The Natural Natural Language Understander

Henry Hamburger, UCI and NSF
and
Stephen Crain, U of Texas

This study of natural language comprehension by natural understanding systems (children) is based on a procedural analysis represented in the form of a programming language. To clarify what is cognitively required for a child to respond appropriately to certain expressions in English, we show how these forms can be translated into procedures in a high-level programming language. It is then possible to discuss two kinds of difficulties a natural language form can present to the listener: (i) incompatibility of the form with its associated procedure and (ii) complexity of that procedure. An example of procedure complexity is the nesting of loops, whereas a contributor to incompatibility is a word or contiguous phrase that corresponds to separated pieces of the procedure. We shall present evidence of both types of difficulty from experiments with children and will compare the predictions of our procedural view with those of a less detailed syntactic explanation that has been advanced for a subset of the phenomena.

Many cognitive tasks can be expressed either as a natural language command or as a programming language procedure. In many tasks requiring some elementary mathematical knowledge (e.g., counting), the cognitive procedure appears to be substantially more complex than the syntactic structure of the command. In such tasks, an explanation of children's difficulties can be pursued more profitably in the realm of cognitive procedure than of syntactic structure or parsing.

To take a particularly simple example, the verb 'count' has the capacity to serve as either a transitive or an intransitve verb, but it is probably never the first such verb encountered by a child; compare 'let's eat' and 'eat your cracker.' Therefore the transition from intransitive use of 'count' to its transitive use involves no syntactic innovation for a child. But despite being syntactically ordinary, 'count' presents complexities both in its procedure and in translation to that procedure. Just ask the next three-year-old you see to count, and then to count a few objects. Chances are good that you will get errors on the transitive (latter) task but not on the intransitive one. A look at these errors will give some perspective on what a correct procedure must do.

One kind of attempt at transitive counting involves blithely swinging a finger past the objects to be counted, while apparently counting essentially intransitively, with no attempt to coordinate individual objects to individual numbers. A somewhat more sophisticated performance has the finger stopping at some or all of the objects, but in imperfect coordination with the numbers. These misperformances, we believe, arise and persist not out of ignorance of the lexical item 'count' or the associated data structure of positive integers, nor from the syntax of transitive verbs (which, it was suggested above, has already been learned), but because of difficulty inherent in forming a correct procedure for the task of transitive counting.

Consider procedures i and ii, intended to represent correct procedures for the two kinds of counting, expressed at a coarse enough grain that individual statements can be taken as having theoretical content. The tokens are intended to be more or less self-explanatory. In the interest of simplicity, no mention is made of initialization, stored data, or output.

```
i procedure: 'Count.'
     repeat                The procedure runs
       next number         through successive numbers
     until fail            until there are no more.
```

```
ii procedure: 'Count them.'
     repeat
       next object         At each cycle, the next
       next number         object and number are
     until fail            noted.
```

For i to be utilized in forming ii, it is necessary to insert new material, specifically 'next object,' into the loop already present in i. Thus one approach to accounting for the difficulty posed by the transitive case is to hypothesize that breaking into a loop is a difficult or low-priority move for acquisition or comprehension. A related view is implicit in iii where it can be seen that a single word, 'count,' corresponds to two separate pieces of procedure. This phenomenon can appropriately be called translational discontinuity in the sense that it pertains to the relationship betweeen two representations. Since it arises in the translation from a (very) high-level language (English) to a less-high-level language (of procedures), it can also be called a compiling discontinuity. Note that there is no discontinuity in the phrase structure of the language form itself.

```
iii procedure: 'Count ...'
      repeat
        ...
        next number
      until fail
```

To take these ideas a step further, consider the phrase 'the second green ball.' Roeper (1972) found that many children interpret this phrase as if 'second' and 'green' each modify the noun in the same way, that is, as if the phrase meant something like 'the ball which has the properties of being both green and second.' Matthei (1978) expresses this observation as a distinction in syntactic phrase structure, assigning the adult interpretation the phrase structure '(second (green ball))', and the non-adult interpretation '(second green ball).' The idea of using syntactic phrase structure in this way to encode semantic interpretations must rely on some assumptions about how semantics relates to syntactic structure, for example, a compositional semantics tied to the syntactic phrase marker. In any case, no such phrase-structural distinction is possible for phrases like 'second ball,' for which Matthei found substantial corresponding errors: 36% of responses (four- and five-year-olds, mostly) interpreted 'second ball' as 'the one that is both second and a ball,' as opposed to the 52% who made the similar error on the example above that had an adjective in the phrase.

