The acquisition of a miniature artificial language under various conditions of feedback

Lester H. Gilbert, B. Sc. (Hons.)

Thesis submitted to the Department of Psychology, University of Cape Town, in partial fulfilment of the requirements for the degree of Master of Science in Psychology.

Cape Town
South Africa
April, 1978

THE ACQUISITION OF A MINIATURE ARTIFICIAL LANGUAGE UNDER VARIOUS CONDITIONS OF FEEDBACK

We have but two theories of
language acquisition at the
moment . . . the miracle theory
or the impossible theory.

G. A. Miller
quoted in Glucksberg et al (1975)

Acknowledgements

Professor P. D. du Preez, my supervisor; for introducing me to a view of Psychology and the problems of mind and behaviour which has stimulated and shaped my thinking.

Paula Briffa, for assistance, encouragement, and more encouragement since the earliest beginnings of this project.

My colleagues and students in the Department of Psychology, and the secretarial staff; particularly Dr. A. Abramovitz on whom I inflicted my first computer programs; for their assistance, encouragement, and patience.

Contents

Part II

Part III

Contents--tables and figures

Abstract

The merits of special purpose computer languages for process control are discussed; it is proposed that the FORTRAN language is an adequate and generally available vehicle for the programming of on-line real-time experiments in Psychology, when used with the techniques which are described.

A broad selective review of the literature on the acquisition of language by the young child yielded two questions which required investigation: the apparent incapacity in principle of current learning theory to account for such acquisition, and the apparent ineffectiveness of explicit reinforcement and feedback for accelerating language learning. An answer to the first question is sketched by a general theory of behaviour (du Preez, 1975), which is outlined, and a discussion of the necessity for maintaining correct levels of discourse in explaining a phenomena.

An experiment was carried out to provide evidence bearing upon the second question, the effects of informational feedback in language learning. The experiment illustrates the computer programming techniques discussed earlier, but primarily serves to demonstrate the kinds of studies which, in line with the general behavioural theory, may fruitfully explore language acquisition. Such studies, it is proposed, should employ miniature artificial languages and computer-controlled environments.

University subjects were required to learn a miniature artificial language, and were given various combinations of semantic and syntactic informational feedback about their errors of production. The language dealt with the description of simple geometrical objects presented on a visual display screen, while the subjects interacted with the computer via a keyboard. The results indicated that while the subjects paid attention to, and processed the feedback, they were not able to effectively utilise the information, and learned the language from the models which were presented every alternate trial. Some weak effects of syntactic feedback were found, and it was noted that the syntax of the language was acquired before the word meanings. The results support Cazden's (1965) findings of the use made of expansions and models by children.

Aim of the present study

Until recently, studies of language acquisition have been almost exclusively concerned with the structure of the child's utterances, under the powerful impact of Chomsky's Transformational Generative Grammar (TGG). A shift of attention has taken place, however, in the realisation that there is more to the child's language than simply its structure. Recent studies have focused upon the function of the child's utterances, the context in which these have taken place, and the nature of the interaction between the child and his environment. These studies, both early and recent, have provided descriptive accounts of the competence of the child at some particular time; but as far as can be seen, there have been no studies which address the (psychologically more interesting) question of the processes which underlie the acquisition of linguistic skills.

It must be said immediately that there has been considerable discussion about possible processes of acquisition, and interesting suggestions have arisen from the studies of structure and function; the point is simply that there is extraordinarily little data which bears directly upon what sorts of acquisition processes are active in what sorts of ways.

Clark and Clark (1977) comment about our knowledge of the influence of adult speech upon the child's acquisition that "there have been virtually no studies on this aspect of acquisition" (p. 329).

> Fodor, Bever, and Garrett (1974) state the lack more strongly:
>
> we need a theory of how the perception-production system is internalised--a kind of theory which thus far psycho-linguistics has been quite unable to provide. . . . . many of the deepest problems about the psychology of language converge, ultimately, on the problem of how language is learned. . . . . The area is wide open for future research, and its potential significance is enormous (p. 513)

Objectives

The lack of data on acquisition processes may be seen to be partially dependent upon the lack of an adequate theoretical and metatheoretical framework with which to view such processes, and the lack of a methodology with which such processes may be investigated. One of the aims of the present study is to suggest a coherent theoretical, metatheoretical, and methodological framework for studies on language acquisition.

The second aim is to illustrate the proposed methodology with an experimental study that bears upon one of the theoretical problems: the role of feedback in language learning.

The experiment which was conducted was controlled by a computer, and the third aim is to present the results of some research into the use of computers for the on-line control of experiments in psychology. The techniques adopted to achieve this are of independent interest.

Structure of the thesis

These three aims are dealt with in three parts of the thesis. The first part deals with the theory and practice of computer-controlled experiments.

The second part deals with the literature of language acquisition, and presents the theoretical and metatheoretical framework.

The third part describes an experiment and its results which illustrate the proposed methodology, and which provides some evidence bearing upon the question of feedback in language learning. The experiment was computer-controlled, utilising the techniques discussed in part one.

The second aim is to illustrate the proposed methodology with an experimental study that bears upon one of the theoretical problems: the role of feedback in language learning.

The experiment which was conducted was controlled by a computer, and the third aim is to present the results of some research into the use of computers for the on-line control of experiments in psychology. The techniques adopted to achieve this are of independent interest.

Structure of the thesis

These three aims are dealt with in three parts of the thesis. The first part deals with the theory and practice of computer-controlled experiments.

The second part deals with the literature of language acquisition, and presents the theoretical and metatheoretical framework.

The third part describes an experiment and its results which illustrate the proposed methodology, and which provides some evidence bearing upon the question of feedback in language learning. The experiment was computer-controlled, utilising the techniques discussed in part one.

Computer-controlled experiments in psychology--theory

The use of on-line computers in a psychological laboratory raises four areas of discussion: the architecture of the computer itself; the associated computer hardware for the control of experimental environments; the computer programs, or software, for process-control; and the point of using a computer system for the control of experiments.

Computer architecture

The choice of a particular computer system has, in the past, depended to a large extent upon available finances; but the continuing drop in the prices of computing equipment allows a more rational approach to the selection of an optimal system. An incorrect selection of equipment will have wide repercussions for the ease of use of the

system, the necessity for expert technical staff, and its suitability for student use and training.

The most important aspect of the computer's architecture is, I would suggest, the I/O interrupt system. The running of experiments in real-time requires a quick, flexible response to input and output; also, accurate timing functions must be available, and these are most satisfactorily arranged through the interrupt system.

The most desirable interrupt system is one which is vectored with priority. This means, firstly, that each device is allocated an interrupt priority, such that devices with higher priority can obtain immediate service before devices of lower priority. The clock is usually allocated the highest possible priority in the interrupt hierarchy. Secondly, when a device interrupts, the computer architecture activates the necessary interrupt handler directly; that is, the interrupt is vectored to the proper subroutine. The alternate procedure, polling, involves the computer software in a search for exactly which device it was that generated the (general) interrupt; a usual concomitant of the polling procedure is that the whole interrupt system is disabled relatively longer. These two aspects make accurate timing very difficult indeed in a non-vectored interrupt system.

Interface hardware

The three aspects of process-control which differentiate the area from data processing in general are the timing functions, control of external devices via non-standard procedures, and information about external events.

Timing functions are obviously best provided by some form of

external clock which can be attended to automatically. Software clocks are unable to provide the satisfactory timing of an unpredictable event; and for the same reasons, an external clock which does not operate through the interrupt system cannot do so either. The emphasis here is on the satisfactoriness of the timing, not upon the question of whether or not it is actually possible. A sufficiently ingenious programmer is usually capable of providing almost any function, but at the cost of considerable effort and specificity of operation.

The non-standard control of external devices, typically lights and motors, has uniformly been achieved through the use of reed relays in applications in psychological laboratories. The advantages of the reed relay for this purpose are, firstly, that the environment is electrically isolated from the computer's expensive electronic circuits; and secondly, only an elementary knowledge of electricity on the part of the user is needed to connect any device to the computer. The disadvantage of the reed relay is its low power rating; this may be overcome by inserting a second relay with a higher power rating between the device and the reed relay, but the closure time of these relays is usually both relatively long (between 20 and 50 ms) and somewhat variable (plus or minus 10 ms). An alternative solution is to introduce power transistors, but then the advantages quoted above are lost.

Information about external events is usually obtained in one of two ways in most applications: from a keyboard which is part of a teletypewriter, visual display unit, or printer; or from an interface board which is capable of sensing switch closures in the environment. This latter solution is, of course, more flexible and

more powerful. It has the same advantages of those associated with the use of reed relays in output: the computer is isolated from the environment electrically; and a user with only elementary knowledge of electricity can connect switches to the computer. An important aspect of the sense switch interface board is that it does not necessarily require the electronic circuitry to allow the generation of an interrupt, provided that the system includes a clock which does generate interrupts. As the chapter on the practical control of experiments shows, all I/O to the environment may be done while the clock interrupt is being serviced. This allows for remarkably simple programming, and simplified electronics on the non-standard I/O interface boards.

Process-control software

While the literature has not been overly concerned with details of computer hardware in the control of experiments in psychology, many articles and reports have been written on details of the desirable software for the on-line, real-time control of experiments. The issue involved, in those cases where an issue is taken, concerns the creation of special-purpose languages for on-line control. Most experimenters are agreed that certain minimum functions need to be programmed in Assembler, and that certain kinds of experiments need to be programmed entirely in Assembler; therafter, however, a wide variety of control languages has been proposed for use by psychologists. The list includes ACT (Millenson, 1971), APCOL (Pinkus & Gregg, 1973), CAN-4 (Durell, 1972), DTES (Spear, Overgard, & Christian, 1975), EMPP (Burkhardt, 1975), FOCLAB (Mullen, 1976), HECL (Doll, 1972), HEPS (Schneider, 1973),

PACER (Henry, 1976), PEPL (Getty, 1975), PROSS (Scholz, 1972), PSYCHOL (McLean, 1969), PSYCLE (Creelman, 1971), SCAT (Polson, 1973), SIMPLE (Aaronson & Brauth, 1972), SKED (Snapper, 1975), and SP-12 (Pitz, 1975). Some experimenters have, on the other hand, directed their efforts towards the use of already-existing computer languages as vehicles for experimental control: BASIC, FOCAL (e.g. Link, 1975), and FORTRAN (Haber, Barry, & Uhlman, 1970; Kaplan, 1977).

In general, formal terms, it is quite certain that a special-purpose language is better suited to its domain of convenience than is a general-purpose language. The most widely known computer language in scientific computing is FORTRAN; this general-purpose language is nonetheless oriented towards computation and data analysis, and there is a strong presumption that it is not particularly well suited to tasks of process-control and general data processing. The presumed deficiencies of a general-purpose language such as FORTRAN have been the motivating factors behind the creation of some of the special languages mentioned above; for the other, the simple absence of any higher-level language supplied by the computer vendor has been the motivation.

I wish to argue here that there is very little to be gained from the creation of a special-purpose process-control language when a FORTRAN compiler is already available (and which includes the COMMON and CALL statements). My remarks are primarily directed towards the single-user computer system; time-shared, multi-user, or multi-subject systems require special operating systems. Even in these latter situations, however, suitable modification of the procedures suggested in the chapter on practical on-line control, along the lines of Kaplan (1977), would allow the use of FORTRAN with no loss of adequacy.

The essential differences between a special-purpose language and FORTRAN for the control of processes lie in two areas: the better efficiency and compactness of the special language, and its provision of the necessary timing functions and non-standard I/O. As I have shown in the chapter on on-line practice, some very simple procedures may be adopted with FORTRAN programs to provide all the necessary timing and I/O functions; the addition of these procedures to some computer system would represent a completely negligible allocation of time and effort by comparison with that involved in the design and creation of a special language. On the issue of efficiency and compactness, it is certain that a well-designed special-purpose language would be better than FORTRAN in general, but it is also likely that there would be some loss of flexibility and some imposition of restrictions which would not be present with FORTRAN. Some trade-offs would probably occur which would leave only restricted data analytic functions and limited standard I/O formatting.

The acquisition of better efficiency in a special purpose language is at the cost of considerable time and effort on the part of the user in developing the language. I would suggest that the FORTRAN language is efficient enough for almost all purposes; I have not yet encountered an experimental situation where inherent limitations in the language have prescribed alternative approaches. This being the case, the time and effort of the user would be better spent in his primary activity--conducting interesting experiments.

The use of a computer in a laboratory

A prior question could be asked about the whole enterprise of the on-line control of experiments, whether genuine advantages are

obtained, or whether the computer is simply a bright new toy for the enthusiastic experimenter. Sidowski (1976) has put this question somewhat irreverently, and concluded that, whatever the formal merits of on-line control, "much of the activity associated with laboratory digital computer use must . . . be labelled 'nonnutritive sucking.' It appears that, for many owners and users, the substitute object intimacy of the pacifier is more attractive and satisfying than the real thing" (p. 54). In a more serious vein, Aaronson, Grupsmith and Aaronson (1976) make the point which has struck me so much: "(computers) have made possible various experimental procedures that were otherwise impossible or nearly impossible" (p. 131). The particular research reported here on the learning of a miniature artificial language could not have been contemplated without the use of a process-control computer. The reason is that the computer can implement very complex procedures and decisions in a short time, without error. In the present example, it was hardly conceivable that a human experimenter would be able to provide the necessary feedback to the subjects quickly and without error during an experiment of about 160 trials and taking about an hour.

Computer-controlled experiments in psychology--practice

The Department of Psychology at the University of Cape Town acquired a Hewlett-Packard 2114B mini-computer in early 1971. The mainframe was equipped with the maximum memory, 8k (16-bit) words, and four interface boards. The only peripheral at the time was an ASR 33 Teletypewriter (TTY), with its I/O interface board. The three other items were a programmable clock, called a time base generator (TBG); a reed-relay output board, containing 16 relays which could be set under program control; and a one-off Hewlett-Packard-built board which could sense and report the state of 16 on/off switches. Within a few months, the system was enhanced by the addition of a fast paper-tape reader, and a fast paper-tape punch. I became involved with the running and programming of the machine within a short time, and the first priority was the development of the necessary techniques and programs to allow the on-line real-time

control of some of the experiments currently in progress, or being contemplated in the Department.

The literature at the time, while aware of the impact of the mini-computer in the psychological laboratory, was somewhat short on actual details. McLean (1969) was the first to describe a specific system; but it ran on a 16k computer with mass-storage. Haber, Barry, and Uhlman (1970) described the use of FORTRAN on a PDP-8; but this was a 4k machine with an impoverished FORTRAN compiler, and Haber et al were mainly concerned with by-passing its limitations. Creelman (1971) described a system in more general terms using a PDP-8; but his solution of writing an operating system to get the most out of his machine represented an investment of time and effort which was not possible. In short, the literature of the time was not particularly encouraging, and in common with many other programmers, I was forced to "re-invent the wheel" (Pitz, 1975) and develop the necessary techniques which I was sure were already being developed elsewhere. As matters have turned out, this was not quite accurate; to return to the metaphor, none of the published "wheels" so far seem suitable for our "wagon". I am concerned, of course, that other "wagons" may benefit from the particular "wheel" which I have designed.

Programming in Assembler and BASIC

My first efforts at programming the computer for the control of experiments were prosecuted using the Assembler provided by Hewlett-Packard. For very simple experiments, these programs were quite successful; but the more complex procedures which I then tackled proved to be extremely time-consuming to write and de-bug. The experiment which was the subject of these explorations was the partial replication

of Howells' (1944) work on colour-tone synesthesia; the experiment was part of the Ph.D. research of Dr. Arnold Abramowitz (1972). The finished program, when written in FORTRAN, comprised about 270 lines of source code, and exhausted the full 8k memory of the computer; when written in Assembler (Gilbert, 1971) the equivalent program was about 3000 statements long and very difficult to handle, despite extensive segmentation and sub-routines.

The possibilities of BASIC were explored after the rejection of Assembly-level programming. The language had the attraction of easy use and quick de-bugging; but the first simple application revealed two faults. Firstly, the timing of external events proved to be quite inaccurate, while secondly, there was only limited memory available for program code. This failure left FORTRAN as the next, and last, vendor-supplied vehicle for the on-line control of experiments, before I was committed to the construction of a special-purpose process-control operating system.

The problem of timing and control

The initial programs illustrated a problem which a more general programming system would need to solve: the integration of the clock with the control of the experimental environment and the timing of external events. The particular problem was the timing of some input event, such as a key-press, given the available hardware.

Conventional computer science procedures suggested that each I/O device should have the ability to interrupt the computer, with an interrupt handler subroutine to either accept or transmit the necessary information. The point of this is to allow the computer to be busy

with other processing while the I/O transfer takes place. When the information which arrives is unsolicited, however, then this scheme makes it difficult to see how a currently running process would be able to utilise the new information (such as timing it) without either a monitor system or simply maintaining a loop condition until the information was available. This latter procedure, of course, negates the whole point of the interrupt in the first place.

The notion of the computer simply waiting in a loop until the arrival of an expected but unpredictable input naturally led to an exploration of alternatives where such a situation would not occur. These alternatives constituted a null set; that is, by the very nature of the real-time on-line control of an environment, the computer will spend most of its time simply waiting for some event or other before proceeding with some change to the environment. This being the case, there was very little point in using the sophisticated interrupt architecture of the computer to monitor an event. It must be realised that there was only one environment to control; where two or more independent environments are involved, then the use of the interrupt system and a monitor program are strictly necessary.

These considerations led to the construction of three Assembler subroutines: one to obtain the current time of day, a second to set the relays to some specified pattern, and the third to examine and store the state of the switches in the environment. These would be used by a main program, written in FORTRAN, via CALL statements. No effort was made to provide for an interrupt when a switch was turned on, for the reason outlined above. A very similar scheme has been proposed by Kaplan (1977). The subroutines were never in fact used; a simpler and more elegant solution was implemented, as described below.

Programming in FORTRAN

(In the interests of readability, certain of the syntactic features of FORTRAN will be ignored in the discussion to follow. In particular, all variables will be of type INTEGER, unless the first letter of the variable name is "X".)

A FORTRAN program for the control of any experiment defines three variables:

TIME containes the current time, and its value is incremented by 1 every 10ms. TIME is thus a counter of hundredths of a second. The maximum value which TIME can reach is 327.67 seconds.

RELAY contains the desired pattern of relay settings. Each bit in the word designated by RELAY corresponds to a reed relay on the interface board, such that 0=off and 1=on. For example, to turn relay 3 on, the third bit of RELAY must be 1; this may be achieved by setting RELAY=4 in the program. (Decimal 4 = 100 in binary.)

SWITCH contains the pattern of the switch settings in the environment. Each bit of SWITCH is either 0 or 1 depending upon the state, off or on, of the corresponding switch in the laboratory. A response button or key might be connected to switch number 5, for example; when pressed or closed, SWITCH will assume a value of 16 (binary 10 000).

The values of these three variables are updated every 10ms by an Assembler subroutine called TIMER. That is, the pattern of relay settings specified by RELAY is applied every 10ms, while the state of the environment switches is examined and stored in SWITCH.

These three variables are declared to be in COMMON in the FORTRAN main program, and in the Assembler subroutine. The TIMER is started by

a subprogram CALL, and thereafter the values of TIME and SWITCH are defined by TIMER, while RELAY is defined by the main program.

Example 1. Every FORTRAN program will have the two statements at the beginning:

```
      COMMON TIME,SWITCH,RELAY
      CALL TIMER
```

Example 2. A wait of 450ms is typically accomplished as follows:

```
      TIME=0                          Set counter to zero.
100   CONTINUE                        Loop until the counter
      IF(TIME-45)100,110,110            reaches a value of 45.
110   .                               Carry on with processing.
      .
      .
```

Example 3. Calculations or I/O may take place while the wait interval is being timed, providing care is taken that these activities do not exceed the required interval. Suppose that the program needs to establish which stimulus to present, randomly selected from 10 stimuli, on the next trial. This could be done during the wait as follows:

```
      TIME=0
      CALL RANDOM (X)                 Obtain a random digit
      I=X*10.0 + 1.0                  and convert it to a
      STIM=ARRAY(I)                   pointer between 1 and 10.
100   CONTINUE
      IF(TIME-45)100,110,110
```

Example 4. The laboratory environment is manipulated by turning

items of equipment on and off. A light connected to reed relay number 4 may be turned on by the single instruction:

```
        RELAY=8
```

Decimal 8 equals 1 000 in binary; the on bit, 1, is in position four, thus turning relay four on. The light will come on within 10ms.

Example 5. The response of a subject may be detected by looping on the value of SWITCH, which will contain a non-zero value whenever a switch is closed. The subject may be provided with a push-button, or a key, which is connected to the sense switch interface board. The closure of this switch will be detected within 10ms by:

```
120     CONTINUE                        Loop until SWITCH is
        IF(SWITCH)120,120,130           non-zero.
130     :
        :
```

Example 6. These elements may be assembled together to provide a program which will measure the reaction time (RT) of a subject to a light. Assuming that no other switch closures are expected except that of the subject, and that the light is connected to relay number 3, the RT may be timed to an accuracy of 10ms as follows:

```
        RELAY=4                         Select relay 3.
        TIME=0                          Set counter to zero.
130     CONTINUE                        Wait for the switch
        IF(SWITCH)130,130,140           closure.
140     RT=TIME                         Capture the value of
                                        the counter
```

Example 7. Various lights and equipment may be turned on and off independently by manipulations of the value of RELAY:

```
          RELAY=8                    Turn on relay 4
          RELAY=RELAY+2              Turn on relay 2 as well.
          RELAY=RELAY-8              Turn 4 off, leave 2 on.
          RELAY=0                    Turn all relays off.
```

Example 8. Multiple time intervals can be conveniently managed by allowing TIME to run freely, and then establishing future points at which to take action:

```
          START1=TIME+50             Begin something in 500 ms.
          START2=TIME+200            Begin something else in 2s.
          :
140       CONTINUE
          IF(TIME-START1)140,150,150
150       :
          :
160       CONTINUE
          IF(TIME-START2)160,170,170
170       :
          :
```

Example 9. Multiple inputs require the use of a masking function to strip the unwanted bits from SWITCH and leave only the bits of interest. This may be done with the AND function:

```
      MASK1=8                 Mask to expose bit 4 only.
      MASK2=32                Mask to expose bit 6 only.
      MASK3=40                Mask to expose bits 4 and 6 only.
      :
180   CONTINUE                Loop until either switch 4
      1=AND(MASK3,SWITCH)     or switch 6 (or both) is
      IF(1)180,180,190        closed.
190   J=AND(MASK1,SWITCH)     J codes switch 4 on or off.
      K=AND(MASK2,SWITCH)     K codes switch 6 on or off.
```

These examples illustrate the use of the three basic control variables TIME, RELAY, and SWITCH. The advantage of the above procedures is that the Assembler subroutine is completely transparent to the user. Full and complete control of the most complex experiment may be obtained from the FORTRAN main program, without the need for special function calls or the transfer of control to any other sub-programs.

The appendix contains examples of FORTRAN programs which I have written to control quite complex experiments. These programs have no particular value in themselves, except as examples of the general procedures which have been outlined above. There is one program from the research of Abramowitz (1972) into colour perception; one program from the research of Oxtoby (1971) into the perception of time; and one program from the research of Marshall (1975) into schedules of reinforcement. These programs are documented more fully in their respective sources. Finally, of course, there is the control

program for the present investigation into the learning of a miniature artificial language.

The Assembler subroutine TIMER

The current version of the general purpose Assembler subroutine is listed below. For particular experiments, it may be, and usually is, modified to suit particular requirements; but the general nature of its use in the FORTRAN main programs will be quite similar.

The Hewlett-Packard programmable clock operates through the computer's interrupt system, and generates an interrupt at the end of the interval which it has timed. The interrupt handler must then re-start the clock in order that another interval may be timed. The length of the timed interval may be programmed; it may be 100us, 1ms, 10ms, 100ms, 1s, 10s, 100s, or 1 000s. The most useful interval has been 10ms; this is accurate enough for almost all experiments, and involves an overhead of 50us in the interrupt handler, less than 1% of the available computer time.

The accuracy of the clock has been checked periodically with a Venner Digital Counter TSA 6636, and with Hewlett-Packard diagnostic programs. On 1ms intervals, over a 1 000ms period, the error in timing was 1ms at most.

```
ASMB,R,B
      NAM TIMER
      ENT TIMER
      COM TIME, SWITCH, RELAY
+
TIMER NOP
      ISZ TIMER          Establish the return address to the
      LDA MS10           main program.  Obtain the required
      OTA TBG            timing interval, and set the clock.
      LDA IJSB           Establish the interrupt linkage.
      STA TBG
      STC TBG,C          Start the clock.
      JMP TIMER,I        Return to the main program.
+
      ORB
IJSB  JSB LINK,I         Interrupt linkage; this vectors the
LINK  DEF CONT           interrupt to the interrupt handler
      ORR                called "CONT".
+
CONT  NOP                Interrupt handler entry point.
      STC TBG,C          Re-start the clock immediately.
      STA SAVEA          Save the contents of the A register.
      LDA RELAY          Obtain the on/off pattern of the
      OTA ROUT           relays, and set this on the interface.
      LIA SWIN           Obtain the on/off pattern of the input
      STA SWITCH         switches, and store.
      ISZ TIME           Increment the time counter.
      NOP                (Guard against a zero value)
      LDA SAVEA          Restore the contents of the A reg.
      JMP CONT,I         Return to the interrupted process.
+
TBG   EQU 10B            The clock is in I/O slot 10.
ROUT  EQU 15B            The relay interface is in slot 15.
SWIN  EQU 14B            The switch sense board is in slot 14.
MS10  OCT 2              Define the 10ms interval setting.
SAVEA BSS 1
+
      END
```

Practical considerations

There is a variety of circumstances where the procedures discussed above need to be implemented with care. The more important of these are mentioned here.

