously, to use abbreviations in a LOGO environment may take the learner out of the general idea of "life-like" communication with the machine. In one sense, abbreviations are third level translations.

If they are used (and we are not sure if they should be), their formation should follow precise rules to avoid things like:

CLEARSCREENCSPRINTPRPRINTOUTPOPRINTSCREENPRS

The use of abbreviations should also follow ENG-LOGO implementation which accepts abbreviations but prints out full words. Following the Edinburgh experience, it seems advisable to maintain LOGO short words so that we do not need to introduce beginners to abbreviations.

With noise words the opposite happens. They are used to make communication more "life-like". Example: PRINT FIRST OF MUMPS, where MUMPS is a procedure which outputs a list. OF was originally used in BBN-LOGO, but the MIT llLOGO version dropped it, as well as other noise words (keeping only THEN).

The decision to use or not to use noise words has to refer to the level of adoption of anthropomorphic ideas, although the introduction of some of them in Portuguese may also create nasty difficulties. For instance, OF THE in Portuguese can be DA or DO.

#### 4.4 Error Messages

Error messages are important feedback information that show how well or poorly the language is understood by the users. One is then faced, initially, with the task of anticipating the kinds of errors that beginners will make, and this will depend on how this language is implemented.

nan ene and					
REPORT:	TRANSLATING	LOGO TO	A ROMANCE	LANGUAGE	

When a computer language is specified, the designer must have models of the user and of the way this language is going to be used. These models will condition him to establish the kinds of most common errors and the form of the corresponding error messages.

In LOGO, the use of anthropomorphic ideas will also have to be brought into consideration because they will induce the learner to certain kinds of behavior which will have to be caught by an appropriate use of error messages.

One valid alternative is to try to separate error messages into syntactic and semantic ones. Another is to create error messages which will indicate to the user that the statement is ambigious. A third alternative is to follow a hierarchy of most common errors.

To separate, a priori, what is a semantic or syntactic error is quite difficult. Even in the case where one decides to impose more stringent grammar rules in order to facilitate error analysis, languages of the class of LOGO are powerful enough to give to the user room to "find" a case where the error system does not work. The case of FD100 versus FD 100 in ENG-LOGO serves to illustrate the points above mentioned. FD100 may be a procedure name (semantics) or a typing error (syntactics). If one decides to restrict procedure names to be formed only with letters (a more constrained grammar) then it is also possible to accept easily FD100 or FD 100 as valid LOGO statements.

To create error messages that give alternative ways of looking at the error requires ingenuity and constant updating of the score of messages. It is technically simple to implement, but one has to carefully maintain the balance between too little or too much information to the user. In the case of FD100, a message of this kind would be: "YOUR STATEMENT WAS NOT UNDERSTOOD. YOU EITHER FORGOT A BLANK OR HAVE YET TO EXPLAIN WHAT FD100 MEANS".

The third alternative can be implemented simplisticly or in a more eleaborate manner. The simple approach would concentrate on the most common errors (something difficult to foresee in a completely new language). The elaborate manner would analyze the error in a tree fashion and sequentially present the user with alternatives.

We do not know yet how to proceed. It will depend upon the need to rewrite or simply adapt the BBN-TOPS 10-LOGO that is now available in Campinas. It will also reflect the experience that the UNICAMP-LOGO Community will have as LOGO is used more and more with children and adults.

#### 5. Conclusion

We hope to have made clear our contention that LOGO languages may be quite dependent on particular features of the corresponding natural language. The consequences are twofold. First, one should take a close look into ENG-LOGO to analyze these dependencies in order to decide which ones should be capitalized on and which ones should be avoided.

Secondly, one should approach a translation to a foreign language very carefully, trying to profit from what was seen in ENG-LOGO and from what can be used from the foreign natural language structures.

The analysis of ENG-LOGO also shows some important features of computer languages in general, which seem to have not been well studied. Issues of COMMAND vs OPERAND, and information compressing in primitives must be dealt with if one wants to understand how people employ (or should employ) computers to solve problems. The translation of LOGO to Portuguese may help to clear some of these points.

Finally, we hope that the ideas brought forth in this paper will be useful to the American LOGO Group interested in creating a Spanish version of LOGO for use in the U.S..

#### Bibliography

- []] Papert, S. "Notes for a LOGO Book"; unpublished notes; August, 1977.
- [2] Montpetit, G. "Un Glossaire des Primitifs LOGO"; unpublished paper; Groupe LOGO, College Edouward, Longueil, Canada; 1973.
- [3] Abelson, Goodman and Rudolph "LOGO Manual"; Cambridge, Mass: LOGO Memo 7; 1974.
- [4] Goldenberg, E.P. "A Glossary of PDP 11 LOGO Primitives"; Cambridge, Mass: LOGO Memo 16; 1975.

and						
H	REPORT:	TRANSLATING	LOGO	TO A	ROMANCE	LANGUAGE
	an acce vize foie dine acce que the					10 Data dago gan yan ban may kan dan dan dan dan ban dan dan dan dan dan dan dan dan dan d

## APPENDIX A

Basic ENG-LOGO primitives used by beginners [4].

General purpose		
EDIT GOODBYE REQUEST MAKE	END PRINT TO TEST	ERASE READ WRITE
Turtle purpose		
BACK PENDOWN CLEARSCREEN HIDETURTLE SETX, SETY SHOWTURTLE LAMPOFF RTOUCH	FORWARD PENUP HEADING HOME XCOR, YCOR BTOUCH LAMPON TOOT	LEFT RIGHT HERE NODISPLAY WRAP FTOUCH LTOUCH
List purpose		
BUTFIRST EMPTYP SENTENCE	BUTLAST FIRST WORD	COUNT LAST
Numeric purpose		
DIFFERENCE LESS RANDOM	EQUAL PRODUCT REMAINDER	GREATER QUOTIENT SUM
Program control purpose		
GO IFTRUE	IF STOP	IFFALSE

	an ann ann an	entr ette wen ette mit ette ette ette	- 1000 AND 1000 1000 4000 4000 4000 4000 4000 400	دي هي من جي هي من	-
<i>REPORT</i> :	TRANSLATING	LOGO TO	A ROMANCE	LANGUAGE	
100 1000 1000 1000 1000 1000 1000 1000	ه هم هم منه منه منه منه منه منه منه منه			nam 1980 939 1989 1980 1980 1985 1986 1986 1986 1985 1985 1986 1988 1988 1988 1988 1988 1988 1988	-

### APPENDIX B

The etymology of the ENG-LOGO commands from Appendix A.

Verb

Non Verb

GOODBYE

FORWARD

LEFT

RIGHT

Unclear

PENUP

PENDOWN

EDIT ERASE PRINT PRINTOUT(\*) READ WRITE BACK HIDETURTLE(\*) HOME SETX, SETY(\*) SHOWTURTLE(\*) TOOT MAKE TEST

\*jointed two words

ACM SIGCSE BULLETIN

### APPENDIX C

The etymology of the ENG-LOGO operations from Appendix A.

Non Verb

Unclear

Verb

XCOR, YCOR(\*) HEADING HERE BTOUCH(\*) FTOUCH(\*) LTOUCH(\*) RTOUCH(\*) BUTFIRST(\*) BUTLAST(\*) EMPTYP(\*) FIRST LAST SENTENCE WORD REMAINDER RANDOM GREATER LESS

REQUEST SUM

PRODUCT

QUOTIENT

DIFFERENCE

EQUAL

COUNT

\* jointed two words