Pursuing the procedural approach with this construction, we now find not only compiling discontinuity but also nested loops. These aspects of the procedure appear both with and without the adjective, as can be seen with reference to iv and v, again omitting initialization and any data structures.

```
iv procedure: 'second ball'
     repeat
       repeat
         next object
       until
         pred ball
       next number
     until
       pred two

v  procedure: 'second green ball'
     repeat
       repeat
         next object
       until
         pred green
         pred ball
       next number
     until
       pred two
```

The task environment is a display of several objects in a row, with a left-to-right ordering clearly communicated in advance. The child is asked to 'take the second ball' from a display of balls and boxes, or to 'take the second green ball' from a display of red and green balls. To take the former case, suppose that the first object is a box. Then even if the second object is a ball, so that it is both second and a ball, it is still not the second ball. Procedure iv correctly makes this distinction (which eludes so many children) by means of nested loops in which successive objects are checked for ballhood in the inner loop and counting proceeds in the outer only in case the current object is a ball.

With an adjective present the appropriate procedure is v, which is like iv except that the inner loop is exited only in the event that two successive predicates are satisfied. Rather than introduce an 'and' operator in v, we have allowed the possibility of multiple exit tests, thereby making conjunction look simpler than disjunction. Even so, one still should expect the extra predicate to add some processing burden and indeed the phrase in v does lead to more errors than that in iv. Furthermore the outer predicate should also impose a processing burden. Suppose we remove it and interpret the absence of any exit test as signifying 'repeat forever or until system breakdown (say by failure of a 'next' operator).' What is then left of v is a procedure for 'count the green balls,' which does indeed appear to cause children less difficulty than 'second green ball.' (Matthei regarded the two as cognitively equivalent).

Not only do the words 'second' and 'ball' together translate into nested loops, but they do so in a discontinuous manner. This is so because 'second' is responsible for the outer loop and 'ball' (in iv) is responsible for the inner loop. Worse yet, the programming statement 'next object' is implicit in the counting loop, so that under an absolutely left-to-right control structure, the state of (cognitive) affairs after 'second' is processed would be as in vi. Processing the remainder of the phrase requires, on this model, locating 'next object' in the existing loop, using it ('next object') in constructing a new loop, and finally nesting the new loop in the existing one.

```
vi procedure: 'second ...'
     repeat
       ...
         next object
       ...
         ...
       next number
     until
       pred two
```

It is our view that these complexities are what make such phrases difficult for children. Not only does this account indicate where complexities lie, but in addition it provides an account of how specific errors can arise from a straightforward attempt to avoid breaking into loops and incurring compiling discontinuity. Suppose that a child simply tacks on the appropriate test ('pred ball') at the end of vi to form vii. This is possible if the convention introduced in v, above, is used again here: allowing a sequence of 'pred' statements as a compound exit test. This ploy yields precisely the observed incorrect result whenever it is possible (when the second object is a ball), and an infinite loop otherwise.

```
vii procedure: 'second ball'
               (incorrect interpretation)
      repeat
        next object
        next number
      until
        pred two
        pred ball
```

To gain perspective on the earlier examples and raise some new issues, consider the phrase 'second biggest ball.' Here the ordering along which 'second' is to be counted is based not on position but on size. This ordering must be at least partially determined by the subject, using an appropriate algorithm. The usual sorting algorithms (ripple, bubble, Shell, etc.) are probably poor models both because they are severely nonparallel and because they provide a complete ordering where only a partial one is needed.

In addition to some sorting, this task demands memory of some order relationships that have been established by that sorting. These size relationships in short-term memory must be used in concert with the ordering of positive integers held in long-term memory. This requirement is not present for the earlier tasks since the positional ordering is continually available from the display. In our experiments the display has been a card with a row of pictured objects, so subjects cannot create a physical order. If separate physical objects were used, then a child's sequence of moves might reveal use of a specific sorting algorithm.

Results of pilot experiments suggest that 'second biggest ball' is, as one would expect from these considerations, an extremely difficult phrase to interpret. We devised the most straightforward we could that would test comprehension of this phrase: all the objects were identical except in size; all but two were of the same small size; the remaining two were both substantially larger than the smaller ones, adjacent to each other and noticeably different from each other in size; neither was in second position. In experiments so far the error rate has been 80%; we will report more comprehensive testing at the conference.

It is possible to set up more complex displays in which different miscomprehensions lead to distinct choices as response. With balls and boxes of various sizes there exist arrays in which, say, one object is both second biggest and a ball but is not the second biggest ball; or in which the second ball is the biggest; etc. A welter of possibilities exists for testing the way in which ordinals, superlatives, relative adjectives, absolute adjectives, nouns and relative clauses are comprehended by children and what the course of development looks like, in terms of the kinds of procedures we have posited here. Such developmental sequences are, in turn, the raw material for theories of an acquisition device.