The I/O control programs of the Hewlett-Packard implementation of FORTRAN, while effective, are somewhat primitive; in particular, the I/O is not buffered, and the I/O control program turns off the interrupt system while I/O decisions are made. The first problem means that a READ or WRITE statement will not be executed until all previous READ or WRITEs have been completed. Thus, any control statement such as an IF will not receive control until the I/O has been completed. Where accurate timing is required, the second problem means that no I/O should be done while a timing loop is in progress. In practice, these two difficulties do not cause much of a problem; typically, the I/O which is done within a timed wait loop is of a stereotyped nature, and takes the same amount of time at each execution. A nominal interval of 500ms would thus in fact become a consistent interval of, say, 510ms.

The Hewlett-Packard 2114B does not possess hardware multiply and divide functions; consequently, these functions and others such as exponentiation take some time to complete. Where possible, these should be located outside of a timing loop, or located within a loop of quite generous dimensions.

The maximum value of an integer is 32 767; incrementing this value by one will yield -32 767. Care is needed to ensure that TIME never overflows in this way; if it does, the computer will appear to "freeze" for about five minutes.

The value of TIME should always be captured by a simple assignment statement such as RT=TIME before being used. Otherwise, if the variable TIME itself is used in a calculation, its value could change considerably before the calculation is completed.

For some experiments, switches in the environment may change state, and change again, within 10ms. Also, a switch may only be on for a short time such that it is missed by the FORTRAN main program. In such cases, the Assembler subroutine would be modified to provide timing intervals of 1ms, and to provide a latch or flag which maintains any change in the value of SWITCH until tested or cleared by the main program.

Language acquisition--what is acquired

An examination of the processes of language acquisition has two prerequisites: a characterisation of language itself, and a working knowledge of what structures and functions are acquired by the child.

Language from the standpoint of a Psychologist

Linguists, and indeed psycholinguists, have tended to view language as being composed of three strata: semantics, syntax, and phonology (Halliday, 1975). This does not, of course, prevent them from discussing other aspects of language, such as its functions; but these three levels are seen to characterise "language" quite well. Philosophers have been much attracted by the symbolic aspect of language, and consequent questions about signs, reference, and meaning (Russell, 1940). Behaviourally inclined psychologists have, on the other hand, been attracted by the function of language, and have tended to prefer

discussion of "verbal behaviour" or "speech" (Skinner, 1957). Cognitively orientated psychologists and computer scientists have placed an emphasis upon process in the understanding of natural language (Dalenoort, 1973; Winograd, 1972) which has not been reflected elsewhere.

I would like to suggest an analysis which integrates these various conceptions; such an analysis would have no virtue apart from the assistance it gives in organising the wide variety of studies of language and its acquisition. This analysis will be used to organise the following discussion of what the child acquires when he is said to acquire language.

Analysis of studies of language

Structure:
- Syntax. Surface regularities; word order.
- Semantics. Meaning of words; elements of meaning

Function:
e.g. Tacts and mands (Skinner, 1957); speech acts (Searle, 1971).
- Instrumental, regulatory, personal, etc. (Halliday, 1975).

Process:
- Comprehension and production of sounds.
- Comprehension and production of semantics.
- Acquisition of structure, function, and process. Discovery procedures; operating principles.

Surface structure

As mentioned earlier, most studies of the child's competence have focused upon the grammatical structure of the child's utterances, particularly when these utterances were comprised of two words. The most influential of the earlier studies was probably that of Braine (1963);

using data on the distributions of the words in the child's utterance, Braine suggested that the two-word utterance is comprised of two classes of word, pivot and open. The two-word utterance would usually either be of the form Pivot + Open, or of the form Open + Pivot. The words in the pivot class were restricted in number, while those in the open class were much more numerous. Under the impact, if not the direct influence, of Chomsky's (1957) TGG, the structural description of the two-word utterance is given by Braine as:

$S \rightarrow P_1 + X$

$S \rightarrow X + P_2$

$P_1 \rightarrow$ allgone, byebye, more, my, see, etc.

$X \rightarrow$ boy, sock, boat, milk, shoe, mommy, . . .

$P_2 \rightarrow$ on, off, etc.

Essentially similar findings were reported by Brown and Fraser (1963), and Miller and Ervin (1964). Considerable dissatisfaction has been expressed with these accounts of the child's first "sentences"; the criticisms have been particularly associated with the work of Bloom (1970) and Bowerman (1973). The major difficulty is that the bare structural description of pivot and open class words does not properly capture the richness of the child's utterance; in particular, the same surface structure may be ascribed to quite different "deep structures". Bloom's (1970) well-known example is the utterance Mommy sock, which occurs in two quite distinct contexts: when the child picks up her mother's sock, and when the mother puts the sock on the child. An analysis of Mommy sock as $P_1 + X$ misses the fact that the meaning and the intention of the utterance is quite different in each context; presumably an analysis of an utterance is required which quite clearly includes its function and meaning. Bloom's analysis involves the "rich interpretation" of an utterance,

examining its context and its presumed meaning, in order to give a "deep structural description" more in line with those given by TGGs.

Bloom's (1970) critique may be seen as a turning-point in the investigation of the child's language. Thereafter, researchers entertained a far broader conception of the child's language, and began asking pertinent questions about the language functions of an utterance, the context of an utterance, and the meaning of an utterance (Slobin, 1972).

Function

A functional analysis of language has, of course, been the principle mode of examination of behaviourally-inclined psychologists, exemplified by Skinner's Verbal behaviour (1957). This particularly operant approach has not won a position for itself in psycholinguistics, nor is it much evident in the area of investigation defined by the Journal of verbal learning and verbal behaviour. Speculation upon the reasons for this are not strictly relevant here. (But are nonetheless interesting; two possibilities are the apparently devastating critique by Chomsky (1959), and the rather difficult and implausible hypotheses themselves as advanced by Skinner.) The point which I wish to make is simply that a functional analysis of language is nothing new to psychology, though it may be to psycholinguistics.

Much more relevant functional analyses, for the purposes of developmental psycholinguistics, have been made by Searle (1971) and Halliday (1975).

Searle is associated with the notion of the "speech act":

> it is essential to any specimen of linguistic communication that it involve a (speech) act. It is not, as has generally been supposed, the symbol or word or sentence, or even the token of the

> symbol or word or sentence, which is the unit of linguistic communication, but rather it is the production of the token in the performance of the speech act that constitutes the basic unit of linguistic communication.
>
> (emphasis as in the original, p. 39)

Halliday (1975) has defined, and examined the emergence of, some six functions:

> (1) The instrumental function is the function that language serves of satisfying the child's material needs, of enabling him to obtain the goods and services that he wants. . . . .
>
> (2) The regulatory function . . . is the function of language as controlling the behaviour of others . . . .
>
> (3) The interactional function . . . is language used by the child to interact with those around him . . . .
>
> (4) The personal function . . . is language used to express the child's own uniqueness; to express his awareness of himself . . . .
>
> (5) . . . the child can turn towards the exploration of the environment; this is the heuristic function of language . . . .
>
> (6) The imaginative function . . . is the function of language whereby the child creates an environment of his own.
>
> (Emphasis as in the original, p. 20)

The vital importance of these views of language is that an essential continuity is proposed between the pre-verbal, the proto-linguistic, and the language-using child. Halliday's analysis of his child, Nigel, graphically illustrates the growth of language from, and its roots in, the sensori-motor (Piaget and Inhelder, 1966) period of the child's development. What has been captured by Halliday is the essentially natural unfolding of the child's linguistic competence based on his cognitive and social

development.

Process

The processes by which the stream of speech is segmented on input, and assembled on output, are not strictly relevant to the present study. It is important to note, however, that these processes are not particularly simple. Mention must be made of the theory of distinctive features of phonemes (Chomsky & Halle, 1968), the theory of the identification of speech sounds by analysis-by-synthesis (Halle & Stevens, 1962), and the perception of continuous speech by active processing along lines similar to the analysis-by-synthesis (Neisser, 1967).

The processes by which sentences are comprehended for their meaning, or produced on the basis of some intended meaning, have mainly been the province of computer scientists working on artificial intelligence. These processes are startlingly complex; the studies of machine translation in the early 1960's seemed to fail for two main reasons: a naivety about what was involved in understanding an isolated sentence, and the fond belief that there would only be minor mopping-up to be done after the sentence had been analysed syntactically along the lines of Chomsky's TGG.

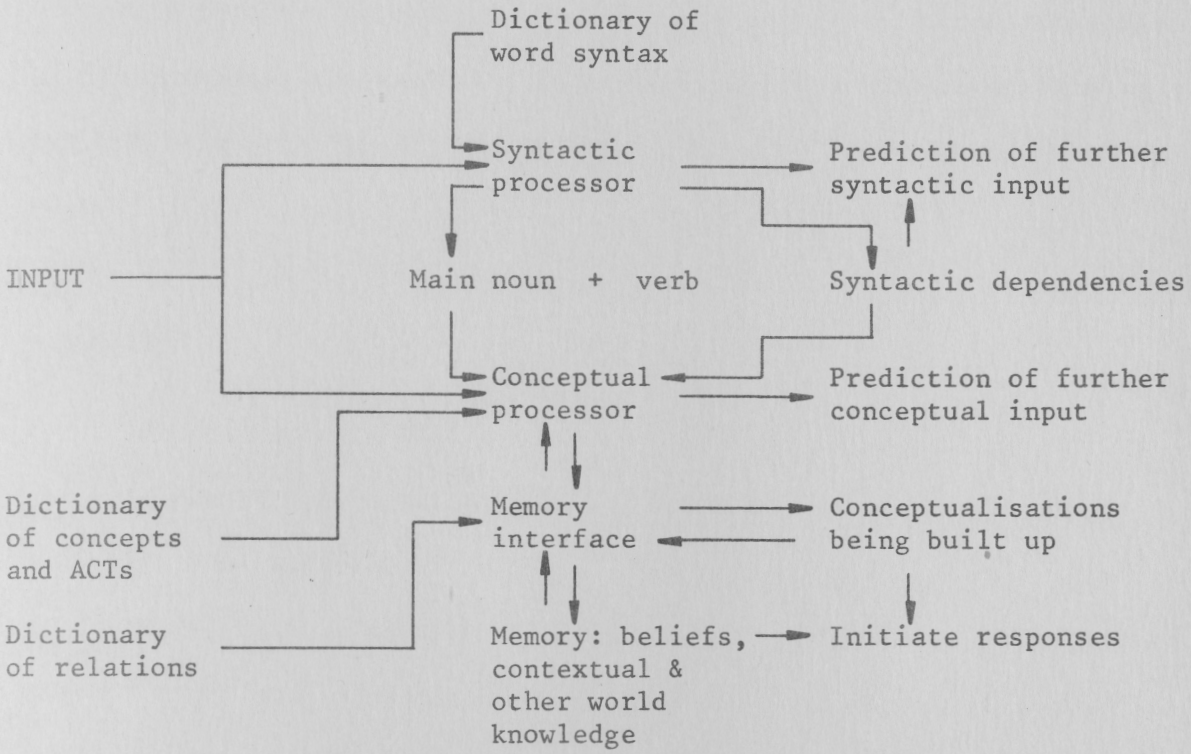
The more recent and psychologically relevant studies have shown that comprehension (and production) rely upon the hearer's knowledge of the context and probable intentions of the speaker, the hearer's vast "knowledge of the world", besides the syntax of the language and the way linguistic functions are expressed. Comprehension is the result of a complex interaction between these knowledge systems: it does not seem to arise from a simple linear process, say, syntactic analysis followed

by semantic analysis followed by examination of the context; rather, the tentative results of a semantic analysis are fed to a syntactic analysis, which refers back to the semantic system and then forward to the "world knowledge" system, for example, before the sentence is finally understood.

The most influential theories of comprehension are those of Schank (1972), and Winograd (1972). It is important to realise at the outset that these theories are in fact stateable as computer programs, and were intended as such. It is thus necessary that some care be taken to distinguish between aspects of the theories which are dictated by considerations of the computer implementation, and those aspects which claim to reflect hypothetical psychological processes; it is also necessary that any irritation about the former do not cloud any judgements about the virtues of the latter. In the case of Schank (1972), for example, a number of separate dictionaries are involved in comprehension, whose separate psychological status is uncertain; or, there is a specific detail that a "triple" is passed to the verb-ACT dictionary for processing, a detail which is presumably rather unimportant for psychologists at their current state of understanding of the whole process of comprehension.

A flowchart of Schank's model (or theory) is given on the next page. His central concept is that of conceptual dependency, specifically that between an actor and an act. The input sentence is analysed syntactically and conceptually, simultaneously and interactively, to yield the main noun and verb. The analysis of the sentence proceeds with the aid of various dictionaries and stored information from memory about context and the world. Of particular importance is the point that the process of understanding depends in a fundamental way upon the prediction of future input. These predictions are of the syntactic and

conceptual form of the incoming sentence, and serve to guide and shape the process of understanding.



Winograd's (1972) system differs in many details, but the same quality of processing is assumed to take place. A more important point of difference will be examined in the section on semantics; for the present, it is that Winograd's approach is within the framework of procedural semantics, while Schank's approach may be characterised as structurally semantic.

A process theory of production has been advanced by Schlesinger (1971a, 1971b) which bears certain similarities to the theory of Schank (1972). The central notion that Schlesinger uses is that of intention; the speaker is assumed to have a network of intentions or concepts, a "deep structure" which can give rise to a wide variety of utterances. Which utterance or series of utterances are actually realised depends upon the application of one or other realisation rules. These rules

are learned by the child in its interaction with the linguistic environment; specifically, the relation between an utterance and its intention or meaning defines the realisation rule which was used to arrive at the utterance from the intention. The child is in (independent) possession of (understands) the speaker's intention, and hears the utterance; he has the necessary information to infer, and thus learn, the realisation rule.

Semantics

Semantic aspects of language have been identified at each of the three levels of language: structure, function, and process. Until recently, most analyses of "meaning" dealt with the structure of meaning in terms of semantic elements (e.g. Katz and Fodor, 1963), while the then current functional accounts of meaning in terms of a conditioned mediating response (Mowrer, 1960) seemed, at the least, quite implausible to psycholinguists (Fodor, 1965). The issue which these theories addressed was that of how a word may be said to have meaning, and current theorising is in terms of semantic elements (e.g. Clarke, 1973). This is only a partial view of the problem of meaning, however, in that it cannot provide a fully coherent account of how a sentence or utterance may be said to have meaning. The work of investigators into artificial intelligence and the understanding of natural language has shown the very close relationship between the meaning of an utterance on the one hand and syntax, function, and process on the other. The accounts of meaning fall under the general heading of procedural semantics (Johnson-Laird, 1977).

The meaning of a word, in the theory of semantic elements (Clark and Clark, 1977), is given by its collection of elementary semantic components. These elements or components are considered to provide a

list of the attributes which comprise the word meaning, and are stated as "atomic" propositions. For example, the meaning of boy is specified by the list /+Male/, /-Adult/, and /+Human/. A number of physical (as well as social and psychological) dimensions may be characterised by poles; for example, height has the poles Tall and Short. In order to provide a semantic component for, say, boy, two possibilities arise: the focus may be placed upon Tall, in which case the component is /-Tall/; or the focus may be placed upon Short, when the component is /+Short/. The choice, however, is not arbitrary; it is natural to say He is six feet tall rather than He is six feet short. This illustrates the notion that Tall characterises the whole dimension of height, and is called unmarked; while /+Tall/ and /-Tall/ characterise the pole of the dimension, and Tall is marked in this case. The point of this detail has been illustrated by Clarke (1973), where the child, in his acquisition of the meaning of words is shown to first use the word Tall in its unmarked sense before using it in its marked sense.

The meaning of a sentence may be claimed to be given by the meanings (semantic elements) of the words in that sentence; such a claim is falsified by the simple pair John bit the dog and The dog bit John. An alternative, and potentially very powerful, view is given by procedural semantics. The procedural account was first fully brought to the attention of psychologists by Winograd (1972) in his discussion of natural language understanding. The account is developed by analogy with the way in which computer languages are typically implemented: a computer program is generally first compiled (scanned for syntactic regularities and assigned a structure) and then executed by the computer, such that the output or results from the program is said to define the meaning of the program. This may seem, at first, to be a violation of the notion of

meaning; the full power of the approach may be best appreciated by considering the understanding of questions. Take the question Is the book on the table?; the view asserts (roughly) that the hearer understands this question by compiling it into a structure and then executing this structure to obtain the products. That is, the question defines a program which is then executed such that the results of the execution define what the hearer understands. There are several virtues to this approach: the most important one is that the distinction between comprehension and production falls away to yield an integration of the two processes (Johnson-Laird, 1977). This may be seen from the way in which the question above is understood by the hearer; the very process of understanding naturally generates a structure which provides the required answer. A second important outcome is that the hearer's understanding is based in a fundamental way upon his capability for acting on the environment. It is suggested that the child's first "programs" are sensori-motor programs, and the development of the child's ability to understand utterances is the development of his ability to compile and execute increasingly complex and abstract programs to provide firstly, the understanding of the world and then the understanding exemplified in concrete and formal operations (Piaget and Inhelder, 1966).

The metatheory of language acquisition

The previous chapter presented a selective review of studies dealing with what the child acquires when it acquires language. Before proceeding with the particular focus of this study, the processes of how language may be acquired, it is necessary to first establish the feasability in principle of such an exercise. Part of the heritage of Chomsky's TGG and its remarkable impact upon psychology is the still lingering view among psycholinguists that language is not learned so much as it is acquired through the mediation of some genetically pre-programmed language acquisition device (LAD; Chomsky, 1968). LAD will be examined for its merits in the next chapter; the purpose here is to justify the thoroughly empirical metatheoretical position of the present study by analysing and answering the nativist position on language acquisition. The champions of the nativist view are taken to be Fodor, Bever, and Garrett (1974); the following presentation of this view relies heavily

upon their analysis (Fodor et al, 1974, pp. 452 - 462).

To begin with, Fodor et al (1974) make it clear that the point at issue is not whether there are genetic components in language acquisition (since there certainly are such components, in common with perception and learning), but how much such genetic components account for in language acquisition. The nativist view argues that most, if not all, language acquisition is mediated by a genetic endowment. Again, the issue is not that the environmental input counts for nought (since the child could not possibly acquire the language otherwise), but that an account of acquisition cannot be provided by current theories of learning. The major arguments are as follows.

The nativist position

1. There is a considerable degree of dissociation between cognitive and linguistic development. If true, this statement removes any link which may be claimed between the cognitive capabilities of the child and his linguistic competence, such that the competence depends upon the cognitive capabilities. Presumably a broad psychological account, if not a learning-theoretic account, would wish to have the developing cognitive functions mediate the development of language. The essential data for this statement of dissociation derives from Lenneberg's (1964) examination of the relationship between speech development and IQ. Lenneberg claims that there is a low correlation between these two variables; an examination of his Figure 4 shows, however, a very direct and clear relationship between standard (mental) age and stages of language development. That is, the quality or stage of language that the child reaches may be predicted almost perfectly from a knowledge of the child's mental age (and vice versa).

2. There exist linguistic universals, "profound formal and substantive similarities between all natural languages" (p. 455). But then Fodor et al (1974) undermine their own argument by saying "we have no a priori grounds for rejecting the suggestion that there are linguistic universals because there are genetic universals" (missing italics added; p. 455); by the same token, unless I misunderstand the force of their argument, there are equally no grounds for accepting the suggestion. That is, the existence of linguistic universals, if true, is neither evidence for or against a hypothesis of the genetic mediation of language acquisition (Putnam, 1968).

3. Reinforcement or explicit teaching plays little part in the acquisition of language by the child. The most well-known study bearing on this problem is that of Cazden (1965, but inaccessible; summarised in Brown, Cazden and Bellugi, 1969). This point is indeed a thorn in the side of an empirical account, and forms the basis of the experiment reported in this study.

4. The complexity of what is acquired (internalised) cannot be accounted for without the postulation of innate mechanisms. This is the central argument of the nativist position; Chomsky (1959) puts it very well:

> The fact that all normal children acquire essentially comparable grammars of great complexity with remarkable rapidity suggests that human beings are somehow specially designed to do this (p. 171 in Jakobovits & Miron, 1967).

There are two aspects to this argument. The first is its logical force, while the second is its indictment of the (apparent) poverty of current theories of learning. The logical force is dealt with by Putnam (1968):

> In the absence of any knowledge of what general multi-purpose

> learning strategies might even look like, the assertion that such strategies . . . cannot account for this or that learning process, that the answer or an answer schema must be 'innate', is utterly unfounded. (original emphasis; p. 138 in Searle, 1971)

Very similar views are expressed by Black (1970).

A general theory of behaviour

The second aspect of the nativist argument needs a more serious and reasoned rebuttal, since it states quite simply that current theories of learning (and presumably, by extension and implication, of psychology) are incapable of accounting for the broad and vital area of language acquisition. In the following section, I will outline a general theory of behaviour, incorporating current learning theoretic views, that is sufficiently powerful for the analysis of complex behaviour such as language and its acquisition. This general theory has been stated, in a somewhat different form, by Du Preez (1975) and incompletely by Kintsch (1970); the following account is based upon these sources.

The first question for such a general theory is the broad shape that it will assume. The problem of what counts as an explanation in science has occupied philosophers of science for many years; the orthodox position (Braithwaite, 1953), briefly put, is that an event is explained when it may be deduced from a "covering law" and a statement of the antecedent conditions. However explanation is conceived, it may be recognised that there are various levels of explanation; for example, the hardness of a diamond may be explained by a theory of crystalline structure, which in turn may be explained by atomic structure, which in turn may be explained by subatomic (particle) structure. Polanyi (1968) has argued very clearly that in such a descent to lower levels of

explanation certain aspects of the "to be explained" event are completely lost. The lower levels of explanation provide laws which relate to the event, but which do not in any way state how these laws are integrated:

> Mechanisms, whether man-made or morphological, are boundary conditions harnessing the laws of inanimate nature, being themselves irreducible to those laws. . . . . Further controlling principles of life may be represented as a hierarchy of boundary conditions extending, in the case of man, to consciousness (p. 1312)

For the current problem, it is clear that the "laws" of learning theory, while relating in some way to language acquisition, do not and necessarily cannot state how the self-same laws are to be assembled into an integrated explanation. This explanation belongs to a next-higher level, at the very least, which will presumably enjoy its own (different) concepts and terminology.

These considerations dictate a general theory of at least two levels (in fact five levels will be employed), which takes the laws of learning theory as the lowest level. For the present purposes, these basic laws are taken to be those of operant conditioning (Skinner, 1969), where the unit of behaviour is defined as the three-term contingency: an operant, which occurs in the presence of a discriminative stimulus, and which has consequences called a reinforcing stimulus.

The second level comprises the stimulus-sampling theory of Estes and his colleagues (Estes, 1950; Neimark & Estes, 1967). The laws of operant behaviour are assembled into a theory which accounts for (relatively simple) learning and forgetting.

The formalised stimulus-sampling theory has been shown to be asymptotically representable (modelled) by a finite automaton by

Suppes (1969). The importance of this lies in the fact that the TOTEs of Miller, Galanter, and Pribram (1960) are also formally representable by finite automata (Millenson, 1967; Suppes, 1969). The importance of this is that the TOTE formalism has been accepted by Chomsky (Miller & Chomsky, 1963) as being capable in principle of accounting for the sorts of behaviour which underlie language comprehension, production, and acquisition. This is a crucial level in the proposed theory, in that it links S-R theories with the cognitive theories in whose terms language is usually discussed:

> The advantage . . . (is) that we may find it convenient to represent complex sequences of behaviour by means of TOTE hierarchies and yet account for the acquisition of the operations in these hierarchies by learning theory. (du Preez, 1975, p. 30)

The fourth level has already been introduced; it is the requirement that TOTE hierarchies are required to be able to account for even moderately complex behaviour. These hierarchies have been called plans by Miller et al (1960), who give examples of their use in accounting for behaviour.

The fifth, and final level for present purposes, is not strictly needed for an account of language learning; it is nonetheless presented because the mature use of language may be viewed in its terms. The important concept at this level is that of a state of multiple simultaneous readiness to respond. Such a state defines a conscious organism for Taylor (1962; 1968). The notion of simultaneity is important in the account of conscious activity (Neisser, 1967) as is the notion of readiness for response or action (Arbib, 1972). In addition, it provides an escape from the problems associated with a strict serial S-R account of behaviour (Chomsky, 1959 , who quotes

Lashley's (1951) famous paper; MacCorquodale, 1970, who notes Skinner's (1957) use of this notion).

Implications for methodology

The general theory of behaviour has been presented in some detail, since it occupies a central place in psycholinguistics, and in the present research.

It has long been assumed by psycholinguists that mainstream (behaviourist) psychology is unable to provide a satisfactory basis for any accounts of language, particularly language acquisition. Current accounts are generally phrased in terms of rule-governed behaviour (Segal & Stacey, 1975) or information processing (Pollio, 1974). The general theory shows that these accounts can be reduced, if necessary, to operant conditioning; it provides for a unification between behaviourist psychology and psycholinguistics.

The general theory also suggests alternative approaches to the investigation of the acquisition of language, in that its roots in learning theory posits an essential continuity between all types of learning. I have taken this continuity as a (partial) sanction of the study of language learning in adults as being relevant to the child's first language learning. I have also taken the learning theory tradition in the use of verbal materials and have employed nonsense syllables in an experimental laboratory investigation. Finally, I have accepted the view of information feedback as a form of reinforcement (Estes, 1972), and have examined the effects of different forms of feedback upon the acquisition of a miniature artificial language.

Processes of language acquisition

Glucksberg, Krauss, and Higgins (1975) quote a personal communication from George Miller that "We have but two theories of language acquisition at the moment. . . . the miracle theory or the impossible theory" (p. 338). In what is an unusual situation for Psychology, Miller might well have added that we also have no data, as the quotes in the chapter on the aims of this study suggest.

In the two chapters after the "Aims", this review of studies of how language is learned has been prefaced by a discussion of the current conception of what the child has learned, and a discussion of a theory which is powerful enough to account for this learning. A broad and quite detailed stage has been set, but the play is discovered to be a short one-act. The reasons for this have been alluded to earlier; in summary, research up to 1970 was concerned with the grammatical structure of the child's first utterances, while research after 1970

has broken free with studies of semantic structure, language functions, the extra-linguistic context, and processes of comprehension and production. There has presumably been insufficient time in which these recent findings could begin to pose questions about how they might occur, and as Miller observes any such questions as might have been put seemed answerable only in terms of miracles or impossibilities, Finally, there is an important methodological issue: the dominant data collection strategy involved the tape-recorder in the acquisition of a corpus of utterances for purposes of analysis. Such a strategy is an essentially observational one where, at best, correlations between variables could be described. While such correlations provide provocative suggestions and hypotheses about underlying causes in acquisition, they cannot serve as very strong evidence for such causes, and neither can such hypotheses about causes be satisfactorily tested by the observations. This point is developed further in the introductory chapter to the experiment.

The acquisition of syntax

The "short one-act" which deals with processes underlying language acquisition mainly consists of Brown and Bellugi's (1964) suggestions and the research which they have generated. Brown and Bellugi (1964) outlined three processes in the child's acquisition of syntax: imitation and reduction, imitation and expansion, and induction of latent structure.

The process of imitation has long been a favourite of social psychologists seeking to account for the development of social behaviour (e.g. Bandura and Walters, 1963), and the occurence of linguistic imitation has been widely observed (e.g. McNeill, 1970) though there

is evidence for considerable variation between children in the extent to which it occurs (Bloom, Lightbrown, & Hood, 1975). Brown and Bellugi (1964) considered that the imitation of parental speech by the child could lead the child to assimilate and then produce more advanced syntactic features. Such a prediction was checked by Ervin (1964), who found that a grammar written on a corpus of spontaneous speech was quite consistent with the same child's imitations; had the imitations been in any way advanced, the grammar would presumably not have fitted the data. McNeill (1970) simply dismisses imitation as a plausible process in language acquisition with his well-known observation:

Child: Nobody don't like me

Mother: No, say "nobody likes me"

Child: Nobody don't like me

(eight repetitions of this dialogue)

Mother: No, now listen carefully:

say, "nobody likes me"

Child: Oh! Nobody don't likes me (p. 107)

Whitehurst and his colleagues (Whitehurst & Vasta, 1975) have defended imitation, emphasising the occurrence of novel content within an imitated grammatical structure. They propose a notion of selective imitation rather than the quite restrictive definition of imitation which has tended to trivialise what has been subsumed under the term. They view the role of selective imitation as the mediation between initial comprehension of a sentence employing a novel structure (the receptive abilities of the child) and the spontaneous production of a sentence which incorporates the structure.

Imitation and expansion, called simply "expansion", is a form of

parent-child interaction:

> If the child said Eve lunch or Throw daddy, the parent often responded with the nearest complete sentence appropriate in the particular situation--Eve is having lunch or Throw it to Daddy (Brown, Cazden, & Bellugi, 1969, p. 139 in Brown, 1970)

Brown argues that such expansions perform an important function for the child, providing him with data from which he may build his grammatical knowledge. Brown also notes that the expansion of the parent is a model; that is, a well-formed utterance which the child may use independently of its informational content. An alternative possibility for the growth of the child's syntactic competence is simply that the child may abstract grammatical regularities from appropriate models heard in everyday life. Since model sentences and expansions occur uncontrollably in the natural environment, advances in the child's knowledge cannot be reliably ascribed to either one or the other. Cazden (1965, summarised in Brown et al, 1969) carried out an experiment

> which was designed to separate these two aspects of the child's language environment. . . . The expansion group (of children) received forty minutes per day of intensive and deliberate expansions. The modelling group received exposure to an equal number of well-formed sentences that were not expansions. (p. 140 in Brown, 1970)

The results contradicted the expectation of the experimenter--modelling was found to be more effective than expansions in improving various indices of linguistic competence. This rather counter-intuitive result provoked the experiment reported in this study. Cazden (1972) has remained interested in this finding, and notes the more general paradox the parents do not generally correct grammatical errors in the child's

utterance, yet the child still acquires syntax.

The third process, the induction of latent structure, is an account of the use the child might make of model sentences. The child is viewed as a tester of hypotheses and developer of rules, where the input provides potentially disconfirming evidence. This position owes much to Chomsky's (1965) discussion of the LAD, but also shares its pitfalls. As Fodor, Bever, and Garrett (1974) conclude, the formal properties of the LAD are not sufficient to guarantee the acquisition of the full grammatical complexity of the child's first language. The point has also been made by Braine (1971), who shows that the LAD, as specified, would in fact acquire very little. The notion of the child as an active processor of input information is a congenial one, however, and Slobin has made some efforts (1973) to specify "operating principles" which the child might employ as heuristic strategies in the acquisition of syntax. Examples of such principles are phrased as instructions which the child carries out: Pay attention to the ends of words; pay attention to the order of words; avoid interruption or rearrangement of linguistic units. Whatever difficulties there may be in accounting for the process of syntax acquisition in terms of such strategies, there is no doubt that the child uses rules in forming his utterances. This was demonstrated in the seminal study of Berko (1958), where the child was told of some object "This is a wug; there are two ____" and the child would obligingly say "Wugs".

However, it has become increasingly clear that the acquisition of syntax (let alone its use in comprehension and production) is intimately tied to the acquisition of semantics and linguistic functions, and that the question of real interest, meaning, regards syntax as simply one way to convey (communicate) meaning. Such a view is put by Bloom (1975),

and Glucksberg et al (1975) state:

> It is our view that so long as meaning, communication, and intentions to communicate are excluded from consideration as central to the interactive acquisition process, efforts to deal with acquisition of syntax per se or lexicons per se are doomed to failure. (p. 339)

MacNamara (1972) has gone so far as to say that

> infants learn their language by first determining, independent of language, the meaning which a speaker intends to convey to them, and by then working out the relationship between the meaning and the language. (p. 1)

The acquisition of semantics

The approach of procedural semantics, discussed in the chapter on what is acquired, has certain implications for a theory of language acquisition which are entirely consistent with the general theory of behaviour developed earlier. If a sentence is considered to be understood through some procedure of "compile and execute", then the question naturally arises of how this procedure is acquired. The approach is to view the infant as a system which encodes the regularities in its environment in terms of action schema; these action schema are procedures for appropriate action and activity, where objects and relationships in the environment are internalised in terms of the ways that they may be interacted with. This very brief statement is hardly adequate to the complex concepts involved; but there are quite obvious parallels in Piaget's account of the cognitive growth of the child (Piaget & Inhelder, 1966). These notions are also present in the behavioural theory of Taylor (1962) and the monograph of Arbib (1972), which provided the necessary

level of discourse for level five of the general theory of behaviour.

There are, of course, other and less esoteric accounts of language acquisition which draw upon the Piagetian approach, and particularly the view that cognitive development is the basis for language development. Sinclair (1971) states the Piagetian (Genevan) position quite explicitly: the emergence of the semiotic function in behaviour is a necessary pre-condition to the emergence of language where reality can be represented symbolically. As she states a little later on, however, (Sinclair-de Zwart, 1973) "to take Piaget as a handbook for developmental psycho-linguistics is not a simple task" (p. 25). The genetic epistemological view by itself is too broad an account of general cognitive (logico-mathematical) development to serve adequately as a detailed account of the acquisition of language.

Finally, the functional views of Halliday (1975) must be introduced. The process underlying language acquisition in this theory is the progressive differentiation of language functions that the child acquires. Put behaviourally, the child is reinforced in his attempts to convey meanings to the listener, and that his growing cognitive and social skills lead to a wide variety of "message conveying techniques". Put more cognitively, and phrased in terms of the communication of meaning, Glucksberg et al (1975) suggest

> one source of pressure upon a child to differentiate his surface utterances in order to differentiate explicitly his propositions (meanings) is the need to be understood. (p. 339)

The current emphasis on the role of meaning in the understanding of utterances, both by the child and the adult, introduces the question of how "meaning" develops. The section so far has dealt with the functional and process accounts advanced by Halliday and Piaget, and it

is important for the present study to examine the account given by research into conceptual structures. The most relevant area is presumed to be that of "concept acquisition", with studies of semantic memory and the perceptual structuring of the environment running a close second.

The extensive literature on the experimental study of concept acquisition, however, does not have much to contribute towards the explication of the acquisition of the natural concepts (Neimark & Santa, 1975). The typical experimental paradigm involves a set of stimuli which vary on a number of dimensions; the experimenter has in mind a rule (concept) of the form "all red squares", and the subject is required to identify this rule by sorting the stimuli into exemplars and non-exemplars and being informed about the correctness of such sorting. There are various mathematical models for the performance of subjects under such conditions (originating mainly from the work of Trabasso & Bower, 1968) which are interpreted in terms of the formation, testing, and rejection of hypotheses. The terminology is the same as that employed within the general theory of behaviour, and as such the studies of concept identification are congenial to the present view of developmental psycholinguistics; but the substantive content is usually not. The introduction of at least two further elements into these studies seems necessary: the subject should be required to acquire multiple concepts simultaneously, rather than just one; and these concepts should be symbolically labelled, preferably with words. Such a step would presumably be in the right direction for the purposes of understanding language acquisition.

Studies of semantic memory and conceptual structures hold out more promise for such a project, if only because such memory has been implicated in the discussions of what the child has acquired in terms of semantic structure, language function, and processes of comprehension and production.

The most comprehensive of such studies has certainly been those of Anderson (Anderson & Bower, 1973; Anderson, 1976) which bear considerable resemblances to the theory of Schank (1972). The essential features of Anderson's later formulation (called ACT; the earlier theory was HAM) are by now quite familiar: parallelism, or multiple simultaneous activity in the brain; procedural rather than declarative knowledge; and a formalism involving directed graphs, which is isomorphic with either TOTE or finite automata formalisms. Although Anderson does not directly address the question of how the semantic memory structures may be acquired by the child, it may be assumed that the account he would give would be very similar to the enactive, behaviourally-based views discussed in the section on the acquisition of semantics.

An integration

Katherine Nelson (1974) has provided an integration of Piagetian theory, concept formation studies, and semantic feature theory, which aims to

> provide a description of the child's formation of concepts, learning of words that fit those concepts, and construction of concept-relating statements. (p. 268)

To put her position more accurately, the experimental studies of concept identification are dealt with as being irrelevant; she employs notions of concept attainment drawn from Cassirer (1953, which I have not been able to view) in preference. In any event, her model is familiar:

> As applied to the language-learning child, the model emphasises that an object is first identified as having important functional relationships; that these relate the object to self and other people through a set of acts; and that perceptual analysis is

> derivative of the functional concept, not a priori essential to it. (p. 284)

There is a remarkably broad agreement on the kinds of processes which underlie language acquisition, at least as it has been presented here. The agreement is exactly as defined by Nelson (1974) in the quote above. What is missing is evidence for such a view; the next chapter will attempt to justify a methodology which follows from these views and which will provide the needed data.

Introduction to the experiment

Any research is the result of a confluence of a number of different streams of interest and relevance; the present experiment is no exception, The experiment serves three functions. The first, and most important function, is to illustrate a methodology; that is, it suggests the kinds of studies which might be fruitfully carried out by students of the acquisition of language. The second function is the attempt to adduce evidence for the role of feedback in language acquisition, while the third function is again illustrative, this time of the computer techniques involved in controlling an on-line real-time experiment.

Feedback in language acquisition

Perhaps the strongest argument in favour of the nativist position on language acquisition (Fodor, Bever, & Garrett, 1974) is the evidence, or lack of it, for any influence of reinforcement or feedback upon the

child's language learning. The general theory of behaviour suggests that language learning can be accounted for by utilising the concepts of operant learning theory; such a crucial concept is reinforcement, and it is presumably evidence against the general theory should reinforcement prove to be ineffective. This conclusion is not warranted, however, because the relevant levels of discourse have not been properly separated. The general theory seeks to account for language learning at level four at least while the operant concepts are located at level one. As Polanyi (1968) shows so clearly, organisation of the processes at a certain level are irreducible to lower levels even while the content of these processes may be explained by the lower level.

The notion of reinforcement (at level one) has been sufficiently attractive, however, to encourage its translation and use at level three. The cognitive view of reinforcement is called feedback (Estes, 1972) or knowledge of results (Annett, 1969), but is not equivalent to reinforcement. There are features of feedback in the general information processing model which cannot be accounted for by the operant notion of reinforcement.

Feedback or knowledge of results, however, is not necessarily explanatory at level four, belonging as it does at level three. That is, feedback is an essential part of the TOTE formalism which has been located at level three; to the extent that plans are TOTE hierarchies and that such hierarchies may themselves be treated as single TOTE units, there is the suggestion that feedback will also play a part in the use of plans, which are located at level four. But this is no more than a suggestion, a hypothesis, since different principles of organisation at level four are probably operating. This latter statement is itself a hypothesis, since very little is known about the processes by which

plans are assembled and used; it is nonetheless rendered quite likely, given that plans have been located at a level above single TOTEs.

The outcome of this discussion is that the notion of operant reinforcement is not applicable to the issues of language acquisition; that notions of feedback and knowledge of results are hypothesised to be applicable; and that there may be other forms of organisation which operate in language learning which may not be reducible to feedback. There have been no suggestions as to what these other forms of organisation could be; and so the present study will examine the hypothesis that feedback is implicated in language acquisition.

The evidence for the role of feedback has been discussed in the previous chapter where the issue was the effectiveness of expansions in the acquisition of syntax. The most important experiment was conducted by Cazden (1965), with negative results. This outcome was very puzzling, and some possible reasons for it have been advanced, both by Cazden and his critics. Brown, Cazden, and Bellugi (1969) note three such criticisms of the experiment. The first is that the models given to the children could have provided a richer and wider linguistic data base than the expansions, since these necessarily follow on from the child's initial utterance. The second factor is that the lack of clear external references of the child's initial utterance could have resulted in it being misinterpreted by the experimenter, so that the resulting expansion might in fact confuse and hinder the child. The third factor is that "artificial elevation of the expansion rate may depress attentional processes in the child" (Brown et al, 1969), making the models more interesting than the expansions. Other criticisms are possible, of which I want to make two. Firstly, the study was fundamentally concerned with the effects of models and expansions upon the growth of grammatical

competence, and it is still an open question whether expansions improve semantic or functional competence. Secondly, there are difficulties with the introduction of an experimental manipulation into the natural situation. That is, the child in the experiment is open to a wide variety of other models and expansions from teachers, parents, and other adults, under conditions where unknown variables may be operating. This point does not bear directly upon the question of why modelling was more effective, but bears upon the issue that there may have been (unknown) variables and influences which may have reduced the child's ability to profit from expansions.

Form of the proposed study

The study of the role of feedback in language acquisition must be constrained by the criticisms made of Cazden's experiment. The most important question left unanswered by Cazden's investigation is the role of feedback in the development of semantic competence, while the findings with respect to syntactic competence are open to doubt. In order to study the role of feedback upon the development of syntactic and semantic competence, it is necessary to provide feedback where semantic and syntactic information is clearly separated. Failure to separate these forms of information would make the interpretation of any changes in competence quite difficult.

Cazden notes (quoted in the section on the acquisition of syntax, previous chapter) that an experimental study was dictated by the necessity of separating models from expansions; in a similar fashion, a laboratory experiment is dictated here by the necessity of separating syntactic from semantic information. This is because any natural utterance contains (and conveys) both semantic and syntactic information inseparably.

Any attempt to control for the criticisms made of Cazden's study in a natural or semi-natural environment, employing a natural language, would be extraordinarily difficult. In particular, the very richness and diversity of natural languages would make it almost impossible to equate models and expansions for their degree of richness, and the influences of unknown variables associated with the natural linguistic environment would be unknown.

These considerations dictate a manipulative, laboratory experiment which has the following features:

(1) Subjects are required to learn a novel, simple, miniature, artificial language, which can be mastered within a single session. The advantage here is that the past linguistic history of the subject is not directly relevant; and the whole course of learning can be encompassed within a reasonably short period of time.

(2) The errors in sentences produced by the subjects are corrected by forms of feedback where syntactic and semantic information are separated. This is the most important advantage of an artificial language. The two forms of feedback, called semantic feedback and syntactic feedback, define four possible experimental conditions: (i) no feedback; (ii) syntactic feedback only; (iii) semantic feedback only; and (iv) both syntactic and semantic feedback.

(3) The subjects are presented with model sentences, which they are required to copy. This is necessary to allow the learning of the language in the no feedback condition; it allows an assessment of the ability of the subject to "induce the latent structure" of the language, a process discussed earlier (Brown & Bellugi, 1964).

(4) The language is learned in the context of a "world" (Winograd, 1972) of two-dimensional geometrical objects. This provides the necessary

referents for the language and the vocabulary, since it has been shown that subjects will learn nothing about a language where they are simply exposed to well-formed strings with no referents (Moeser & Bregman, 1973) (5) The experiment is under the control of a computer. This is a necessary, and not a frivolous, requirement, since a human experimenter would require many hours of practice to be able to provide exactly and only the feedback demanded by a particular experimental condition; and even with such extensive practice, errors would inevitably occur along with variable delays in providing feedback. The computer, on the other hand, is capable of storing the target grammar and vocabulary, of providing error-free feedback almost instantly, and in addition can maintain a trial-by-trial record of relevant performance data.

Illustration of methodology

The focus of this whole study is upon the processes involved in language acquisition. The particular experiment discussed above represents an investigation into one aspect of these processes: the role that feedback plays in language learning. It is suggested that the methodology employed in this investigation could serve as a model for other investigations into language acquisition.

The elements of such a model would be the use of a miniature artificial language, and a computer to control the experimental environment. All laboratory investigations of natural phenomena involve a degree of idealisation of the phenomena, to obtain a clearer view of the basic elements in the phenomena. I would suggest that the phenomena of language acquisition should be brought into the laboratory as a start to its investigation; this will allow the isolation and detailed study of its elements. I would further suggest that the investigation should

concern itself with artificial languages, since these will have the advantages outlined in the previous section, particularly that of freedom from contamination from natural languages. It is necessary, however, that the idealisation of the phenomena does not reduce it to triviality (in the sense instantiated by most of the studies of "concept acquisition") and it is for this reason that I argue for the use of the computer in controlling the experimental studies. The use of the computer will allow sufficiently complex linguistic material and experimental procedures such that the "essence" of the phenomena will not be lost.

It is fairly clear that this methodology will allow the investigation of "the child's formation of concepts, learning or words that fit those concepts, and construction of concept-relating statements" (Nelson, 1974, p. 268; also quoted in the section on an integration, previous chapter), but so do other methodologies. What other methodologies do not do is "emphasise that an object is first identified as having functional relationships(and) that these relate the object to self . . . through a set of acts" (Nelson, 1974, p. 284; also quoted in the section on an integration, previous chapter). In this sense, the proposed methodology is consistent with the metatheory of the general theory of behaviour; the essential aspect which makes this so is the capability of the methodology to represent "alternate realities" to the subject, laboratory environments where the functional relationships between objects can be manipulated and where these relations to the subject can also be manipulated. These alternative "worlds" are strictly necessary in order to investigate the variables which the general theory identifies as being of fundamental importance. A comparison of the outcome of language learning in these various "worlds" will allow the identification of relevant variables

which are then presumed to be operating in the "real" world.

These observations are not particularly original, of course; many experiments which are carried out deal with "alternative worlds" in the sense that they deal with idealisations of the "real world", and many experimenters utilise the strategy of constructing artificial environments in order to study interesting variables. Such strategies were the preferred experimental method of Taylor (1962), upon whose work the general theory of behaviour partially rests. What I have done is to simply explicitly identify two techniques (artificial languages and computer-controlled experimental environments) which should serve the needs of students of language acquisition, to the extent that these techniques allow the experimental manipulation of the basic relationships invoked in theories of language learning.

The reason for the argument about methodology, which has been presented here, may be somewhat unclear. It is due to the fact that the experiment was run using university students as subjects; and that this feature may have obscured the relevance of the methodology to theories of language acquisition in the child. This use of students as subjects is addressed in the discussion of the experimental findings. In any event, it is useful to be explicit about the theoretical and metatheoretical overtones implicit in any methodology.

Other studies using artificial languages

The use of artificial words may be dated from Berko's (1958) famous study: "This is a wug; there are two ____". The technique has since become a part of the methodology of diary and longitudinal studies, when the investigator wants to know if the child has mastered a certain grammatical form (e.g. Braine, 1963; Brown & Bellugi, 1964).

The use of artificial languages which were content-less may be dated from Braine's (1965a, 1965b, 1966) studies. These studies utilised nonsense syllables for words, and grammatical productions such as S -> aXb and S -> pXq, but where the words did not have any meaning, that is, they did not refer to anything.

The use of artificial languages where the words have meaning in the context of some set of referents is associated with the work of Moeser and her colleagues (Moesser & Bregman, 1972; Moeser & Bregman, 1973; Moeser & Olson, 1974; Moeser, 1975). The emphasis of her work, however, has been upon the acquisition of syntax by the subject, and the thrust of her findings is that the learning of syntax depends upon the subject's understanding of the meaning of the sentence in relation to the referents. These results provide support for the arguments of MacNamara (1972) and Nelson (1974) for the importance of semantics in language acquisition. While Braine used children in his studies, Moeser has used university students, with the exception of Moeser and Olson (1974), which used children in a replication of Moeser and Bregman (1972). Essentially similar results were obtained.

Method

The miniature artificial language which the subjects were required to learn consisted of a simple three-place grammar:

S	->	Adj	+	N	+	Num
Adj	->	BUF, JIK				Shaded, Blank
N	->	TUG, ZAB				Square, Triangle
Num	->	LAT, WIX				One, Two

The vocabulary consisted of six CVC nonsense syllables, chosen randomly from a list with moderately high association values (Archer, 1960). As may be seen from the English meaning of the words, the language is capable of describing pictures which contain one or two triangles or squares which may be either shaded or blank. There are thus eight possible pictures which may be described.

During the experimental session, subjects were seated in front of a visual display unit (VDU) and keyboard which was connected to the computer, and their communication for the entire experiment was with the computer via the VDU and keyboard. The first six keys on the keyboard were each labelled at random with one of the

words in the vocabulary.

The unit of the experimental procedure was the trial. On every trial, one of the eight possible pictures was displayed on the VDU. For a modelling trial, a correct, well-formed description of the picture, in the language, was provided below it, and the subject was required to copy the model sentence by pressing the correct keyboard keys in their correct order. For a feedback trial, the subject was required to provide a description of the picture as best as he could, using the words provided on the keyboard. This description was then analysed by the computer, and the subject was given feedback information about his attempt according to the experimental condition.

The modelling and feedback trials were provided alternately. Eight trials, four modelling and four feedback, made up a block of trials. Within any block, each of the eight possible pictures occurred once, at random; and for any two adjacent blocks, a given picture was the subject of a modelling trial once, and a feedback trial once. The experiment was terminated after two consecutive error-free blocks. Exactly which picture followed which from one trial to the next defined a measure of connectivity; for example, if a following picture had two out of three elements in common (such as two shaded squares following one shaded square) then the connectivity between the trials was of degree two. This measure was an important parameter of the ease with which the language could be learned; for the present experiment, the average connectivity was of degree 3/4.

The computer maintained a record of errors and response latencies for each block of trials. This record was punched on

paper tape for later examination; an example of a record is provided in the appendix. Three kinds of error were possible in a feedback trial; order errors occurred when the subject misplaced a particular word; for example, by placing the noun last instead of second. Class errors occurred when a grammatical class was missing from a sentence; for example, the class of adjective would be missing from the sentence "TUG ZAB LAT" (square triangle one). Word errors occurred when an inappropriate word was used in the description of the picture; for example, by calling a picture of a square "ZAB" (triangle). Any errors made in copying a model during a modelling trial were also noted. The response latency of each block was the average of the four response latencies in the feedback trials. A response latency was taken from the onset of the request to provide a description to the instant that the third word of the sentence was given.

Equipment

A Hewlett Packard 2114B mini-computer controlled the experiment. The two devices connected to it which were relevant to the experiment were the visual display unit (VDU), and a keyboard. The VDU was an ADDS consul 520 operating as a direct teletypewriter replacement, at a baud rate of 2400. The ADDS keyboard was covered and was not used; a special keyboard was constructed for the use of the subject. This keyboard had provision for 63 push-button switches, where each button or key could be conveniently labelled. Internal electronics generated a six-bit code for each key, with a latch to prevent this code being disrupted if another key was pressed simultaneously. The six-bit code closed or opened six relays, whose state could be

sensed by the computer. Further details of the computer hardware may be found in the section dealing with the computer control of experiments.

Two adjacent rooms were used; the one contained the VDU and keyboard on a table, while the second contained the computer. Sound-proofing was good, such that a person seated at the VDU could not hear any noise from the computer room under normal circumstances.

Subjects

Thirty volunteers from a university population participated in the experiment, for a fee of R2,00. Subjects were both males and females, between the ages of 18 and 24, who were neither senior psychology students nor language students. Each subject was allocated at random to an experimental treatment upon arrival. The protocols of some six subjects were discarded due to either computer failure or inability to learn the language. This latter outcome was not related to the experimental conditions.

Procedure

Subjects were seated at the table with the keyboard and VDU. The first page of instructions was already present on the VDU's screen; when this had been read and understood, the subject pressed a button, and the next page was presented, where the same procedure was adopted. After the final page, the experimenter answered any queries, and the first trial was then presented, which was always a modelling trial. The subjects were alone in the room until the computer halted the experiment, when the experimenter conducted a

post-test and de-briefing.

On any trial, the subject was allowed ten seconds in which to either copy the model provided, or to produce a description. A buzzer began sounding after ten seconds, and only stopped when a sentence of three words had been typed in. In a modelling trial, one second then elapsed before the next trial; while during a feedback trial, the display and feedback was available to the subject for four seconds before the next trial was started.

Any error made in copying a sentence on a modelling trial elicited a request from the computer to try again, while any attempt to use the same word twice in any sentence yielded the message "Please do not use a word more than once in a sentence; sorry--there was an error in your input, please try again."

Instructions to subjects

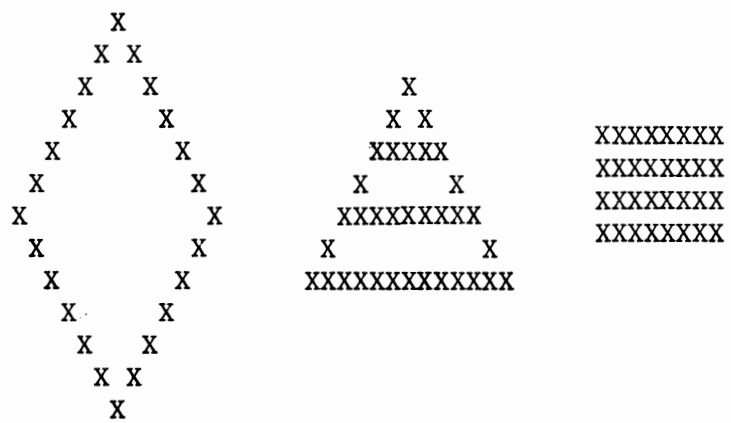
These were presented in 12 "pages" to each subject on the VDU. Pages 10 and 11 were different, depending upon the particular experimental condition. These are specified below:

(1) In this experiment, you will learn a simple language. This language, or code, bears no relationship to English or to any other language.

The general procedure is that the computer will display simple geometric figures on this screen, and you will learn how to describe the pictures in the language.

The computer will give you some help in learning this language; it will also be keeping a record of your right and wrong answers, so that we can study how people learn a language.

(2) This new language, or code, is used to describe simple geometric pictures; here are some examples of the figures that you will see:



(3) The words in the language are all simple syllables like 'JIK', 'BUF', or 'ZAK'. If you look at the keyboard in front of you, you will see all the words which are used in the language.

A sentence in the language always consists of exactly three words; for example, 'JIK BUF ZAK'.

During the experiment, you will need to make up possible sentences. You should use the keyboard to do this; just press the key for a word when you want to 'say' that word. Check this screen to see that the word was printed correctly.

(4) Now for some details about the experiment.

The general procedure will be that the computer will draw a picture on the screen; then, sometimes the computer will tell you what the code for the picture is, while sometimes the computer will ask you to give the code for the picture.

When the computer tells you the code, you will be asked to copy it. When you give a code, i.e. a sentence which describes

the picture, the computer will check it and tell you if your code or description was right.

(5) When the computer gives you the code, it will say for example:

'Please copy this description: JIK BUF ZAK'.

Now the sentence 'JIK BUF ZAK' will be a good and accurate description of the picture, so you should study it carefully.

You must then copy the sentence, using the keyboard. Do not add to, or change, the original; just type in what the computer said.

(6) When the computer asks you for the code or correct description of the picture, it will say:

'Please describe the picture'

and you must type in what you think the correct description should be.

After you have described the picture, the computer will check your sentence. If you made no errors in your description, then the computer will say:

'Good description, completely correct.'

When you begin, you will probably be at a loss what to say when asked to describe a picture. Don't worry about this; say something (anything) and concentrate on working out the language or code.

(7) There is a set time limit for you to respond to the computer's request to either copy a sentence, or to type in your own description.

If you exceed this time limit, the computer will sound a

beeper, and it will only be turned off when you finish typing in a sentence.

If you hear this beeper, do not worry, but simply type in your reply straight away, without thinking about it any more. This is very important--you must type in your reply immediately when you hear the beeper.

(8) Now for some remarks about the language you will learn.

There are two features of any language (including the simple language we are using) which you should keep in mind.

The first is the feature of word usage, or word meaning. This refers to the fact that each word in the language means something, and means something different from each of the other words in the language.

The second is the feature of word order. This refers to the fact that the words in a sentence must be in a certain order for that sentence to make sense.

(9) For example, you might see a boy hitting a dog; the correct way to describe this in English is to say 'The boy hit the dog.'

Now if you said 'Hit the boy the dog', this would be incorrect, because your word order is wrong, even though you have used the correct words.

If you said 'The bat hit the ball', then this would also be incorrect, because you have used the wrong words, even though your word order is quite right.

When you learn the language, you will really be learning these two things: what each word means, and what is the right order of the words in a sentence.

Pages 10 and 11 are specific to the experimental condition. For condition 1, the pages are as follows:

(10.1) The computer will give you some help in learning the language. After you have typed in your description of a picture, it will look at your description or code, and tell you if it was completely correct or not.

If you used the words with the correct meaning, in the right order, then the computer will say:

'Good description, completely correct'.

If your word order was incorrect, or if you used the wrong words, or both, the computer will say nothing.

(11.1) Here is an example. Suppose you were learning English, and you were given the following picture:

```
XXXXX        XXXXX     (The correct description might
X   X        X   X
X   X        X   X      be 'Two small squares.')
XXXXX        XXXXX
```

Now if you said 'Two small squares', then the computer would say 'Good description, completely correct'.

On the other hand, if you said 'small squares two', or 'one big triangle', or any other incorrect sentence, then the computer would say nothing.

For condition 2, syntactic feedback only, the instructions are as follows:

(10.2) The computer will give you some help in learning the word order in the language. After you have typed in your description

of a picture, it will look at the order of the words you have used, and will tell you if your word order was correct.

If you used the words in the right order, but had incorrect word usage, then the computer will say:

'Good word order, but not completely correct'.

If your word order was incorrect, then the computer will rearrange the words that you used so that they are in the order required by the language, and will say:

'Word order should be: AAA BBB CCC'

Sometimes a word or two will be replaced by 'XXX', which indicates that the computer could not re-order your words completely.

(11.2) Here is an example. Suppose you were learning English, and you were given the following picture:

```
XXXXX        XXXXX
X   X        X   X     (The correct description might be
X   X        X   X
XXXXX        XXXXX      'Two small squares')
```

Now if you said 'One big triangle', the computer will say 'Good word order, but not completely correct'. Your word order is excellent English, but your word usage is quite wrong.

On the other hand, if you said 'Big triangle one', then the computer will correct your word order, using your own words, and will say 'Word order should be: One big triangle'. What happens here is that the computer takes whatever words you used, and puts them in the right order.

For condition 3, semantic feedback only, the instructions are as follows:

(10.3) The computer will give you some help in learning the word usage, or meaning, in the language. After you have typed in your description of a picture, it will look at the meaning of the words you have used, and will tell you if your word usage was correct.

If you used the words with the right meaning, but had incorrect word order, then the computer will say:

'Good word usage, but not completely correct'

If your word usage was incorrect, then the computer will substitute the wrong words that you used with the correct words, keeping your own word order, and will say:

'Word usage should be: AAA BBB CCC'

Sometimes a word or two will be replaced by 'XXX' which indicates that the computer could not tell you what all the correct words should be.

(11.3) Here is an example. Suppose you were learning English, and you were given the following picture:

```
XXXXX       XXXXX
X   X       X   X       (The correct description might be
X   X       X   X
XXXXX       XXXXX       'two small squares')
```

Now if you said 'Small squares two', the computer would say 'Good word usage, but not completely correct'. Your word usage is correct English, but your word order is quite wrong.

On the other hand, if you said 'Big triangle one', then the computer will correct your word usage by substituting correct words, and keeping your own word order, will say 'Word usage should be: Small squares two'. What happens here is that the computer takes your words and corrects them, whatever the

word order that you used.

Finally, for condition 4, both syntactic and semantic feedback, the instructions are as follows:

(10.4) The computer will give you some help in learning the word order and word usage in the language. After you have typed in your description of a picture, it will look at the meaning and order of the words you have used, and will tell you if your word order and usage was correct.

If you used the words with the correct meaning in the right order, then the computer will say:

'Good description, completely correct'

If your word order or word usage was incorrect, then the computer will tell you the correct way to describe the picture, and say:

'Correct description should be: AAA BBB CCC'

(11.4) Here is an example. Suppose you were learning English, and you were given the following picture:

```
XXXXX        XXXXX
X   X        X   X        (The correct description might be
X   X        X   X
XXXXX        XXXXX         'Two small squares')
```

Now if you said 'Two small squares', then the computer would say 'Good description, completely correct'.

On the other hand, if you said 'Small squares two', or 'one big triangle', or any other incorrect sentence, then the computer would say 'Correct description should be: Two small squares'. What happens here is that the computer corrects any error of word order or word usage.

The last page is common, as for pages 1 to 9, for all the

conditions.

(12) Finally, let us recap briefly. The task you must solve is to learn a language; or, in other words, to 'break a code'. Remember that you must learn what the words mean, and what their correct order is in a sentence.

The computer will draw pictures; sometimes it will tell you the correct code for the picture which you will then copy, while sometimes you will have to describe the picture and the computer will check your description. There is a time limit for you to study the information you will be given; if you hear a beeper, you must immediately respond to the computer's request.

The computer will automatically end the experiment when you have mastered the language completely.

Thank you for your help.

Please ask the experimenter about any matters which may still puzzle you.

Press a key only when you are ready to begin.

The post-test

Immediately after the computer terminated the experiment, the experimenter asked each subject to complete a short questionnaire; this is reproduced in the appendix. There were two sections; in the first, the subject was asked to indicate the English meaning of each of the six nonsense syllables, while in the second section ten sentences in the artificial language were given, of which three were grammatically correct, and the subject was asked to place a tick or cross against each to indicate whether it was grammatically correct.

The subject was then paid, and any questions were answered in a general way, without any details of the forms of feedback being given.

Results

On a given feedback trial, four measures of performance were taken for each subject: the number of class errors (either 0 or 1), the number of order errors (either 0, 1, or 2), the number of word meaning errors (0, 1, 2, or 3), and the response latency (in hundredths of a second). The response latency was taken from the onset of the request to describe the picture, to the input of the third word of the description. On a modelling trial, the number of errors made in copying the model were also noted.

These performance measures were accumulated for a block of eight trials (four feedback trials) and then punched to paper tape. The response latency was averaged before being recorded. The paper tape was later printed, and the total number of errors made in each category over the whole experiment was calculated. The average response latencies were summed and multiplied by 4 to yield a measure of the amount of time taken by the subject to respond to requests to describe a picture, while the number of blocks of trials taken to learn the language was calculated as the block number on which the last error was made; two blocks later the experiment was terminated.

For the data analyses, two-way analyses of variance were usually computed. Factor A represented the presence or absence of semantic feedback (A1--no semantic feedback; A2--semantic feedback), while factor B represented the presence or absence of syntactic feedback (B1--no syntactic feedback; B2--syntactic feedback).

The 5% level of significance was adopted for all analyses. The computational procedures were as outlined by Kirk (1968), Winer (1971), and Roscoe (1975). The probability values were indicated on the tables in the following way: +p=,05, ++p=,01, and +++p=,001.

An example of the raw data is provided in the appendix. Also, the summary data for each subject is provided in the appendix.

Measures of time and trials

The first variable of interest was the number of blocks of trials to criterion. The anova summary table is given in Table 1a, with the summary statistics in Table 1b; the graph of the cell mean profiles is given in Figure 1. There were no significant effects, indicating that there were no reliable differences between the feedback conditions in terms of the number of blocks of trials taken to learn the language. An examination of the graph of cell mean profiles, however, suggests the presence of an interaction effect which has presumably been masked by particularly high random sampling variation. Further data which is relevant to this insignificant interaction is presented in a later section of the results on measures of time and trials to partial completion.

The second variable examined was the total response latency in all feedback trials. This measure was taken to quantify the amount

of time, in total, that the subject needed to master the language; that is, no attempt was made to record the response latency of the subject on any of the modelling trials, in the expectation that such time as was taken would not be directly relevant to the question of "How much time did the subject actively need?" This was a mistaken presumption, however; as will be mentioned in the discussion, the modelling trials provided important information to the subject in learning the language, and hence measures of time used during the modelling trials would have been necessary.

The anova summary table of total response latency is given in Table 2a, with summary statistics in Table 2b. There were no significant effects, indicating that the various feedback conditions did not reliably influence the total time taken by the subjects to respond to requests to describe a picture.

A related question is that of the average response latency in a feedback trial; this variable was analysed, and the anova summary table is given in Table 3a, with the summary statistics in Table 3b, and the graph of the cell mean profiles is given in Figure 2. The interaction effect was significant, indicating that the average response latency depended upon the particular feedback condition. Tukey HSD pairwise comparisons were calculated between the cell means, and these are reported in Table 3c. Significant differences were shown between condition 3 (semantic feedback) and condition 4 (syntactic and semantic feedback), where the average response latency was lower in condition 3; and between condition 3 and condition 1 (no feedback), where again the average response latency was lower in condition 3.

Table 1a

Analysis of variance summary table for number of blocks of trials to criterion under conditions of semantic and syntactic feedback.

Source	SS	df	MS	F ratio
A (semantic)	8,21	1	8,21	0,12
B (syntactic)	37,5	1	37,5	0,54
AB	60,29	1	60,29	0,86
Within	1395,94	20	69,80	

Table 1b

Cell and treatment means (and cell standard deviations) of the number of blocks of trials to criterion.

		Factor B (syntactic feedback) B1 (absent)	B2 (present)	
Factor A (semantic feedback)	A1 (absent)	15,83 (6,05	16,50 (9,2)	16,17
	A2 (present)	20,17 (8,42)	14,5 (9,33)	17,34
		18,0	15,5	

Table 2a

Analysis of variance summary table for the total response latency under conditions of semantic and syntactic feedback.

Source	SS	df	MS	F ratio
A (semantic)	5508,5	1	5508,5	0,09
B (syntactic)	6,0	1	6,0	0,00
AB	253,5	1	253,5	0,004
Within	1 260 122	20	63 006	

Table 2b

Cell and treatment means (and cell standard deviations) for the total response latency, in seconds.

		Factor B (syntactic feedback) B1 (absent)	B2 (present)	
Factor A (semantic feedback)	A1 (absent)	525,5 (244,5)	520 (329)	522,8
	A2 (present)	488,7 (159,8)	496,2 (241,8)	492,5
		507,1	508,1	

Table 3a

Analysis of variance summary table for the average response latency under conditions of semantic and syntactic feedback.

Source	SS	df	MS	F ratio
A (semantic)	3 288,17	1	3 288,17	0,26
B (syntactic)	34 492,03	1	34 492,03	2,76
AB	79 241,43	1	79 241,43	6,35^{+}
Within	249 575,8	20	12 478,79	

Table 3b

Cell and treatment means (and cell standard deviations) for the average response latency.

		Factor B (syntactic feedback) B1 (absent)	B2 (present)	
Factor A (semantic feedback)	A1 (absent)	715,33 (130,24)	676,67 (125,69)	696,0
	A2 (present)	577,17 (114,42)	768,0 (63,74)	672,59
		646,52	722,34	

Table 3c

Tukey HSD pairwise comparisons for the average response latency, between cell means.

Difference	Means	Tukey HSD t'(1)
A1B1 - A2B1	715,33 - 577,17	3,03+
A1B1 - A1B2	715,33 - 676,67	0,85
A2B2 - A2B1	768,00 - 577,17	4,18+
A2B2 - A1B2	768,00 - 676,67	2,00

(1) Refer to tables of the Studentised range statistic with degrees of freedom 2 and 20.

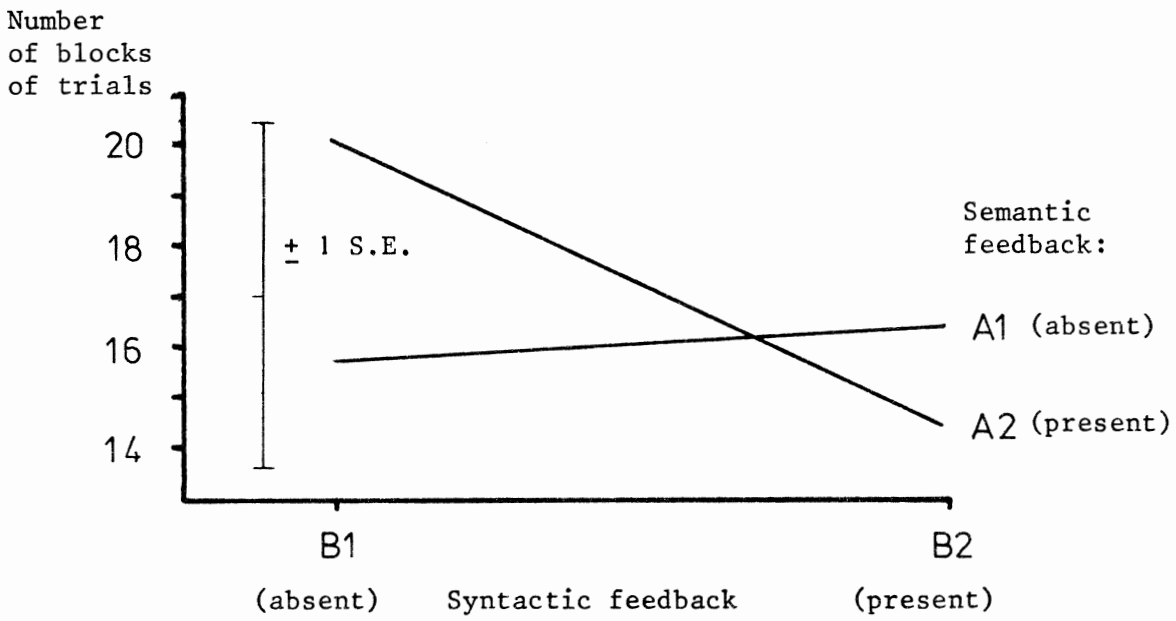


Figure 1

Graph of cell mean profiles for the experimental conditions on the number of blocks of trials to reach criterion.

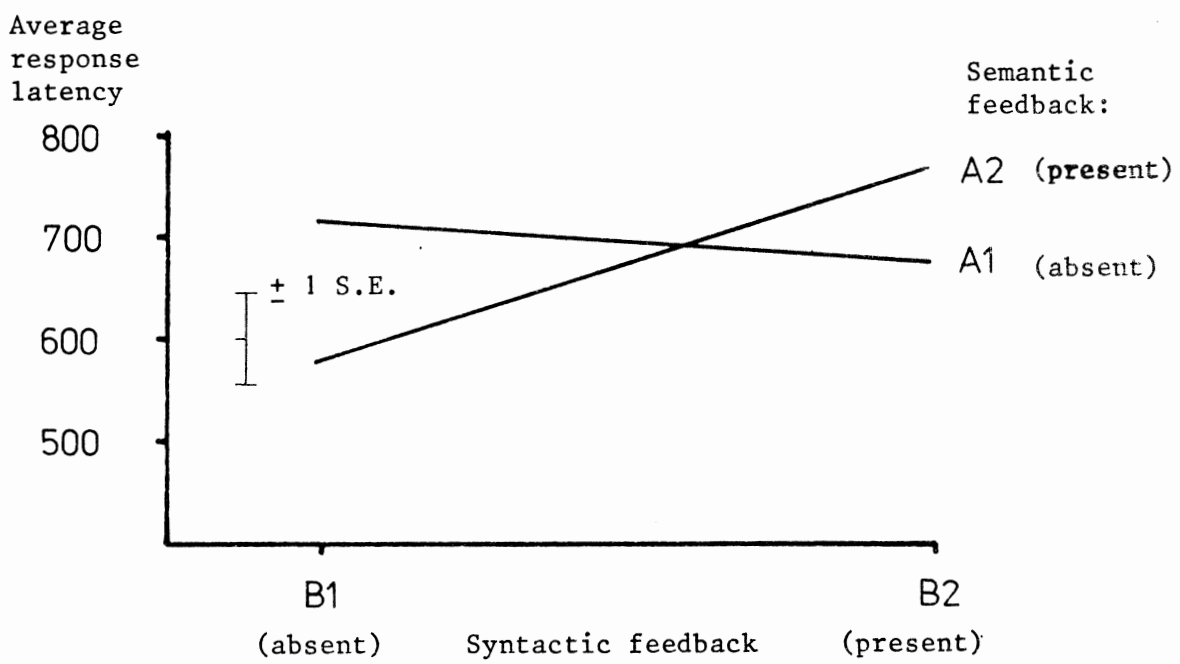


Figure 2

Graph of cell mean profiles for the experimental conditions on the average response latency on feedback trials.

Measures of errors made

The number of copy errors made by the subjects was examined as a check upon the occurrence of possible side-effects of the various feedback conditions, such as confusing the subject. No significant differences were found, indicating that the incidence of copy errors could not be associated with any of the particular feedback conditions. The anova summary table and summary statistics are provided in the appendix.

Of far more interest were the number of class, order, and word errors made by the subjects. The analysis of the class errors is provided in Table 4: the anova summary table in Table 4a, with the summary statistics in Table 4b. Similarly, the analysis of order errors is presented in Table 5, and word errors in Table 6. The graphs of cell mean profiles are presented in Figures 3, 4, and 5 for class, order, and word errors respectively.

The only significant effects were those of factor B (no syntactic versus syntactic feedback) for class and order errors. In particular, factor A showed no significant effect for word meaning errors. These results indicated that the syntactic errors of class and order were significantly affected by syntactic feedback; an examination of the treatment means showed that class and order errors were higher for subjects who did not receive syntactic feedback, and lower for those who did receive syntactic feedback. An examination of the graph of cell mean profiles for word meaning errors, however, seemed to show a possible interaction effect which was presumably being masked by particularly high random sampling variation. The evidence suggests, though, that this interaction is more apparent than real, being well

Table 4a

Analysis of variance summary table for class errors under conditions of semantic and syntactic feedback.

Source	SS	df	MS	F ratio
A (semantic)	125,9	1	125,9	1,2
B (syntactic)	459,4	1	459,4	4,39^{+}
AB	1,1	1	1,1	0,01
Within	2 092	20	104,6	

Table 4b

Cell and treatment means (and cell standard deviations) for class errors.

		Factor B (syntactic feedback) B1 (absent)	B2 (present)	
Factor A (semantic feedback)	A1 (absent)	19,5 (9,57)	11,17 (7,99)	15,34
	A2 (present)	24,5 (13,49)	15,33 (9,00)	19,92
		22,0	13,25	

Table 5a

Analysis of variance summary table for order errors under conditions of semantic and syntactic feedback

Source	SS	df	MS	F ratio
A (semantic)	450.0	1	450,0	1,32
B (syntactic)	1 665,0	1	1 665,0	4,89 +
AB	68,3	1	68,3	0,20
Within	6 812	20	340,6	

Table 5b

Cell and treatment means (and cell standard deviations) for order errors.

		Factor B (syntactic feedback) B1 (absent)	B2 (present)	
Factor A (semantic feedback)	A1 (absent)	33,5 (14,84)	20,17 (13,67)	26,84
	A2 (present)	45,5 (26,77)	25,5 (15,45)	35,5
		39,5	22,84	

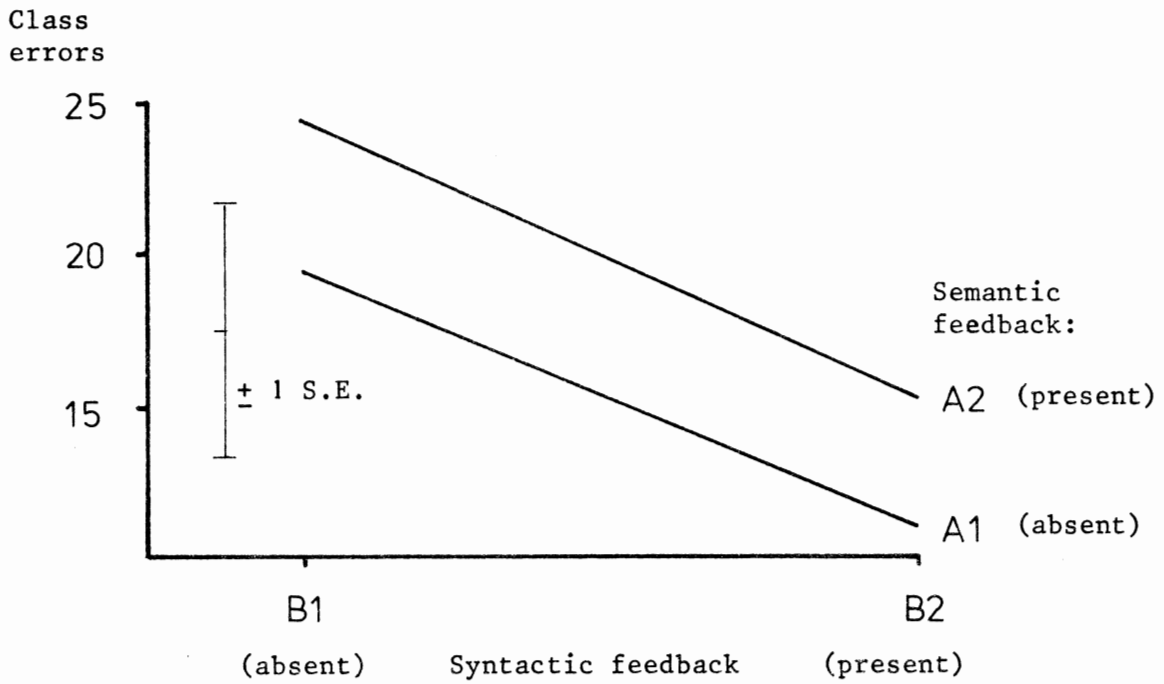


Figure 3

Graph of cell mean profiles for the experimental conditions on the number of class errors made during the experiment.

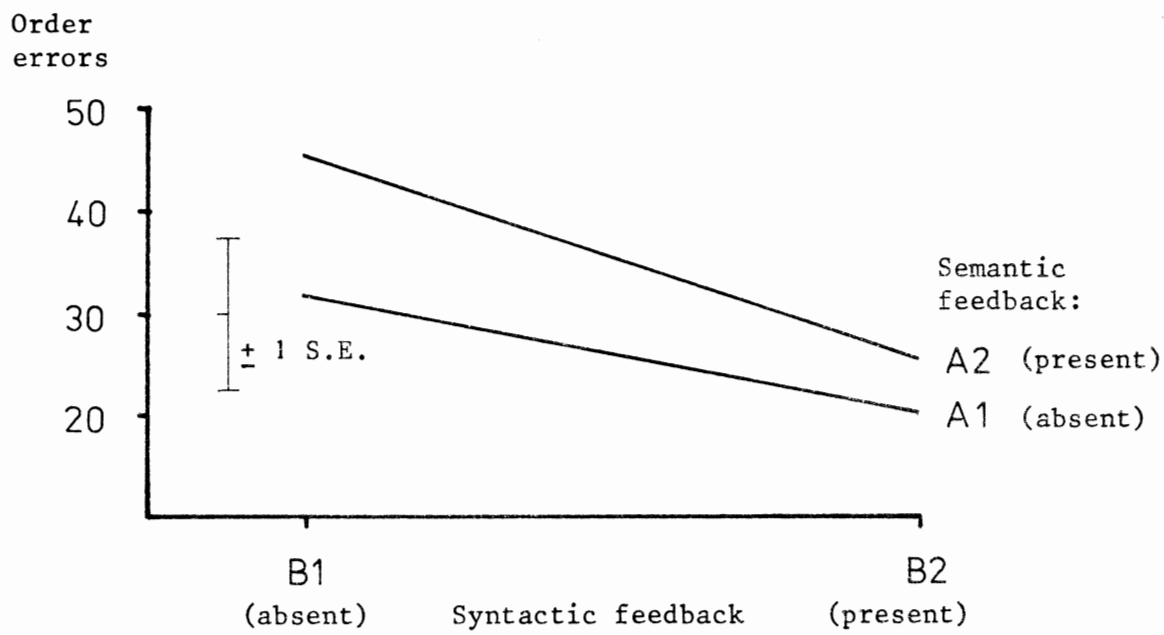


Figure 4

Graph of cell mean profiles for the experimental conditions on the number of order errors made during the experiment.

Table 6a

Analysis of variance summary table for word meaning errors under conditions of semantic and syntactic feedback.

Source	SS	df	MS	F ratio
A (semantic)	59,9	1	59,9	0,05
B (syntactic)	384,0	1	384,0	0,33
AB	433,5	1	433,5	0,37
Within	23 134	20	1 156,7	

Table 6b

Cell and treatment means (and cell standard deviations) for word meaning errors.

		Factor B (syntactic feedback) B1 (absent)	B2 (present)	
Factor A (semantic feedback)	A1 (absent)	65,17 (33,13)	65,67 (36,05)	65,42
	A2 (present)	76,83 (29,92)	60,33 (36,53)	68,58
		71,0	63,0	

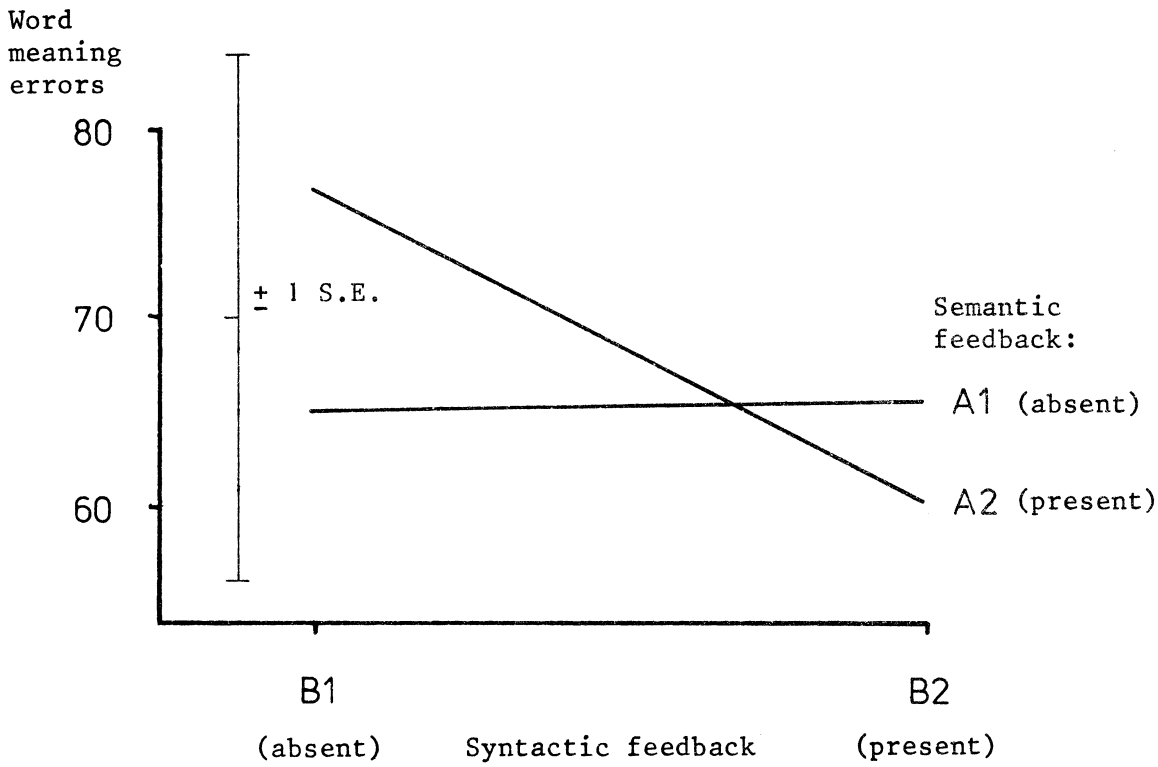


Figure 5

Graph of cell mean profiles for the experimental conditions on the number of word meaning errors made during the experiment.

within the limits of sampling variation. Further data on this issue is examined in the section on the results of the rates of error production.

Measures of time and trials to partial mastery

Three stages in the acquisition of the language were defined: when class errors fell to zero, when order errors fell to zero, and when word meaning errors fell to zero. The number of blocks of trials taken to achieve these stages were calculated for each subject; also, the average response latency in each stage was calculated.

The number of trials taken to achieve zero word meaning errors was found to be identical with the number of trials taken to obtain criterion; that is, it was uniformly the case that when word meaning errors fell to zero, the subject had learned the language to criterion, the other two types of error having fallen to zero either earlier or simultaneously.

The number of trials taken to achieve zero class or order errors was more difficult to determine, since it was often the case that an occasional error would occur after two error-free blocks. Two criteria were adopted to decide upon the block number where an error type first fell to zero. The first was that two error-free blocks should follow the block of putative last error. The second involved the calculation of the probabilities of various error scores under conditions of random, uninformed responding. If it was found that, after two error-free blocks, errors were made which had better than

5% chance of occurring by random responding, than the earlier error-free blocks were ignored for purposes of determining the block of last error. These probabilities may be found in the appendix.

Using the blocks of trials measures defined above, the average response latency for each subject was found for each of the three phases of the experiment. The first phase of the experiment was where no errors had yet fallen to zero; the second phase of the experiment was found to be where class errors had fallen to zero, but where order errors were not yet zero; and the third phase was where class and order errors were zero, but word meaning errors were not. As the description of the three phases suggests, it was found to be the case that the first type of error to fall to zero was the class error; order errors then fell to zero either simultaneously, or afterwards. Then, order errors fell to zero either before, or simultaneously with, word meaning errors.

These data were analysed by three-way analyses of variance with repeated measures on one factor. The A and B factors were as before: A1--no semantic feedback; A2--semantic feedback; B1--no syntactic feedback; B2--syntactic feedback. Factor C comprised three levels: C1, where class errors fell to zero, defining phase 1; C2, where order errors fell to zero, defining phase 2; and C3, where word errors fell to zero, defining phase 3.

The results of the analysis of the number of blocks of trials to complete each phase is presented in Table 7: the three-way anova summary is in Table 7a, with the summary statistics in Table 7b. Figure 6 shows the graphs of cell mean profiles. Two significant effects were found: the ABC interaction, and the C main effect.

The significant ABC interaction indicated that the first-order interactions were non-additive; this may be most clearly seen with reference to the graph of cell mean profiles for the A and B treatments at each level of factor C. There is no, or very little, interaction between factors A and B at C1, or at C2; but at C3, interaction seems to be present. Of course, the graph at C3 is identical with figure 1, the graph of cell mean profiles for the number of blocks of trials to achieve criterion. An analysis of the simple interaction effect AB at C3 would yield results identical with the anova presented in Table 1a; that is, the AB interaction at C3 would be declared to be insignificant. The resolution of this somewhat paradoxical result is to note that the ABC interaction is tested with an error term from which variation due to differences between subjects has been removed, while the test of the AB interaction at C3 is done with an error term which necessarily includes this source of variation. Since the variation due to the subjects is, in fact, extremely large, the AB interaction at C3 is well within sampling variation. The significant ABC interaction does provide evidence, however, for an effect which is, so to speak, just below the surface. An examination of the graph of cell mean profiles, Figure 6, suggests that the profile line of B2 at C3 has an inconsistent trend due to a particularly high cell mean at A1B2C3. The interpretation, to be taken cautiously, is that experimental condition 2, syntactic feedback only, introduces a particular difficulty to subjects in phase 3 of the experiment, since there is a particularly high number of blocks of trials needed to reach criterion.

The nature of the significant ABC interaction allows an analysis

Table 7a

Analysis of variance summary table for the number of blocks of trials to complete each of three experimental phases under conditions of semantic and syntactic feedback

Source	SS	df	MS	F ratio
A (semantic)	112,59	1	112,59	0,59
B (syntactic)	167,99	1	167,99	0,88
AB	43,52	1	43,52	0,23
Subj. w. grps.	3 839,13	20	191,96	
C (phases)	159,14	2	79,57	21,98^{+++}
AC	16,12	2	8,06	2,23
BC	2,87	2	1,44	0,40
ABC	24,82	2	12,41	3,43^{+}
C x SWG	144,64	40	3,62	

Table 7b

Cell, summary, and treatment means (and cell standard deviations) for the number of blocks of trials to complete each experimental phase.

		Factor C (experimental phase) C1 (phase 1)	C2 (phase 2)	C3 (phase 3)
At B1 (no syntactic feedback)				
Factor A (semantic feedback)	A1 (absent)	13,17 (5,56)	13,5 (5,43)	15,83 (6,05)
	A2 (present)	16,83 (7,63)	17,67 (8,87)	20,17 (8,42)
At B2 (syntactic feedback)				
Factor A (Semantic feedback)	A1 (absent)	10,33 (8,59)	11,17 (8,33)	16,5 (9,20)
	A2 (present)	13,17 (9,41)	13,17 (9,41)	14,5 (9,33)

Table 7b (continued)

		Factor C (experimental phases)		
		C1 (phase 1)	C2 (phase 2)	C3 (phase 3)
Factor A (semantic feedback)	A1 (absent)	11,75	12,335	16,165
	A2 (present)	15,0	15,42	17,335
Factor B (syntactic feedback)	B1 (absent)	15,0	15,585	18,0
	B2 (present)	11,75	12,17	15,5
Treatment mean :		13,375	13,878	16,75

		Factor B (syntactic feedback)		Treatment Mean:
		B1 (absent)	B2 (present)	
Factor A (semantic feedback)	A1 (absent)	14,167	12,667	13,417
	A2 (present)	18,223	13,613	15,918
Treatment mean :		16,195	13,14	

Table 7c

Tukey HSD pairwise comparisons for the number of blocks of trials to completion of each experimental phase.

Difference	Means	Tukey HSD t'(1)
C3 - C2	16,75 - 13,878	7,39++
C2 - C1	13,878 - 13,375	1,30
C3 - C1	16,75 - 13,375	8,69++

(1) Refer to table of the Studentised range statistics with degrees of freedom 3 and 40.

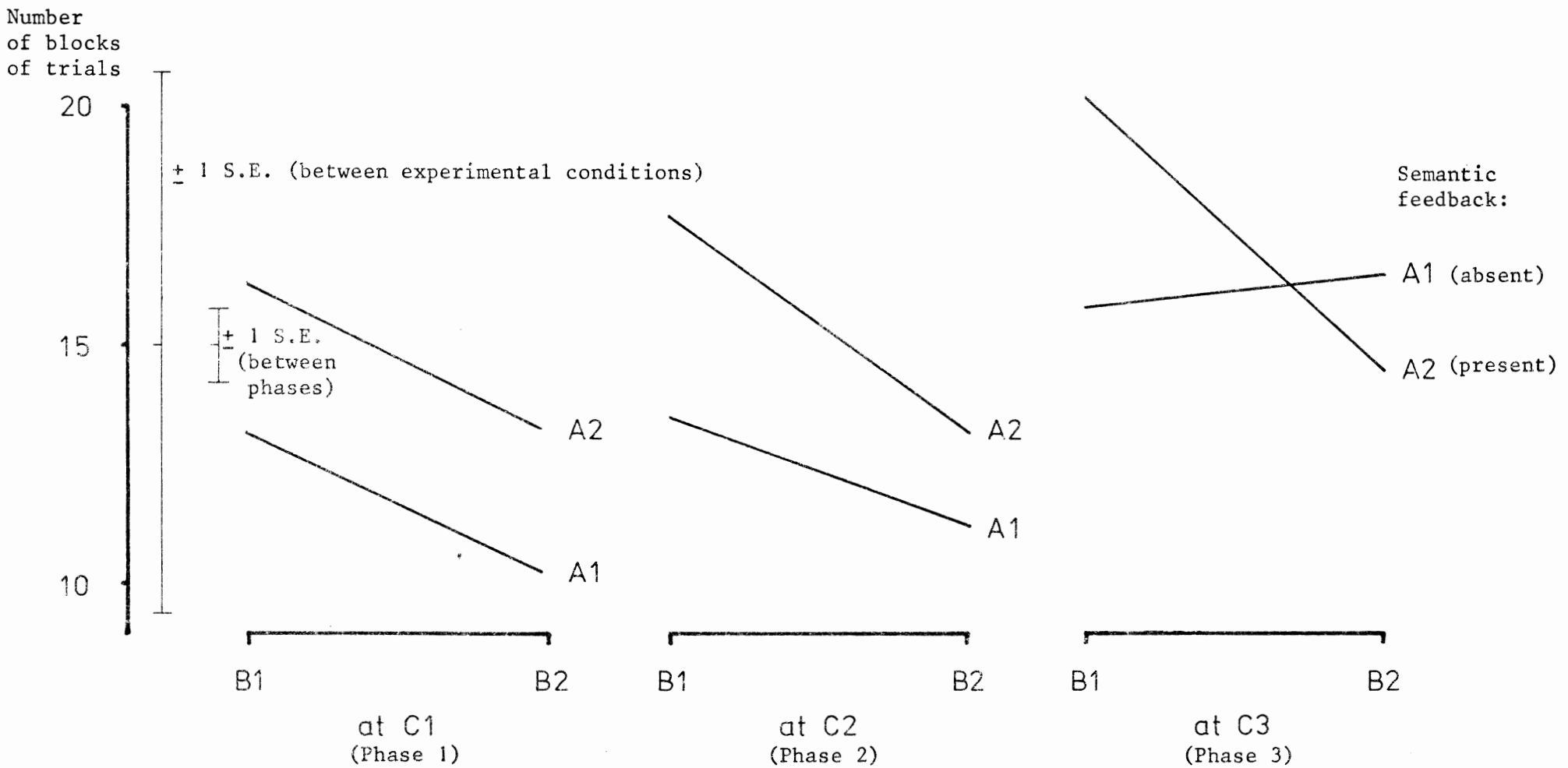


Figure 6

Graph of cell mean profiles for the experimental conditions across phases on number of blocks of trials to criterion.

of the significant C main effect. The Tukey HSD pairwise comparisons between the factor C means is presented in Table 7c. These results indicated that order errors fell to zero significantly earlier than word meaning errors, but not significantly later than class errors. This result is reinforced by the finding, noted earlier, that there was no subject whose order errors fell to zero after his word meaning errors.

The results of the analysis of the average response latencies in each phase of the experiment are presented in Table 8: the three-way anova summary is given in Table 8a, with the summary statistics in Table 8b. The graphs of the relevant cell mean profiles are given in Figure 7. Clearly significant effects were found for the AB interaction, and the C main effect. Effects which were just insignificant were found for the BC interaction, and the B main effect, with p values of 0,063 and 0,067 respectively. Tukey HSD pairwise comparisons were conducted on the AB summary means, and are reported in Table 8c. These showed that significant differences existed between condition 3 (semantic feedback only) and condition 4 (syntactic and semantic feedback), such that the average response latency was lower in condition 3, and between condition 1 (no feedback) and condition 3, such that the average response latency was lower in condition 3 as well. This result is consistent with the results of Table 3.

The just insignificant results of the BC and B effects were examined further, on the basis of two considerations; firstly, that the pursuit of suggestive results would be better data analysis than the slavish adherence to conventional levels of significance, and secondly, that the appropriate analysis of the BC interaction via simple B and C main effects would pool the variation of the BC and

Table 8a

Analysis of variance summary table for the average response latency at each experimental phase under conditions of semantic and syntactic feedback.

Source	SS	df	MS	F ratio
A (semantic)	6 985	1	6 985	0,13
B (syntactic)	193 939	1	193 939	3,74(1)
AB	347 627	1	347 627	6,70+
Subj. w. Grps.	1 037 439	20	51 871,9	
C (phases)	252 887	2	126 444	20,66+++
AC	1 600,2	2	800,1	0,13
BC	36 327,9	2	18 163,9	2,97(2)
ABC	4 660,7	2	2 330,4	0,38
C x SWG	244 863	40	6 121,6	

(1) p = 0,067

(2) p = 0,063

Table 8b

Cell, summary, and treatment means (and cell standard deviations) for average response latency in each experimental phase.

		Factor C (experimental phase)		
		C1 (phase 1)	C2 (phase 2)	C3 (phase 3)
At B1 (no syntactic feedback)				
Factor A (semantic feedback)	A1 (absent)	763,2 (136,7)	771,7 (144,3)	703,0 (110,0)
	A2 (present)	631,0 (160,4)	604,2 (187,6)	526,7 (95,0)
At B2 (syntactic feedback)				
Factor A (semantic feedback)	A1 (absent)	787,2 (140,4)	745,0 (96,9)	600,2 (102,9)
	A2 (present)	889,7 (198,5)	889,7 (198,5)	710,8 (128,1)

Table 8b (continued)

		Factor C (experimental phase)		
		C1 (phase 1)	C2 (phase 2)	C3 (phase 3)
Factor A (semantic feedback)	A1 (absent)	775,2	758,35	651,6
	A2 (present)	760,35	746,95	618,75
Factor B (syntactic feedback)	B1 (absent)	697,1	687,95	614,85
	B2 (present)	838,45	817,35	655,5
Treatment mean:		767,78	752,65	635,18

		Factor B (syntactic feedback)		Treatment mean:
		B1 (absent)	B2 (present)	
Factor A (semantic feedback)	A1 (absent)	745,97	710,8	728.39
	A2 (present)	587,3	830,07	708,69
Treatment mean:		666,64	770,44	

Table 8c

Tukey HSD pairwise comparisons for the average response latency in each experimental phase, between the conditions of semantic and syntactic feedback.

Difference	Means	Tukey HSD t' (1)
A1B1 - A2B1	745,97 - 587,3	2.96^{+}
A1B1 - A1B2	745,97 - 710,8	0,66
A2B2 - A2B1	830,07 - 587,3	$4,52^{++}$
A2B2 - A1B2	830,07 - 710,8	2,22

(1) Refer to tables of the Studentised range statistic with degrees of freedom 2 and 20.

Table 8d

Simple main effects summary table for the average response latency in each experimental phase, under conditions of syntactic feedback.

Source	SS	df	MS	F ratio
C at B1	48 769,6	2	24 384,8	1,14
C at B2	240 445,3	2	120 222,7	$5,63^{++}$
Within	1 281 180	60	21 353	

Table 8e

Tukey HSD pairwise comparisons for the average response latency in each experimental phase, between experimental phases under the condition of syntactic feedback.

Difference	Means	Tukey HSD t'(1)
C1 - C2 at B2	838,45 - 817,35	0,50
C2 - C3 at B2	817,35 - 655,5	3,84+
C1 - C3 at B2	838,45 - 655,5	4,34+

(1) Refer to tables of the Studentised range statistic with degrees of freedom 3 and 40.

Table 8f

Tukey HSD pairwise comparisons for the average response latency in each experimental phase, between conditions of no syntactic, and syntactic feedback.

Difference	Means	Tukey HSD t'(1)
B2 - B1 at C1	838,45 - 697,1	3,35+
B2 - B1 at C2	817,35 - 687,95	3,07+
B2 - B1 at C3	655,5 - 614,85	0,96

(1) Refer to tables of the Studentised range statistic with degrees of freedom 2 and 60.

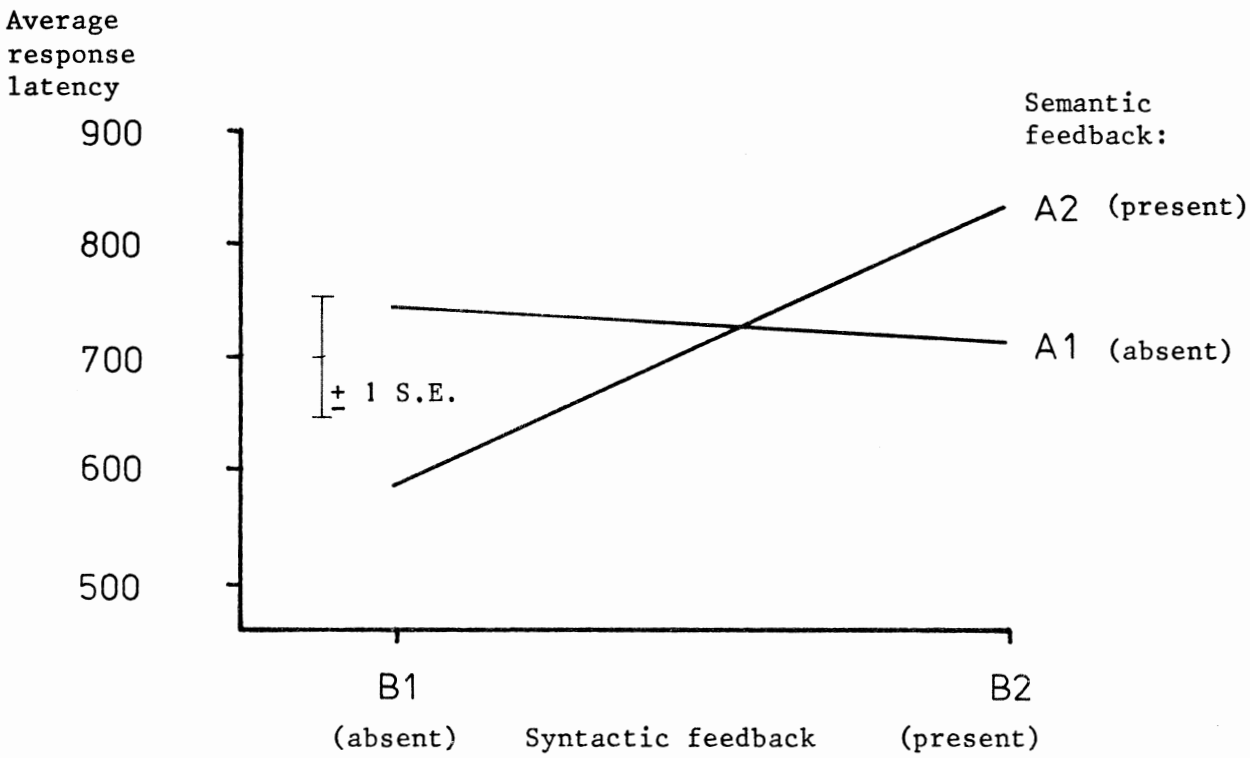


Figure 7a

Graph of cell mean profiles for the experimental conditions on the average response latency to feedback trials, averaged across phases.

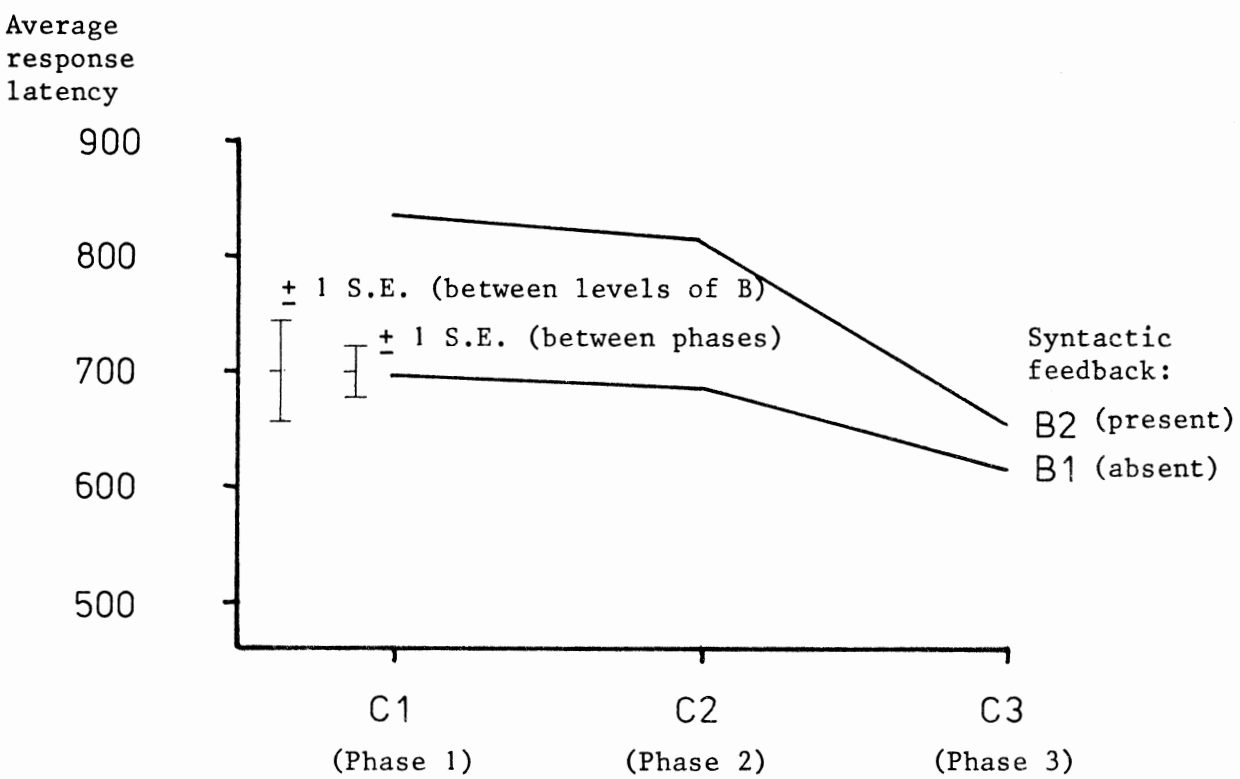


Figure 7b

Graph of summary mean profiles for syntactic feedback and experimental phases averaged across semantic feedback on average response latency.

B sources when the simple main effects of B were examined, allowing the possibility of two almost significant results yielding one fully significant result.

This strategy was successful. The simple main effects of C at B1 and at B2 are reported in Table 8d, while the Tukey HSD pairwise comparisons of the B1 and B2 means at C1, C2, and C3 are reported in Table 8e, with the Tukey comparisons of the C1, C2, and C3 means at B2 being given in Table 8f. These results showed that there were no significant simple main effects of C at B1, but that there were significant simple main effects of C at B2. In particular, the Tukey HSD analysis showed that there was a significant difference between C2 and C3 at B2. This indicated that the average response latency was not significantly different for all phases of the experiment for subjects under condition B1, no syntactic feedback. On the other hand, for subjects under condition B2, syntactic feedback present, a significant drop in response latency was observed in phase 3 of the experiment. The Tukey HSD results of Table 8f indicated that the average response latency of subjects who received syntactic feedback was significantly higher than that of subjects who received no syntactic feedback in phases 1 and 2, but not phase 3, of the experiment.

Measures of error rates

Although no significant differences were observed in the number of trials to obtain criterion (Table 1a), there was some evidence to suggest that there was an interaction between the feedback conditions (Table 7a).

In addition, a question about the rate of commission of errors is relevant; that is, was there any evidence that any of the feedback conditions allowed a reduced rate of error production, regardless of whether the total number of errors showed any effects. These considerations lead to the derivation of error-rate data: for each subject, the total number of class, order, and word errors was divided by the number of blocks of trials needed to reach criterion for the mastery of the language.

The form of the analysis was the two-way anova, with factors A and B as before. Tables 9, 10, and 11 present the results of the anova (a) and the summary statistics (b) for class, order, and word error rates respectively.

Significant effects were found for factor B on class and order error rates only. These findings are entirely consistent with the analysis of the total number of class, order, and word errors. They indicated that a significant drop in the error-rate of class and order errors was associated with conditions of syntactic feedback. There was no evidence for any drop in word meaning error-rate; in particular, the F ratios for the word error-rates are all totally insignificant.

Post-test measures

The results of the post-test were analysed via two-way anovas, with factors A and B as before. Tables 12 and 13 present the results of the anova (a) and the summary statistics (b) for the number of sentences correctly identified as grammatical, and for the number of words whose meaning was known, respectively.

Table 9a

Analysis of variance summary table for class error rate under conditions of semantic and syntactic feedback.

Source	SS	df	MS	F ratio
A (semantic)	0,163	1	0,163	1,24
B (syntactic)	0,844	1	0,844	6,43^{+}
AB	0,257	1	0,257	1,96
Within	2,625	20	0,131	

Table 9b

Cell and treatment means (and cell standard deviations) for class error rates.

		Factor B (syntactic feedback)		
		B1 (absent)	B1 (present)	
Factor A (semantic feedback)	A1 (absent)	1,265 (0,409)	0,683 (0,328)	0,974
	A2 (present)	1,223 (0,391)	1,055 (0,312)	1,139
		1,244	0,869	

Table 10a

Analysis of variance summary table for order error rate under conditions of semantic and syntactic feedback.

Source	SS	df	MS	F ratio
A (semantic)	0,565	1	0,565	0,77
B (syntactic)	3,773	1	3,773	5,12^{+}
AB	0,273	1	0,273	0,37
Within	14,75	20	0,737	

Table 10b

Cell and treatment means (and cell standard deviations) for order error rates.

		Factor B (syntactic feedback) B1 (absent)	B2 (present)	
	A1 (absent)	2,25 (1,025)	1,243 (0,587)	1,747
(semantic feedback)	A2 (present)	2,343 (1,072)	1,765 (0,645)	2,054
		2,297	1,504	

Table 11a

Analysis of variance summary table for word meaning error rate under conditions of semantic and syntactic feedback.

Source	SS	df	MS	F ratio
A (semantic)	0,001	1	0,001	0,00
B (syntactic)	0,030	1	0,030	0,04
AB	0,031	1	0,031	0,04
Within	17,09	20	0,855	

Table 11b

Cell and treatment means (and cell standard deviations) for word meaning error rates.

		Factor B (syntactic feedback) B1 (absent)	B2 (present)	
Factor A (semantic feedback)	A1 (absent)	3,995 (0,814)	3,993 (1,132)	3,994
	A2 (present)	3,907 (0,851)	4,05 (0,866)	3,979
		3,951	4,022	

Table 12a

Analysis of variance summary table for the number of sentences correctly identified as correct under conditions of semantic and syntactic feedback.

Source	SS	df	MS	F ratio
A (semantic)	1,5	1	1,5	3,09
B (syntactic)	0,65	1	0,65	1,35
AB	0,0	1	0,0	0,0
Within	9,7	20	0,485	

Table 12b

Cell and treatment means (and cell standard deviations) for the number of sentences correctly identified as correct.

		Factor B (syntactic feedback)		
		B1 (absent)	B2 (present)	
Factor A (semantic feedback)	A1 (absent)	10,0 (0,0)	9,67 (0,82)	9,84
	A2 (present)	9,5 (0,84)	9,17 (0,75)	9,34
		9,75	9,42	

Table 13a

Analysis of variance summary table for the number of words known for their meaning under conditions of semantic and syntactic feedback.

Source	SS	df	MS	F ratio
A (semantic)	5,08	1	5,08	1,52
B (syntactic)	0,04	1	0,04	0,01
AB	7,01	1	7,01	2,10
Within	66,76	20	3,34	

Table 13b

Cell and treatment means (and cell standard deviations) for the number of words known for their meaning.

		Factor B (syntactic feedback) B1 (absent)	B2 (present)	
Factor A (semantic feedback)	A1 (absent)	6,0 (0,0)	4,83 (2,04)	5,42
	A2 (present)	4,0 (2,53)	5,0 (1,67)	4,5
		5,0	4,92	

As was expected, no significant results occurred. This indicates that all the subjects could distinguish between grammatical and non-grammatical sentences regardless of the form of feedback to which they had been exposed, and that a similar result held for knowledge of the meaning of the words in the vocabulary. What may be more interesting was the fact that four subjects knew the English meaning of only two or less of the words, yet could achieve criterion in learning the language.

Summary

Taking the learning of the language as a whole, the results show that all subjects were equally proficient at the end of the experiment (no significant differences on the post-test measures), and that all subjects learned the language in about the same number of trials and in the same overall time taken to respond to feedback trials (no significant differences on the number of blocks of trials to criterion, and no significant differences in the total amount of time spent in producing sentences). Fewer grammatical errors, however, were made by subjects who received syntactic feedback about their errors, regardless of whether or not they received semantic feedback (significant syntactic feedback main effect, insignificant interaction and semantic feedback main effects, for class and order errors and class and order error-rates). Somewhat surprisingly, all subjects produced about the same number of semantic errors, regardless of feedback condition (no significant effects on the number of word meaning errors and word error-rate). Finally, subjects who received semantic feedback only responded more quickly in feedback trials than subjects who received no feedback, and more quickly than subjects who received both semantic and syntactic feedback (significant interaction, and sig-

nificant Tukey HSD pairwise comparisons as indicated, on the average response latency).

Examining the learning of language in its phases provided a confirmation of the differences in average response latency which were found for the experiment as a whole; in addition, these differences were the same at each phase of the experiment (significant interaction between the feedback conditions, but no significant second-order interaction). A more important finding was that the average response latency was higher during the early and middle part of the experiment (phases 1 and 2) for subjects who received syntactic feedback, but dropped to equal the average response latency of subjects who did not receive syntactic feedback in the last part (phase 3) of the experiment. This finding applied regardless of whether the subjects received semantic feedback or not (significant interaction between phases and syntactic feedback, no significant second-order interaction, and significant Tukey HSD pairwise comparisons where indicated). A difference was inferred for the number of trials needed to complete the last phase of the experiment for subjects who received syntactic feedback only, such that these subjects took longer than was expected (no significant results except a significant second-order interaction, with the sense of the graph of cell mean profiles alllowing the inference).

Discussion

What is the general tenor of the results? Firstly, the feedback effects were rather weak; giving the subject information about his errors of syntax, or semantics, or both, or neither, made no effective difference to the number of trials needed to learn the language, nor to the overall time taken to do it. This latter measure must be clarified, however, precisely what is meant is that there were no feedback effects upon the total amount of time spent by the subject in producing sentences upon request.

Though the feedback effects were rather weak, syntactactic feedback did produce some significant results. In particular, this form of feedback served to lower the total number, and rate, of grammatical errors.

Semantic feedback, on the other hand, seemed to be quite ineffective in the experiment.

The results of the average response latencies, however, are more

difficult to interpret. A diagram of the response surface for the experiment is given in figure 8. There are two interactions present here, and the easier one to begin with is the relationship between syntactic feedback and the experimental phases. For the first two experimental phases, where the subject is presumed to be mastering the grammar of the language, the average response latency of subjects receiving syntactic feedback is higher than the latency of subjects receiving no syntactic feedback. On the final experimental phase, however, there is no difference between these two groups of subjects in terms of their average response latencies; in fact, the latency of subjects who received syntactic feedback has dropped to become roughly equal to those who did not receive such feedback. Now this result holds regardless of whether the subject received semantic feedback or not; that is, the pattern of the drop in phase three is the same whether the subjects received syntactic feedback only, or syntactic plus semantic feedback.

The second interaction is that between the forms of feedback: subjects who received both semantic and syntactic feedback had higher response latencies than subjects who received semantic feedback only, while subjects who received semantic feedback only had the same average latencies as subjects who received no feedback. As the response surface shows, these latter subjects were intermediate in latency between the former two experimental conditions. This pattern of results holds at each experimental phase.

Some comments are in order before implications are drawn from the results. The first is the fact that there was considerable variability between the subjects in their ability to learn the language. One subject learned the language in four blocks of trials, while another took 32

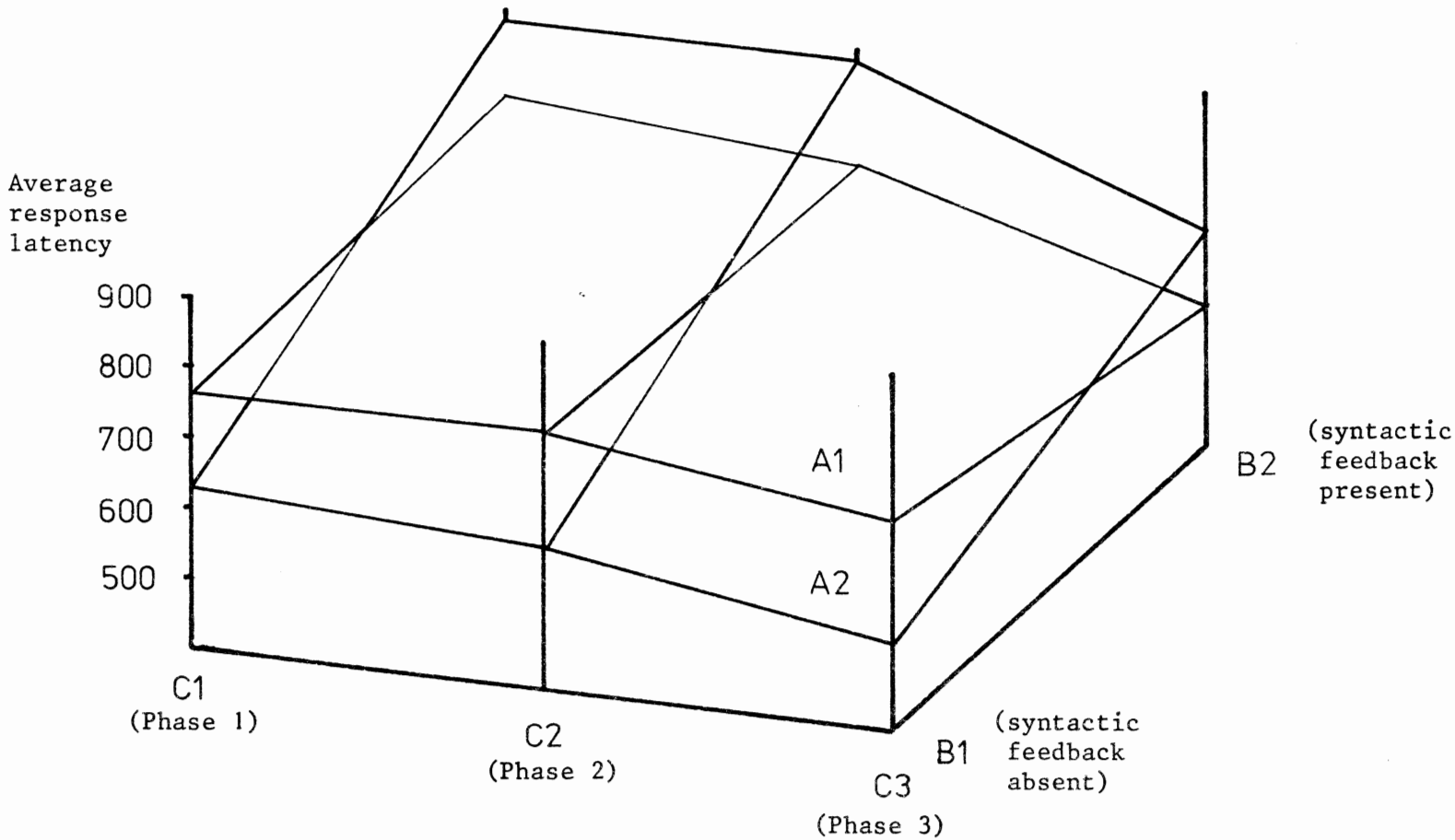


Figure 8

Graph of the cell mean profiles for the experimental conditions across phases on average response latency.

blocks; this is the range which occurs in each experimental condition. It is thus clearly evident that the subjects employed widely differing problem-solving strategies and, despite the lengthy instructions, experienced quite different experimental demand characteristics (Orne, 1962; Rosenthal, 1963). Despite this high variability, some results are significant, in some cases extremely so when the form of statistical analysis has been able to partial out the variability due to the subjects. In other cases, averages or rates have been calculated which have a similar effect of making the subjects more comparable in their results, so that the data in which I am inclined to place full confidence are those dealing with the error rates, and the average response latency in each experimental phase.

The second comment is on the very clear course of learning which every subject seemed to follow, no matter how quickly or slowly he learned the language. That is, errors of word meaning only fell to zero at the end of the experiment, while the grammatical errors of class and order fell to zero significantly earlier on average. It was almost certainly the case that learning of syntax and semantics was progressing in parallel, but the evidence suggests that the learning of word meanings was not completed at the time that the grammar was fully mastered. It must be noted that any error of class necessarily entails an order error and a word meaning error, since a class error occurs when a necessary word in a sentence has been left out. When a class error does not occur, however, the order and word errors are quite independent measures.

Interpretation

What is being measured by the average response latencies? A hint is provided by the inferred finding that subjects who received syntactic

feedback only, took longer than would be expected in phase three, when the grammar had been mastered but the word meanings were still being learned. Such a result is natural; the necessary information on word meaning was not available for the subject. A similar result is observed for the response latencies; there is a significant drop, for subjects who received syntactic feedback, in phase three. The suggestion is that the response latency is measuring the amount of information which the subject has available from the feedback. If there is very little information, then the subject will respond quickly; if there has been a lot of information, the subject will respond more slowly: presumably, the response latency is measuring the amount of thought and concentration that the subject is putting into each trial. Alternatively, if the subject wants to see the next model as soon as possible, his response latency will be quicker than if he is in no particular hurry to see the next model--i.e. if he has some information already which requires processing.

Implications

A paradox arises when the results are drawn together: the subject makes grammatical errors at a lower rate when he is given syntactic feedback (whether by itself or with semantic feedback) than when he is not given such feedback, yet he does not learn the language any more quickly. In the same vein, the subject takes longer to respond to each request to describe a picture when he is presumed to have and to be working on relevant information; yet he does not learn the language in any fewer trials. In some fashion, the acquisition process seems to be functionally independent of the errors made during acquisition.

The implications of this finding (assuming its accuracy) are quite startling. Consider an experiment, like the present one, where the language under study was far more complex, such that the subjects were studied (exposed to various forms of feedback) at some mid-point in their acquisition, there being no reasonable possibility of studying them from start to finish. Then, the reduced rate of grammatical errors would be adduced as evidence for some facilitative and enhancing effect of syntactic feedback. As is illustrated in the present experiment, such a conclusion may not necessarily follow; the failure of syntactic feedback to accelerate the acquisition of syntax in phases one and two (there was no significant difference in the number of trials taken to reach zero order errors between the various feedback conditions), the failure of semantic feedback to accelerate the learning of word meanings, and the equal time and trials taken overall to learn the language, suggest far more strongly that the drop in error-rate represents the fine-tuning of some sub-system which is not directly implicated in the acquisition of the language. Speculative as this conclusion may be, it serves to indicate the importance of studies of language acquisition where the entire process from start to finish can be examined in full perspective.

Does the finding that grammatical errors fall to zero before semantic errors suggest that the acquisition of word meanings depends, to some extent, upon the mastery of the syntax; or was it simply the case that the syntax of the artificial language was more easy to learn than the meanings of the words? The data shows that not one subject learned the meaning of the words before mastering the grammar; it would have been expected, had the acquisition of these two aspects been independent, that at least one or two subjects should obtain zero word errors before zero order errors. It is almost certainly the case, however, that the syntax of the language was easier than the semantics, since there were

better chances of being right in syntax than in word meaning; the exact probabilities are shown in the appendix. To put it another way, the theory of communication and information (Shannon & Weaver, 1949) states that the information in a sequence is inversely related to its probability; hence there is more information, and more to learn, in the word meanings.

The data is not strong enough to decide the question put in the previous paragraph; I would suggest that the syntax was simply easier, since a few subjects reported, in the post-experimental de-briefing, that they took some time to realise what aspects of the geometrical shapes were relevant in deciding what the words meant.

The thrust of the evidence is that the feedback of information to the subject did not really help him; certainly, there were no dramatic differences between the four experimental conditions. Yet, on the other hand, the subject did learn the language. In particular, subjects in condition 1, no feedback, performed about as well as subjects in condition 4, complete feedback; they must have been obtaining and utilising information from somewhere. The inference, of course, is that the subjects were using the models which they were given every alternate trial, and which they were required to copy, as sources of information about the grammar and semantics of the language. That the subjects should use the information provided by the models is not unexpected; but that they should rely exclusively upon the models certainly is. What is particularly puzzling is the finding that condition 4 did not have marked effects; that is, there is a certain unusualness, a certain difficulty, with the syntactic only and semantic only feedback, since these forms have no counterpart in natural language, so it would have been expected that these conditions (2 and 3) could be more difficult than either conditions 1 or 4; but in fact this was not the case. The perfectly

natural form of feedback given in condition 4 was as good as that given in condition 2 in the effects of syntactic feedback upon error rates, and was as good as conditions 1, 2, or 3 in the number of blocks of trials taken to reach criterion.

The implication is either that the feedback was simply ignored, or that it was somehow not useful information. The evidence of the response latency measures suggests very strongly that the subjects took a lot of time processing the feedback of condition 4, took just a little time on semantic feedback only, and a moderate amount of time on the feedback of conditions 1 and 2. The evidence is also that the subjects knew when a form of feedback was no longer useful, as when the response latency dropped in phase three for subjects who were receiving syntactic feedback. The conclusion, then, is that the subjects were processing the feedback, they could evaluate its usefulness, but they could not use it effectively enough to make much of a difference to their learning of the language; while the semantic feedback was not/could not be used at all, the syntactic feedback brought a reduction in grammatical errors.

Perhaps the results can be interpreted as follows: syntactic feedback enjoyed some weak effects, and so it seems that the learning of syntax can be assisted by informational feedback about grammatical errors; semantic feedback enjoyed no effects at all, and so it seems that the learning of work meanings must be accomplished from the models presented to the subject.

The choice of subjects

The ideal subject in an experiment of this nature is presumably the three-year old child learning his first language, if the results are to have direct relevance to issues of the processes underlying

language acquisition. Since the present experiment employed university students, it needs to be shown that the results have some relevance to language learning in the child.

The first point that must be made is that it was not possible to use children in this experiment, due to practical considerations. In particular, the subjects were required to read and understand a set of instructions which were quite complex, and to interact with a computer via a keyboard requiring a minimum level of fine motor co-ordination. There are, of course, other difficulties, such as the short attention span of children, the necessity of either taking the computer system to their home or spending considerable time familiarising them with the laboratory rooms, and so on. Obviously, none of these were problems which would absolutely exclude children as subjects; but they were sufficiently daunting to force the (reluctant) choice of other subjects.

The second point is that the experiment was more illustrative of the possibilities of a certain methodology than it was concerned with substantive findings. The relevance of the experiment was mainly directed at assessing the general usefulness and feasibility of laboratory studies of language acquisition.

Finally, the question must be asked whether, in fact, children would necessarily be the only subjects which would allow the findings to be relevant to questions of language acquisition in the child. The point is whether the processes underlying first language learning in children bear any resemblance to the processes underlying some artificial language learning in young adults. With reference to the present experiment, the experimental procedures were designed in such a way as to minimise the use of cognitive problem-solving strategies on the part of the subject, while the general theoretical framework on

which the experiment was based asserts an essential continuity in the learning processes of all organisms. With these two provisions, I would argue that the results of the present experiment do have some implications for language learning in children. These are explored in the next section; it may finally be noted that Moeser and Olson (1974) found very similar results when a miniature artificial language was used with children as when it was used (Moeser & Bregman, 1972) with students.

Relation to other studies

The form of the results are, in general, quite consistent with Cazden's (1965) findings on the effectiveness of expansions and modelling. Cazden found that models were most effective in raising the linguistic competence of the children in his study; the present experiment has shown that the student subjects relied upon models for the learning of word meanings and mostly for the learning of syntax. Also, Cazden found no evidence that expansions aided the acquisition of grammar; the present experiment found only weak effects for syntactic feedback. Of course, "expansions" and "syntactic feedback" are not at all the same thing; the latter is an operational definition of the former for purposes of laboratory study. Nonetheless, there is a broad agreement between the two sets of results.

The results also have some relevance to MacNamara's (1972) thesis that the grammar of the language is acquired through the prior semantic competence of the child. Put in a rather simplistic form, the child is presumed to acquire word order on the basis of his knowledge of word meaning. Such a theory is not supported by the present experiment, where it was found that, on the contrary, word order was always acquired before or simultaneously with word meaning. Even though the syntax was probably easier to learn, MacNamara's thesis would still maintain that semantics preceeds syntax.

Finally, the results (obstinately) support the nativist position on language learning. It was argued earlier, of course, that a radical nativist claim about genetic endowment and the LAD cannot be supported, on purely metatheoretical grounds; but if the claim is taken, instead, to indicate that there is more in the language acquisition of the child than is dream't of in current psychological philosophies, then the

nativists are possibly quite correct in pointing this out. How are the results to be interpreted on a cognitive basis? Perhaps they cannot, because of two shortcomings of the experiment.

Criticisms of the experiment

There is no data on the activities of the subject during the modelling trials. In view of the fact that most of the learning seemed to occur on these trials, this lack is crucial. There would, of course, have been no difficulty in arranging for the computer program to time the response latencies of the subject on these modelling trials; that this was not done was a simple (unfortunate) oversight on my part, since I was quite sure that the major dynamics of the language learning would be located on the feedback trials. Such are the costs of preconceptions.

The second criticism is a more fundamental one. There is evidence for considerable inter-subject variability, such that different subjects were presumed to approach the experiment in different ways. Under such circumstances, an individual analysis of each subject's performance during the course of learning the language would have been more appropriate, rather than pooling the group results and conducting statistical tests on the averages. This form of data gathering has been illustrated most thoroughly by Newell and Simon (1972), who consider it to be the optimal strategy in the investigation of complex problem-solving. This point needs very careful consideration: in very general terms, where each subject in an experiment apparently conducts himself in a unique and different way, then an individual approach is needed; but where the subjects appear to act in similar ways, then the group approach is satisfactory. For the present experiment, a pilot study revealed the individual differences which have been noted, and certain methodological

nativists are possibly quite correct in pointing this out. How are the results to be interpreted on a cognitive basis? Perhaps they cannot, because of two shortcomings of the experiment.

Criticisms of the experiment

There is no data on the activities of the subject during the modelling trials. In view of the fact that most of the learning seemed to occur on these trials, this lack is crucial. There would, of course, have been no difficulty in arranging for the computer program to time the response latencies of the subject on these modelling trials; that this was not done was a simple (unfortunate) oversight on my part, since I was quite sure that the major dynamics of the language learning would be located on the feedback trials. Such are the costs of preconceptions.

The second criticism is a more fundamental one. There is evidence for considerable inter-subject variability, such that different subjects were presumed to approach the experiment in different ways. Under such circumstances, an individual analysis of each subject's performance during the course of learning the language would have been more appropriate, rather than pooling the group results and conducting statistical tests on the averages. This form of data gathering has been illustrated most thoroughly by Newell and Simon (1972), who consider it to be the optimal strategy in the investigation of complex problem-solving. This point needs very careful consideration: in very general terms, where each subject in an experiment apparently conducts himself in a unique and different way, then an individual approach is needed; but where the subjects appear to act in similar ways, then the group approach is satisfactory. For the present experiment, a pilot study revealed the individual differences which have been noted, and certain methodological

changes were then made in an attempt to remove these differences. These changes were not fully successful.

Conclusions

While there are problems about the relevance of the results to language acquisition in the child, and while there are some data which it would have been useful to have and which were not collected, the general thrust of the experiment indicated that explicit informational feedback about errors did not help subjects to learn the language in any fewer trials. Rather, the subjects are presumed to have relied upon the models for information about the language syntax and semantics. There is weak evidence that word meanings were obtained only from the models, while aspects of syntax were partially obtained from the feedback. The subjects certainly did pay attention to, and process, the feedback; the problem was that they could not utilise the information effectively. Finally, the grammatical characteristics of the language were learned before the word meanings were finally mastered.

The findings are in general agreement with Cazden's (1965) results; there, the children benefited most from models, and seemed unable to utilise the information contained in the expansions.

The experiment illustrated a methodology, based upon miniature artificial languages and computer-control of experimental environments, which was suggested for the investigation of the phenomena of language acquisition.

The implications for further studies are that most attention needs to be given to the individual examination of each subject's performance; the reasons for the inability of the subject to utilise informational feedback needs investigation; and so does the way in which models are utilised.

References

Aaronson, D., & Brauth, S. SIMPLE guidelines for developing a computer-based laboratory. Behaviour Research Methods and Instrumentation, 1972, 4, 257-267.

Aaronson, D., Grupsmith, E., & Aaronson, M. The impact of computers on cognitive psychology. Behaviour Research Methods and Instrumentation, 1976, 8, 129-138.

Abramovitz, A. Experimental transformation of the perception of colour. Unpublished Doctoral dissertation, University of Cape Town, 1972.

Anderson, J. R. Language, memory, and thought. New York: Lawrence Erlbaum, 1976.

Anderson, J. R., & Bower, G. H. Human associative memory. New York: Lawrence Erlbaum, 1973.

Annett, J. Feedback and human behaviour. London: Penguin, 1969.

Arbib, M. A. The metaphorical brain. New York: Wiley, 1972.

Archer, E. J. A re-evaluation of the meaningfulness of all possible CVC trigrams. Psychological Monographs, 1960, 74, whole no. 497.

Bandura, A., & Walters, R. H. Social learning and personality development. New York: Holt, Rinehart, & Winston, 1963.

Bellugi, U., & Brown, R. (Eds.) The acquisition of language. Monographs of the Society for Research in Child Development, 1964, no. 92.

Berko, J. The child's learning of English morphology. Word, 1958, 14, 150-177.

Black, M. Comment on Chomsky's "Problems of explanation in linguistics". In R. Borger & F. Cioffi (Eds.), Explanation in the behavioural sciences. Cambridge: Cambridge University Press, 1970.

Bloom, L. Language development: Form and function in emerging grammars. Cambridge, Mass.: M.I.T. Press, 1970.

Bloom, L. Language development. In F. D. Horowitz (Ed.), Review of child development research. Chicago: University of Chicago Press, 1975.

Bloom, L., Lightbrown, P., & Hood, L. Structure and variation in child language. Monographs of the Society for Research in Child Development, 1975, 40, no. 160.

Bowerman, M. Structural relationships in children's utterances: Syntactic or semantic? In T. E. Moore (Ed.), Cognitive development and the acquisition of language. New York: Academic Press, 1973.

Braine, M. D. S. The ontogeny of English phrase structure: The first phase. Language, 1963, 39, 1-13.

Braine, M. D. S. The insufficiency of a finite state model for verbal reconstructive memory. Psychonomic Science, 1965, 2, 291-292. (a)

Braine, M. D. S. Inferring a grammar from responses: Discussion of Gough and Segal's comment. Psychonomic Science, 1965, 3, 241-242. (b)

Braine, M. D. S. Learning the position of words relative to a marker element. Journal of Experimental Psychology, 1966, 72, 532-540.

Braine, M. D. S. The acquisition of language in infant and child. In C. E. Reed (Ed.), The learning of language. New York: Appleton-Century-Crofts, 1971.

Braithwaite, R. B. Scientific explanation. Cambridge: Cambridge University Press, 1953.

Brown, R. Psycholinguistics. New York: Free Press, 1970.

Brown, R., & Bellugi, U. Three processes in the child's acquisition of syntax. Harvard Educational Review, 1964, 34, 133-151.

Brown, R., Cazden, C., & Bellugi, U. The child's grammar from I to III. In J. P. Hill (Ed.), Minnesota symposium on child psychology, Vol. 2. Minneapolis: University of Minnesota Press, 1969.

Brown, R., & Fraser, C. The acqusition of syntax. In C. N. Cofer & B. S. Musgrave (Eds.), Verbal behaviour and learning. New York: McGraw-Hill, 1963.

Burkhardt, J. S. EMPP: An extensible multiprogramming system for experimental psychology. Behaviour Research Methods and Instrumentation, 1975, 8, 239-244.

Cazden, C. Environmental assistance to the child's acquisition of grammar. Unpublished Doctoral dissertation, Harvard University, 1965.

Cazden, C. Two paradoxes in the acquisition of language structures and functions. Paper presented at the conference on The development of competence in early childhood, London, CIBA Foundation, 1972.

Chomsky, N. Syntactic structures. The Hague: Mouton, 1957.

Chomsky, N. Review of Skinner's Verbal behaviour. Language, 1959, 35(1), 26-58.

Chomsky, N. Aspects of the theory of syntax. Cambridge, Mass.: M.I.T. Press, 1965.

Chomsky, N. Recent contributions to the theory of innate ideas. In Boston studies in the philosophy of science. New York: Humanities Press, 1968.

Chomsky, N., & Halle, H. The sound pattern of English. New York: Harper & Row, 1968.

Clark, E. V. What's in a word? On the child's acquisition of semantics in his first language. In T. E. Moore (Ed.), Cognitive development and the acquisition of language. New York: Academic Press, 1973.

Clark, H. H., & Clark, E. V. Psychology and language. New York: Harcourt Brace Jovanovich, 1977.

Creelman, D. D. Rapid response and flexible experimental control with a small on-line computer: PSYCLE. Behaviour Research Methods and Instrumentation, 1971, 3, 265-267.

Dalenoort, G. J. (Ed.) Process models for psychology. Rotterdam: Rotterdam University Press, 1973.

Doll, T. J. A 4k computer language for experimentation with human subjects. Behaviour Research Methods and Instrumentation, 1972, 4, 27-31.

du Preez, P. D. The development of language functions. South African Journal of Psychology, 1975, 5, 29-40.

Durell, A. B. Painless computer-controlled experimentation. Behaviour Research Methods and Instrumentation, 1972, 4, 165-166.

Ervin, S. M. Imitation and structural change in children's language. In E. H. Lenneberg (Ed.), New directions in the study of language. Cambridge, Mass.: M.I.T. Press, 1964.

Estes, W. K. Towards a statistical theory of learning. Psychological Review, 1950, 57, 94-107.

Estes, W. K. Reinforcement in human behaviour. American Scientist, 1972, 60, 723-729.

Fodor, J. A. Could meaning be an r_m? Journal of Verbal Learning and Verbal Behaviour, 1965, 4, 73-81.

Fodor, J. A., Bever, T. G., & Garrett, M. F. The psychology of language. New York: McGraw-Hill, 1974.

Getty, D. J. The PEPL system for control of experiments by a PDP-8 computer. Behaviour Research Methods and Instrumentation, 1975, 7, 131-136.

Gilbert, L. H. The computer-based control and analysis of complex experiments. Paper presented at the 23rd. Annual Congress of the South African Psychological Association, 1971.

Glucksberg, S., Krauss, R., & Higgins, E. T. The development of referential communication skills. In F. D. Horowitz (Ed.), Review of child development research. Chicago: University of Chicago Press, 1975.

Haber, R. N., Barry, S. H., & Uhlman, T. On-line FORTRAN for the PDP-8. Behaviour Research Methods and Instrumentation, 1970, 2, 248-252.

Halle, M., & Stevens, K. Speech recognition: A model and a program for research. (1962) Quoted in U. Neisser, Cognitive psychology. New York: Appleton-Century-Crofts, 1967.

Halliday, M. A. K. Learning how to mean. London: Edward Arnold, 1975.

Henry, R. B. Implementation of experimental procedures in the PACER system of on-line control. Behaviour Research Methods and Instrumentation, 1976, 8, 508-512.

Howells, T. H. The experimental development of colour-tone synesthesia. Journal of Experimental Psychology, 1944, 34, 87-103.

Johnson-Laird, P. N. Procedural semantics. Cognition, 1977, 5, 189-214.

Kaplan, H. L. Clock-driven FORTRAN task scheduling in a multi-programming environment. Behaviour Research Methods and Instrumentation, 1977, 9, 176-183.

Katz, J. J., & Fodor, J. A. The structure of a semantic theory. Language, 1963, 39, 170-210.

Kintsch, W. Learning, memory, and conceptual processes. New York: Wiley, 1970.

Kirk, R. E. Experimental design: Procedures for the behavioural sciences. New York: Brooks-Cole, 1968.

Lashley, K. I. The problem of serial order in behaviour. (1951) Quoted in N. Chomsky, Review of Skinner's Verbal behaviour. Language, 1959, 35, 26-58.

Lenneberg, E. H. A biological perspective of language. In E. H. Lenneberg (Ed.), New directions in the study of language. Cambridge, Mass.: M.I.T. Press, 1964.

Link, S. W. 4k laboratory FOCAL. Behaviour Research Methods and Instrumentation, 1975, 7, 137-142.

Marshall, A. J. An investigation of the relationship between rate of responding during acquisition and resistance to experimental extinction. Unpublished Masters thesis, University of Cape Town, 1975.

Millenson, J. R. An isomorphism between stimulus-response notation and information processing flow diagrams. Psychological Record, 1967, 17, 305-319.

Millenson, J. R. A programming language for on-line control of psychological experiments. Behavioural Science, 1971, 16, 248-256.

Miller, G. A., Galanter, E., & Pribram, K. H. Plans and the structure of behaviour. New York: Holt, Rinehart, & Winston, 1960.

Miller, G. A., & Chomsky, N. Finitary models of language users. In R. D. Luce, R. R. Bush, & E. Galanter (Eds.), Handbook of mathematical psychology (3 vols.). New York: Wiley, 1963.

Miller, W., & Ervin, S. M. The development of grammar in child language. In U. Bellugi & R. Brown (Eds.), The acquisition

of language. Monographs of the Society for Research in Child Development, 1964, whole no. 92.

Moeser, S. D. Iconic factors and language word order. Journal of Verbal Learning and Verbal Behaviour, 1975, 14, 43-55.

Moeser, S. D., & Bregman, A. S. The role of reference in the acquisition of a miniature artificial language. Journal of Verbal Learning and Verbal Behaviour, 1972, 11, 759-769.

Moeser, S. D., & Bregman, A. S. Imagery and language acquisition. Journal of Verbal Learning and Verbal Behaviour, 1973, 12, 91-98.

Moeser, S. D., & Olson, A. J. The role of reference in children's acquisition of a miniature artificial language. Journal of Verbal Learning and Verbal Behaviour, 1974, 17, 204-218.

Mowrer, O. H. Learning theory and the symbolic process. New York: Wiley, 1960.

Mullen, S. FOCLAB: A language for computer-controlled psychology research. Behaviour Research Methods and Instrumentation, 1976, 8, 229-232.

MacCorquodale, K. On Chomsky's review of Skinner's Verbal behaviour. Journal of the Experimental Analysis of Behaviour, 1970, 13, 83-99.

McLean, R. S. PSYCHOL: A computer language for experimentation. Behaviour Research Methods and Instrumentation, 1969, 1, 323-328.

MacNamara, J. Cognitive basis of language learning in infants. Psychological Review, 1972, 79, 1-13.

McNeill, D. The acquisition of language. New York: Harper & Row, 1970.

Neimark, E. D., & Estes, W. K. Stimulus sampling theory. San Francisco: Holden Day, 1967.

Neimark, E. D., & Santa, J. L. Thinking and concept attainment. Annual Review of Psychology, 1975, 26, 173-205.

Neisser, U. Cognitive psychology. New York: Appleton-Century-Crofts, 1967.

Nelson, K. Concept, word, and sentence: Interrelations in acquisition and development. Psychological Review, 1974, 81, 267-285.

Newell, A., & Simon, H. A. Human information processing. Englewood Cliffs: Prentice-Hall, 1972.

Orne, M. T. On the social psychology of the psychological experiment: With particular reference to demand characteristics and their implications. American Psychologist, 1962, 17, 776-783.

Oxtoby, R. M. The role of rhythmic factors in the perception of time. Unpublished Masters thesis, University of Cape Town, 1971.

Piaget, J., & Inhelder, B. The psychology of the child. London: Routledge & Kegan Paul, 1969. (Originally published in French, 1966)

Pinkus, A. L., & Gregg, L. W. APCOL: A programming system for computer-controlled psychological laboratories. Behaviour Research Methods and Instrumentation, 1973, 5, 165-172.

Pitz, G. F. Building a programming language for a small computer: Reinventing the wheel. Behaviour Research Methods and Instrumentation, 1975, 7, 42-46.

Polanyi, M. Life's irreducible structure. Science, 1968, 160, 1308-1312.

Pollio, H. R. The psychology of symbolic activity. New York: Addison-Wesley, 1974.

Polson, P. G. SCAT: Design criteria. Behaviour Research Methods and Instrumentation, 1973, 5, 241-244.

Putnam, H. The "innateness hypothesis" and explanatory models in linugistics. In J. R. Searle (Ed.), The philosophy of language. London: Oxford University Press, 1971.

Roscoe, J. T. Fundamental research statistics (2nd. ed.) New York: Holt, Rinehart, & Winston, 1975.

Rosenthal, R. On the social psychology of the psychological experiment: The experimenter's hypothesis as unintended determinant of experimental results. American Scientist, 1963, 51, 268-283.

Russell, B. An inquiry into meaning and truth. London: Allen & Unwin, 1940.

Schank, R. C. Conceptual dependency: A theory of natural language understanding. Cognitive Psychology, 1972, 3, 552-631.

Schlesinger, I. M. Production of utterances and language acquisition. In D. I. Slobin (Ed.), The ontogenesis of grammar. New York: Academic Press, 1971. (a)

Schlesinger, I. M. Learning grammar: From pivot to realisation rule. In R. Huxley & E. Ingram (Eds.), Language acquisition: Models and methods. New York: Academic Press, 1971. (b)

Schneider, W., & Scholz, K. W. Requirements for minicomputer operating systems for human experimentation. Behaviour Research Methods and Instrumentation, 1973, 5, 173-177.

Scholz, K. W. PROSS: A processing control programming language. Behaviour Research Methods and Instrumentation, 1973, 5, 245-247.

Searle, J. R. What is a speech act? In J. R. Searle (Ed.), The philosophy of language. London: Oxford University Press, 1971.

Segal, E. M., & Stacey, E. W. Rule-governed behaviour as a psychological process. American Psychologist, 1975, 30, 541-552.

Shannon, C. E., & Weaver, W. The mathematical theory of communication. Urbana: University of Illinois Press, 1949.

Sidowski, J. B. On-line instrumentation in Psychology: Dildo or the real thing? Behaviour Research Methods and Instrumentation, 1976, 8, 52-56.

Sinclair, H. Sensorimotor action patterns as a condition for the acquisition of syntax. In R. Huxley & E. Ingram (Eds.), Language acquisition: Models and methods. New York: Academic Press, 1971.

Sinclair-de Zwart, H. Language acquisition and cognitive development. In T. E. Moore (Ed.), Cognitive development and the acquisition of language. New York: Academic Press, 1973.

Skinner, B. F. Verbal Behaviour. New York: Appleton-Century-Crofts, 1957.

Skinner, B. F. Contingencies of reinforcement: A theoretical analysis. New York: Appleton-Century-Crofts, 1969.

Slobin, D. I. Seven questions about language development. In P. C. Dodwell (Ed.), New horizons in Psychology (Vol. 2). London: Penguin, 1972.

Slobin, D. I. Cognitive pre-requisites for the acquisition of grammar. In C. A. Ferguson & D. I. Slobin (Eds.), Studies of child language development. New York: Holt, Rinehart, & Winston, 1973.

Snapper, A. G. A new OS/8 SKED. Behaviour Research Methods and Instrumentation, 1975, 7, 233-238.

Spear, T. L., Overgard, D., & Christian, T. W. The CLIPR display terminal experiment system. Behaviour Research Methods and Instrumentation, 1975, 7, 107-112.

Suppes, P. Stimulus-response theory of finite automata. Journal of Mathematical Psychology, 1969, 6, 327-355.

Taylor, J. G. The behavioural basis of perception. New Haven: Yale University Press, 1962.

Taylor, J. G. The role of axioms in psychological theory. Bulletin of the British Psychological Society, 1968, 21, 221-227.

Trabasso, T., & Bower, G. Attention in learning. New York: Wiley, 1969.

Whitehurst, G. J., & Vasta, R. Is language acquired through imitation? Journal of Psycholinguistic Research, 1975, 4, 37-59.

Winer, B. J. Statistical principles in experimental design (2nd. ed.) New York: McGraw-Hill, 1971.

Winograd, T. Understanding natural language. New York: Academic Press, 1972.

Appendix

Appendix I

An example of the raw data for subject M.K. under condition 1 (no feedback).

Block Number	Class Errors	Order Errors	Word Errors	Response Latency	Copy Errors
1	1	3	8	976	0
2	3	3	6	1022	0
3	0	0	8	1063	0
4	4	7	4	807	0
5	0	0	8	792	0
6	1	1	6	792	1
7	1	2	2	858	0
8	0	0	3	737	0
9	1	2	4	603	3
10	0	0	5	688	0
11	2 (1)	2	3	893	0
12	0	1 (2)	3	912	0
13	0	0	4	645	0
14	0	0	1	654	0
15	0	0	4	632	0
16	0	1	4	682	0
17	0	0	4	534	0
18	0	0	3	633	0
19	0	1	3	731	0
20	0	0	1	692	0
21	0	0	1 (3)	685	0
22	0	0	0	532	0
23	0	0	0	487	0
Totals	13	23	85	17050	4
Average				741	

(1) The block of last class error is block 11. The average latency for blocks 1 to 11 is 839.

(2) The block of last order error is taken to be block 12. Two error-free blocks follow it, while the isolated order errors which occur later on have very low probabilities.

(3) The subject has taken 21 blocks to learn the language.

Appendix II

Probabilities of error scores under conditions of random responding.

(1) In a given trial, there are six words available to make up a three-word sentence. These words may be sampled without replacement, since the computer program disallows any sentence in which a word is used more than once.

(2) There are 20 possible combinations of 3 words sampled without replacement from a pool of 6 words.

(3) For each combination of 3 words, there are 6 permutations of their order, yielding a total of 120 permutations on any trial.

(4) The probabilities of error are as follows for a particular trial.

Number of Errors (n)	P of (n) class errors	P of (n) order errors	P of (n) word errors
0	0,60	0,0667	0,05
1	0,40	0,60	0,45
2	-	0,3333	0,45
3	-	-	0,05

(5) A block of trials was made up of four feedback trials. Using the above probabilities of error on one trial, the probabilities of various error scores may be calculated for four trials. These are as follows :

Appendix (continued)

Total number of errors (t)	P of (t) class errors	P of (t) order errors	P of (t) word errors
0	0,0256	0,00002	0,000001
1	0,1536	0,00071	0,00023
2	0,3456	0,01	0,00326
3	0,3456	0,06827	0,02433
4	0,1296	0,22856	0,09939
5	-	0,34133	0,22545
6	-	0,24988	0,29468
7	-	0,08889	0,09939
8	-	0,01235	0,02433
9	-	-	0,00326
10	-	-	0,00023
11	-	-	0,000001

(6) According to these probabilities, only zero class errors could occur by chance less than 5% of the time in a block of four feedback trials. Similarly, two or less order errors could occur less than 5% of the time.

(7) No account has been taken of conditional probabilities; for example, the occurrence of a class error entails the occurrence of at least one order, and one word meaning, error.

Appendix III

Summary data for each subject under each feedback condition.

Subj	TR	CE	OE	SE	Tot.T	Av.T	C	T-CE	T-OE	RL-CE	RL-OE	RL-SE	PG	PS	C/TR	O/TR	S/TR
Condition 1 (no feedback)																	
MM	9	11	20	26	262	595	0	9	9	639	639	639	10	6	1,22	2,22	2,89
DG	11	14	21	44	302	580	1	8	8	627	627	591	10	6	1,27	1,91	4,00
JG	12	22	50	57	477	852	2	11	12	897	875	875	10	6	1,83	4,17	4,75
LN	18	20	34	59	514	643	1	18	18	657	657	657	10	6	1,11	1,89	3,28
MK	21	13	23	85	682	741	1	11	12	839	923	654	10	6	0,62	1,10	4,05
RS	24	37	53	120	916	881	3	22	22	920	920	802	10	6	1,54	2,21	5,00
Condition 2 (syntactic feedback)																	
CP	5	6	11	19	187	669	1	5	5	692	692	692	10	6	1,20	2,20	3,80
RS	9	2	4	23	247	561	1	1	3	903	865	457	8	4	0,22	0,44	2,56
NB	13	8	13	76	408	680	2	6	6	730	730	657	10	6	0,62	1,00	5,85
CVD	18	9	20	82	480	600	11	8	10	709	647	527	10	1	0,50	1,11	4,56
YO	27	21	38	88	734	633	4	22	22	671	671	553	10	6	0,78	1,41	3,26
BD	27	21	35	106	1064	917	9	20	21	1018	865	715	10	6	0,78	1,30	3,93

Appendix III

Summary data for each subject under each feedback condition

Subj	TR	CE	OE	SE	Tot.T	Av.T	C	T-CE	T-OE	RL-CE	RL-OE	RL-SE	PG	PS	C/TR	O/TR	S/TR
Condition 3 (semantic feedback)																	
KG	7	12	27	26	271	752	0	6	6	883	883	634	10	6	1,71	3,86	3,71
PA	16	13	30	63	433	601	2	14	14	618	618	565	10	6	,81	1,88	3,94
GB	17	14	14	91	403	530	0	17	17	559	559	559	9	0	,82	,82	5,35
DB	25	28	50	75	697	645	8	13	13	745	745	582	10	4	1,12	2,00	3,00
KB	26	43	84	111	477	426	1	26	26	430	430	430	8	6	1,65	3,23	4,27
UB	30	37	68	95	651	509	6	25	30	551	390	390	10	2	1,23	2,27	3,17
Condition 4 (syntactic and semantic feedback)																	
MH	4	3	5	10	182	760	6	1	1	1258	1258	822	9	2	,75	1,25	2,50
SL	11	10	15	51	372	716	2	11	11	737	737	737	10	6	,91	1,36	4,64
LL	12	13	27	47	456	814	4	11	11	855	855	631	10	6	1,08	2,25	3,92
VJ	14	15	20	64	532	832	1	12	12	914	914	677	8	6	1,07	1,43	4,57
BF	14	23	40	68	521	814	3	14	14	874	874	874	9	4	1,64	2,86	4,86
PG	32	28	46	122	914	672	7	30	30	700	700	524	9	6	,88	1,44	3,81

Appendix IV

Analysis of variance summary table for copy errors under conditions of semantic and syntactic feedback.

Source	SS	df	MS	F ratio
A (semantic)	0,65	1	0,65	0,07
B (syntactic)	28,25	1	28,25	3,1
AB	8,21	1	8,21	0,9
Within	182,31	20	9,12	

Cell and treatment means (and cell standard deviations) for copy errors.

		Factor B (syntactic feedback)		
		B1 (absent)	B2 (present)	
Factor A (semantic feedback)	A1 (absent)	1,33 (1,03)	4,67 (4,32)	3,00
	A2 (present)	2,83 (3,37)	3,83 (2,32)	3,33
		2,08	4,25	

Appendix V

Post-test questionnaire for the subjects

LANGUAGE LEARNING

Subject: _____

(1) What is the meaning of the following words?

(a) ZAB _____ (d) WIX _____

(b) TUG _____ (e) JIK _____

(c) BUF _____ (f) LAT _____

(2) Which of the following sentences are grammatically correct?[1]

(a) JIK ZAB LAT
(b) BUF ZAB LAT
(c) JIK ZAB TUG
(d) WIX BUF TUG
(e) ZAB WIX BUF
(f) JIK BUF LAT
(g) TUG ZAB WIX
(h) LAT TUG WIX
(i) BUF TUG WIX
(j) BUF TUG ZAB

Thank you for your help.

(1) Sentences (a), (b), and (i) are correct.

Appendix of computer programs

Acquisition of a miniature language--documentation

The general experimental procedure, which the computer controlled was as follows:

1: Accept the parameters of the experiment, particularly the experimental condition.

2: Present the instructions.

3: Present a picture, provide the correct descriptions, and wait for the subject to copy it. (Modelling trial).

4: Present a picture, and wait for the subject to produce a description. (Feedback trial).

5: Note any errors in the description, and provide feedback to the subject about the errors, according to the experimental condition.

6: If a block of trials has elapsed, punch a record of the subject's performance. Check to see if there were any errors made; if there

were, return to step 3. If no errors, then go to step 7.

7: Halt the experiment.

The computer programs which were written to implement the above procedures were:

EXPT2: the FORTRAN main program.

TIMER: the Assembler subroutine to provide timing and I/O.

LAN1: a FORTRAN subroutine to check any description in the artificial language.

KEYIN: a FORTRAN subroutine to acquire the three key-presses made by the subject when describing a picture.

SQURE, DIAMO, TRIAN : three FORTRAN subroutines to draw a square, diamond, or triangle on the VDU.

VDU: a FORTRAN subroutine to provide the special functions available with the VDU.

Three other programs were written to aid the de-bugging of the subroutines:

TESTL: FORTRAN main program to exercise LAN1

TESTF: FORTRAN main program to exercise SQURE, DIAMO, and TRIAN

TESTV: FORTRAN main program to exercise VDU

EXPT 2

The pilot study program was called EXPT1. The documentation provided here is simply sufficient to describe the flow of information and control in the experiment. No details of the program are specified, since it was essentially a one-off item. The important aspects of on-line control have been described in the chapter on the practice of experimental control.

Definition of variables

KEY1(i), KEY2(i): for key i, the word which it represents, e.g. "ÞJIK". Initially set to "Þxxx".

KEY3(i): the grammatical class of key i; 1=number, 2=noun, 3=adjective.

KEY4(i): the meaning of key i within its grammatical class. e.g. 1=one, 2=two; 1=triangle, 2=square, 3=diamond; 1=fully shaded, 2=half shaded, 3=not shaded.

IFUN: VDU function; 1=clear screen, 7=buzzer, 8=place cursor at column IPOS1, 9=place cursor at line IPOS1.

NPICS: the number of pictures available for presentation.

IPICS(i,1), (i,2), (i,3): for picture i, the key number of each of the three words which describes it.

NUMB: the position in the sentence of the word which refers to number; =3

IFIG: the position in the sentenceof the word which refers to the geometrical shape; =2

ICOLR: the position in the sentence of the word which refers to the shading; =1

IPSE: time allowed for the subject to examine his feedback; =400 (4 seconds)

IPAC: time interval following a copy of a description; =100.

IWAIT: time allowed for the subject to type in a response; =1000 (10 seconds)

NTR: number of trials in a block; =8. Defines the interval over which performance data is accumulated.

NBL: number of error-free blocks to reach criterion; =2

IEXP: experimental condition; 1=no feedback, 2=syntactic feedback, 3=semantic feedback, 4=full feedback.

ISLID: the current picture number being presented; ISLID=1, NPICS

IN: environmental input.

IOUT: output to the reed relays.

ISW: state of the computer's switch register.

NB: counter of the number of blocks of trials presented.

ISEM, ISYNO, ISYNM: number of errors made in a block on word meaning, order, or class.

NERR: number of times a model was miscopied.

ITYPE: =0 for a modelling trial, =1 for a feedback trial.

IRT: response latency of subject on a feedback trial, summed within a block of trials.

NT: counter of the number of trials within a block.

IPARA(i): a general-purpose array usually used to assemble the feedback to the subject.

INPUT(i): the keys pressed by the subject; i=1,3.

Information and control flowchart

1: Allocate storage and initialise variables.

2: Clear VDU screen.

3: Request and read the syntax of the language.

4: Read the paper tape which defines the names and meanings of the keys on the keyboard.

5: Read the paper tape which defines the list of pictures to be presented. (These tapes have been prepared off-line.)

6: State the number of pictures in the list.

7: Establish the various parameters of the experiment: the waiting times, number of trials per block, the termination criterion, and the experimental criterion.

8: Clear the VDU screen.

9: Request the name of the subject, and punch a tape with this as part of the header information.

10: Wait for a key press.

11: Present a page of intructions.

12: If this was the last page, go to 13; else return to 10.

13: Wait for a key press.

14: Start the TIMER subroutine.

15: Initialise experimental variables.

16: Begin a block of trials. Increment the block counter, and set performance data to zero for this block.

17: Present each trial in a block:

18: Increment the picture number counter; if the maximum number is reached (the last picture in the list has been shown), reset to 1.

19: Compute the details of the picture: the geometric shape involved, the shading, and the number of shapes.

20: Clear the VDU screen.

21: For each shape, call the appropriate subroutine to draw it on the screen. Compute its position and shading.

22: If this is a modelling trial, goto 23; else go to 30.

23: Compute the correct description of the picture.

24: Display this correct description, and instruct the subject to copy.

25: Call KEYIN to accept the description.

26: Check the copy against the original.

27: If there was any error, goto 28; else go to 29.

28: Increment the counter of copy errors by 1, inform the subject of the error, and go to 25.

29: Wait for a second or so, then return to step 18.

30: Request the subject to describe the picture.

31: Set ITIME to zero.

32: Call KEYIN to accept the description.

33: Capture the response latency upon return from KEYIN.

34: Set the correct description of the picture, and call LAN1 to compare the correct and given descriptions.

35: Accumulate the errors noted by LAN1.

36: Set up the feedback to the subject. LAN1 has already corrected some of the input, according to the experimental condition.

37: If there were any errors at all, goto 38; else give a congratulatory message, and go to 39.

38: Provide the feedback according to the experimental condition:

1 - give no feedback.

2 - check if there was any error of word order. If there was, provide feedback and the description corrected for word order; if there was not an error of word order, provide a warning message.

3 - check if there was any error of word meaning. If there was, provide feedback and the description corrected for word meaning; if there was not an error of semantics, provide a warning message.

4 - provide the correct description.

39: Wait for a few seconds.

40: Return to 17 and present another trial; if the block of trials is completed, however, go to 41.

41: Punch a record of the subject's performance for the block.

42: Check if the language has been learned to criterion. If it has not, return to 16; otherwise terminate the experiment.

TIMER

This subroutine follows essentially the same procedures described in the chapter on the practice of computer control. Some additional instructions in the interrupt handler are needed to assemble the multiple input from the keyboard into a single word, while provision is made for the timing interval (INTER) to be specified by the main program.

LAN1

The subroutine examines a description of a picture, and counts the number of word class, word order, and word meaning errors. According to the experimental condition, the feedback sentence to the subject is then assembled.

The routine assumes that words of class 1 preceed words of class 2, and that words of class 3 come last in the sentence. First the number of missing word classes is counted (IM). Then, the order of words is checked; if a word of a given class is not in its correct position, the number of word order errors (IO) is incremented by one.

Due to the restrictions in the language, one word out of order necessarily implies that at least one other word is out of order; thus the count of word order errors is reduced from either 2 or 3 by one to either 1 or 2, if non-zero. Finally, the input description is searched for each of the words which ought to be present; for each missing word, the count of word meaning errors (IW) is incremented by one.

The feedback is then partially assembled. For experimental condition 1, the feedback array is set equal to the input sentence (though it is never used; this is to hold the response time of the program to roughly similar limits). For condition 2, the words in the input are re-arranged to fall into their syntactically correct order. For condition 3, words which have the wrong meaning are replaced by words with the right meaning. (In each of these feedback conditions, there may be a missing class of word in the input; in this case, the main program sets the missing word to 'XXX' in the feedback.) For condition 4, the feedback is simply the correct sentence.

KEYIN

The subroutine accepts and checks each key-press of the subject. If an undefined key was pressed, or if a key was pressed more than once, an error message was placed on the VDU screen, and the subject was requested to try again.

If three key presses had not been recorded by a certain time, a buzzer was sounded continuously until a full sentence was received.

For each legal key-press, the "name" of the key, a word in the language, was written on the VDU screen. After three legal inouts, control was returned to the main program.

SQURE, DIAMO, TRIAN

These three subroutines simply drew a square, diamond, or triangle on the VDU screen, at a given position, with a specified amount of shading. The size of the figures was arranged such that each covered a comparable area of the screen.

VDU

A simple routine to provide control of the VDU special functions.

TESTL

The program was written to exercise the LAN1 and TIMER sub-routines, using the keyboard of the VDU itself. Certain of the keys could be defined by the operator in a manner identical to that required by the main program EXPT2, and some pictures could be similarly defined. Then, any input sentence was passed to LAN1 for analysis, and the response latency was taken and printed. The feedback constructed by LAN1 was printed; this also served to verify portions of the main EXPT2 program.

TESTF

The program was used to call the drawing subroutines SQURE, TRIAN, and DIAMO, and to check their output.

TESTV

The program exercised the special functions of the VDU.

```
FTN,B
      PROGRAM EXPT2
      DIMENSION INPUT(3),INTYP(3),IPARA(4),IFEED(3)
     DIMENSION KEY1(10),KEY2(10),KEY3(10),KEY4(10)
   DIMENSION IPICS(64,3),ISNAM(10),LINE(36)
      COMMON ITIME,INTER,IN,IOUT,ISW
      INTER=2
      IPRIN=2
      IPUN=4
      IREAD=5
      KEYBD=1
      IVDU=IPRIN
      MAXK=10
      MAXP=64
      IX1=20130B
      IX2=54130B
      DO 150 I=1,MAXK
      KEY1(I)=IX1
      KEY2(I)=IX2
      KEY3(I)=0
      KEY4(I)=0
150   CONTINUE
      IFUN=1
      IPOS1=0
      CALL VDU(IFUN,IPOS1)
      WRITE(IPRIN,42)
42    FORMAT(" POSN OF WORDS FOR NO, SHAPE, SHADE: ~")
      READ(KEYBD,*)NUMB,IFIG,ICOLR
C
      READ(IREAD,12)N
12    FORMAT(I2,2A2,2I1)
```

```
      DO 100 I=1,N
      READ(IREAD,12)KEYNO,(IPARA(J),J=1,4)
      IF(KEYNO)110,110,120
120   IF(KEYNO-MAXK)130,130,110
110   WRITE(IPRIN,13)
13    FORMAT(" *** ERROR ON INPUT ***")
      STOP
130   KEY1(KEYNO)=IPARA(1)
      KEY2(KEYNO)=IPARA(2)
      KEY3(KEYNO)=IPARA(3)
      KEY4(KEYNO)=IPARA(4)
100   CONTINUE
C
      READ(IREAD,15)NPICS
15    FORMAT(4I2)
      DO 200 I=1,NPICS
      READ(IREAD,15)IPICN,(IPARA(J),J=1,3)
      IF(IPICN)210,210,220
220   IF(IPICN-MAXP)230,230,210
210   WRITE(IPRIN,13)
      STOP
230   DO 240 J=1,3
      K=IPARA(J)
      IF(KEY3(K)-J)250,240,250
250   WRITE(IPRIN,13)
240   IPICS(IPICN,J)=IPARA(J)
200   CONTINUE
C
      WRITE(IPRIN,16)NPICS
16    FORMAT(//,I4," PICTURES")
WRITE(IPRIN,41)
41    FORMAT(" PAUSE AFTER FEEDBACK, & AFTER COPY: ←")
READ(KEYBD,*)IPSE,IPAC
     WRITE(IPRIN,43)
43    FORMAT(" WAIT BEFORE BUZZ: ←")
      READ(KEYBD,*)IWAIT
      WRITE(IPRIN,17)
17    FORMAT(" TRIALS PER BLOCK: ←")
      READ (KEYBD,*)NTR
270   WRITE(IPRIN,18)
18    FORMAT(" NO OF ERROR-FREE BLOCKS TO END EXPT: ←")
      READ(KEYBD,*)NBL
C
300   WRITE(IPRIN,19)
19    FORMAT(///," CONDITION: ←")
      READ(KEYBD,*)IEXP
      IF(IEXP)310,310,320
320   IF(IEXP-4)330,330,310
310   WRITE(IPRIN,13)
      GOTO 300
330   IFUN=1
```

```
      IPOS1=0
      CALL VDU(IFUN,IPOS1)
      WRITE(IPRIN,20)
20    FORMAT(" SUBJECT NAME: ←")
      READ(KEYBD,21)(ISNAM(J),J=1,10)
21    FORMAT(36A2)
      NB=0
      WRITE(IPUN,39)(ISNAM(J),J=1,10),IEXP,NTR
39    FORMAT(10A2,2I3)
      WRITE(IPRIN,22)
22    FORMAT(/," READY FOR INSTRUCTIONS AND EXAMPLES",/,
     1 " PRESS ANY KEY TO ADVANCE TO THE NEXT PAGE")
      CALL TIMER
      GOTO 471
440   READ(IREAD,15)I
      IFUN=1
      IPOS1=0
      CALL VDU(IFUN,IPOS1)
      IF(I)450,450,460
460   DO 470 K=1,I
      READ(IREAD,21)(LINE(N),N=1,36)
      WRITE(IVDU,21)(LINE(N),N=1,36)
470   CONTINUE
      ITIME=0
471   CONTINUE
      IF(IN)471,471,440
C
450   CALL TIMER
      ISLID=0
      IN=0
      IOUT=0
      ISW=0
      IFE=NBL
400   NB=NB+1
      ISEM=0
      ISYNM=0
      ISYNO=0
      ITYPE=0
      IRT=0
      NERR=0
C
      DO 500 NT=1,NTR
      ISLID=ISLID+1
      IF(ISLID-NPICS)512,512,514
514   ISLID=1
512   INO=IPICS(ISLID,NUMB)
      IFT=IPICS(ISLID,IFIG)
      ISHAD=IPICS(ISLID,ICOLR)
      INO=KEY4(INO)
      IFT=KEY4(IFT)
```

```
      ISHAD=KEY4(ISHAD)
      IFUN=1
      IPOS1=0
      CALL VDU(IFUN,IPOS1)
      DO 505 I=1,INO
      IFUN=9
      IPOS1=1
      CALL VDU(IFUN,IPOS1)
      IFUN=8
      IPOS1=(I-1)*26+1
      CALL VDU(IFUN,IPOS1)
      GOTO(501,502,503),IFT
501   CALL TRIAN(ISHAD,IPOS1)
      GOTO 505
502   CALL SQURE(ISHAD,IPOS1)
      GOTO 505
503   CALL DIAMO(ISHAD,IPOS1)
505   CONTINUE
      WRITE(IPRIN,38)NB,NT
38    FORMAT(/,2I3," ←")
      IF(ITYPE)510,510,600
510   K=1
      ITYPE=1
      DO 520 J=1,3
      I=IPICS(ISLID,J)
      IPARA(K)=KEY1(I)
      K=K+1
      IPARA(K)=KEY2(I)
      K=K+1
520   CONTINUE
      WRITE(IVDU,23)(IPARA(J),J=1,6)
23    FORMAT("PLEASE COPY THIS DESCRIPTION: ",
     1 6A2," ←")
560   CALL KEYIN(INPUT,KEY1,KEY2,KEY3,IWAIT)
      IERR=0
      DO 530 J=1,3
      IF(INPUT(J)-IPICS(ISLID,J))540,530,540
540   IERR=IERR+1
530   CONTINUE
       IF(IERR)570,570,550
550   WRITE(IVDU,25)
25    FORMAT(" YOUR COPY WAS IN ERROR, TRY AGAIN ←")
      NERR=NERR+1
      GOTO 560
570   WRITE(IPRIN,34)
      ITIME=0
580   CONTINUE
      IF(ITIME-IPAC)580,580,500
C
600   ITYPE=0
      WRITE(IVDU,27)
27    FORMAT(" PLEASE DESCRIBE THE PICTURE ←")
```

```
      ITIME=0
      CALL KEYIN(INPUT,KEY1,KEY2,KEY3,IWAIT)
      IR=ITIME
      IRT=IRT+IR
      DO 610 J=1,3
      IPARA(J)=IPICS(ISLID,J)
      I=INPUT(J)
610   INTYP(J)=KEY3(I)
C
      CALL LAN1(INPUT,INTYP,IPARA,IEXP,IFEED,IM,IO,IW)
C
      ISEM=ISEM+IW
      ISYNM=ISYNM+IM
      ISYNO=ISYNO+IO
      K=1
      DO 620 J=1,3
      I=IFEED(J)
      IF(I)622,622,625
622   IPARA(K)=IX1
      K=K+1
      IPARA(K)=IX2
      GOTO 620
625   IPARA(K)=KEY1(I)
      K=K+1
      IPARA(K)=KEY2(I)
620   K=K+1
C
      K=IW+IO+IM
      IF(K)634,634,655
655   GOTO(650,651,652,643),IEXP
651   IF(IO)632,632,641
652   IF(IW)633,633,642
632   WRITE(IPRIN,28)
      GOTO 650
633   WRITE(IPRIN,29)
     GOTO 650
634   WRITE(IPRIN,30)
      GOTO 650
28    FORMAT(" GOOD WORD ORDER, BUT NOT COMPLETELY CORRECT")
29    FORMAT(" GOOD WORD USAGE, BUT NOT COMPLETELY CORRECT")
30    FORMAT(" GOOD DESCRIPTION, COMPLETELY CORRECT")
C
641   WRITE(IPRIN,31)(IPARA(J),J=1,6)
      GOTO 650
642   WRITE(IPRIN,32)(IPARA(J),J=1,6)
     GOTO 650
643   WRITE(IPRIN,33)(IPARA(J),J=1,6)
      GOTO 650
31    FORMAT(" WORD ORDER SHOULD BE: ",6A2)
32    FORMAT(" WORD USAGE SHOULD BE: ",6A2)
33    FORMAT(" CORRECT DESCRIPTION SHOULD BE: ",6A2)
C
650   WRITE(IPRIN,34)
3
```

```
 34     FORMAT("←")
        ITIME=0
590     CONTINUE
        IF(ITIME-IPSE)590,590,500
C
500     CONTINUE
C
        IR=IRT*2/NTR
        WRITE(IPUN,40)NB,ISYNM,ISYNO,ISEM,IR,NERR
40      FORMAT(6I5)
       IF(ISW)430,430,402
430     I=ISYNM+ISYNO+ISEM
        IF(I)410,410,420
420     IFE=NBL
        GOTO 400
410     IFE=IFE-1
        IF(IFE)402,402,400
C
402     IFUN=1
        IPOS1=0
        CALL VDU(IFUN,IPOS1)
        WRITE(IVDU,35)
35      FORMAT(//," THIS SERIES OF TRIALS IS OVER",/," THANK YOU")
        PAUSE
        GOTO 450
        END
        END$
```

```
ASMB,R,B
 NAM TIMER
 ENT TIMER
 COM TIME,INTER,IN,OUT,SW
*
TBG EQU 10B
IREG EQU 14B
OREG EQU 15B
*
TIMER NOP
 ISZ TIMER
 LDA INTER
 OTA TBG
 LDA IJSB
 STA TBG
 STC TBG,C
 JMP TIMER,I
*
 ORB
IJSB JSB LINK,I
LINK DEF CONT
 ORR
*
CONT NOP
STC TBG,C
 STA SAVEA
 LDA OUT
OTA OREG
 STB SAVEB
 LIA 14B
 AND MASK
 CLB
ALF,ALF
 ALF
 SLA
 INB
 ARS
 ADA 1B
 STA IN
LDB SAVEB
 LIA 1B
 STA SW
 LDA SAVEA
 ISZ TIME
 NOP
 JMP CONT,I
*
TEMP BSS 1
MASK OCT 720
SAVEB BSS 1
SAVEA BSS 1
 END
```

```
FTN,B
SUBROUTINE LAN1(INPUT,INTYP,IPARA,IEXP,IFEED,IM,IO,IW)
      DIMENSION INPUT(3),INTYP(3),IPARA(3),IFEED(3),ICK(3),IPOS(3)
C
C   ASSUME THE CORRECT WORD TYPE ORDER IS 1, 2, 3.
C   CHECK FOR ERRORS, THEN CORRECT ACCORDING TO IEXP
C
C   CHECK FOR CLASS PRESENCE
C   IM=NO OF MISSING WORD TYPES
C
      IM=0
      DO 110 I=1,3
110   ICK(I)=1
      DO 100 I=1,3
      J=INTYP(I)
      IF(J)100,100,105
105   ICK(J)=0
100   CONTINUE
      DO 120 I=1,3
120   IM=IM+ICK(I)
C
C   CHECK FOR ORDER OF CLASSES
C   IO=NO OF DISORDERED WORD TYPES
C
IO=0
DO 130 I=1,3
      IF(INTYP(I)-I)140,130,140
140   IO=IO+1
130   CONTINUE
      IF(IO)150,150,160
160   IF(IM)165,165,155
155   IF(IO-3)150,165,165
165   IO=IO-1
C
C   COUNT THE WRONG WORDS
C   IW=NO OF WRONG WORDS
C
150   IW=0
    DO 170 I=1,3
170   IPOS(I)=IPARA(I)
      DO 180 I=1,3
      DO 180 J=1,3
      IF(IPOS(J)-INPUT(I))180,185,180
185   IPOS(J)=0
180   CONTINUE
      DO 190 I=1,3
      IF(IPOS(I))190,190,195
195   IW=IW+1
190   CONTINUE
C
      GOTO (200,300,400,500),IEXP
C
```

```
C
C   200 - NO CORRECTION OF INPUT
C   300 - SYNTACTIC CORRECTION
C   400 - SEMANTIC CORRECTION
C   500 - CORRECT MESSAGE PROVIDED
C
200     DO 210 I=1,3
210     IFEED(I)=INPUT(I)
        RETURN
C
500     DO 510 I=1,3
510     IFEED(I)=IPARA(I)
        RETURN
C
300     DO 310 I=1,3
        IFEED(I)=0
        DO 320 J=1,3
        IF(INTYP(J)-I)320,330,320
330     IF(IFEED(I))340,340,320
340     IFEED(I)=INPUT(J)
320     CONTINUE
310     CONTINUE
        RETURN
C
400     DO 410 I=1,3
        IFEED(I)=0
410     ICK(I)=0
        DO 420 I=1,3
        J=INTYP(I)
        IF(J)420,420,415
415     IF(ICK(J))430,430,420
430     ICK(J)=1
        IFEED(I)=IPARA(J)
420     CONTINUE
        RETURN
        END
        ENDS
```

```
FTN,B
      SUBROUTINE KEYIN(INPUT,KEY1,KEY2,KEY3,IWAIT)
      DIMENSION INPUT(3),KEY1(10),KEY2(10),KEY3(10),IT(10)
     COMMON ITIME,INTER,IN,IOUT,ISW
      IVDU=2
      ITIME=0
      WRITE(IVDU,10)
10    FORMAT(": ←")
C
910   DO 100 I=1,3
100   INPUT(I)=0
      DO 101 I=1,10
101   IT(I)=0
      I=1
C
200   CONTINUE
      IF(ITIME-IWAIT)270,270,280
280   IFUN=7
      IPOS1=0
      CALL VDU(IFUN,IPOS1)
270   IF(IN)200,200,210
210   INN=IN
      IF(INN-10)240,240,900
240   IF(IT(INN))920,205,920
205   IT(INN)=1
      IF(KEY3(INN))900,900,250
250   WRITE(IVDU,11)KEY1(INN),KEY2(INN)
11    FORMAT(2A2,1X,"←")
      INPUT(I)=INN
230   CONTINUE
      IF(IN)220,220,230
220   IF(I-3)260,300,300
260   I=I+1
      GOTO 200
C
300   WRITE(IVDU,13)
13    FORMAT(/)
      RETURN
C
920   WRITE(IVDU,14)
14    FORMAT(/," PLEASE DO NOT USE A WORD MORE THAN ONCE",
    1 " IN A SENTENCE ←")
900   WRITE(IVDU,12)
12    FORMAT(/," SORRY - THERE WAS AN ERROR IN YOUR INPUT",/,
     1 " PLEASE TRY AGAIN: ←")
      GOTO 910
      END
      END$
```

```
FTN,B
      SUBROUTINE SQURE(ISHAD,IPOSN)
       DIMENSION IBUF(28)
       IBLAN=40B
       IX=130B
       IVDU=2
       IBUF(1)=20B
       IBUF(2)=IPOSN
       DO 100 I=3,28
100    IBUF(I)=IBLAN
       DO 110 I=1,4
110    WRITE(IVDU,10)(IBUF(J),J=1,28)
10     FORMAT(28A1)
       DO 200 I=8,23
200    IBUF(I)=IX
       GOTO(210,220,230),ISHAD
210    DO 300 I=1,9
300    WRITE(IVDU,10)(IBUF(J),J=1,28)
       GOTO 900
220    WRITE(IVDU,10)(IBUF(J),J=1,28)
       DO 310 J=1,4
       DO 320 I=9,22
320    IBUF(I)=IBLAN
       WRITE(IVDU,10)(IBUF(I),I=1,28)
       DO 330 I=9,22
330    IBUF(I)=IX
       WRITE(IVDU,10)(IBUF(I),I=1,28)
310    CONTINUE
       GOTO 900
230    WRITE(IVDU,10)(IBUF(I),I=1,28)
       DO 350 I=9,22
350    IBUF(I)=IBLAN
       DO 340 J=1,7
340    WRITE(IVDU,10)(IBUF(I),I=1,28)
       DO 360 I=9,22
360    IBUF(I)=IX
       WRITE(IVDU,10)(IBUF(I),I=1,28)
900    DO 910 I=3,28
910    IBUF(I)=IBLAN
       DO 920 J=1,4
920    WRITE(IVDU,10)(IBUF(I),I=1,28)
       RETURN
       END
      ENDS
```

```
FTN,B
      SUBROUTINE DIAMO(ISHAD,IPOSN)
      DIMENSION IBUF(28)
      IBLAN=40B
      IX=130B
      IVDU=2
      IBUF(1)=20B
      IBUF(2)=IPOSN
      DO 100 I=3,28
100   IBUF(I)=IBLAN
      IBUF(15)=IX
      WRITE(IVDU,10)(IBUF(I),I=1,28)
10    FORMAT(28A1)
      K1=14
      K2=16
      ISW=1
      DO 230 K=1,2
      INCR=K-1
      IF(INCR)210,210,220
210   INCR=INCR-1
220   DO 200 J=1,8
      L1=K1-INCR
      L2=K2+INCR
      IBUF(L1)=IBLAN
      IBUF(L2)=IBLAN
      GOTO (300,310,320),ISHAD
300   DO 400 I=K1,K2
400   IBUF(I)=IX
      GOTO 320
310   IF(ISW)410,410,420
410   ISW=1
      GOTO 300
420   ISW=0
      DO 430 I=K1,K2
430   IBUF(I)=IBLAN
320   IBUF(K1)=IX
      IBUF(K2)=IX
      WRITE(IVDU,10)(IBUF(I),I=1,28)
      K1=K1+INCR
      K2=K2-INCR
200   CONTINUE
      K1=K1-INCR-INCR
      K2=K2+INCR+INCR
230   CONTINUE
      RETURN
      END
      END$
```

```
FTN,B
      SUBROUTINE TRIAN(ISHAD,IPOSN)
      DIMENSION IBUF(28)
      IBLAN=40B
      IX=130B
      IVDU=2
      IBUF(1)=20B
      IBUF(2)=IPOSN
      DO 100 I=3,28
100   IBUF(I)=IBLAN
      DO 110 J=1,2
110   WRITE(IVDU,10)(IBUF(I),I=1,28)
10    FORMAT(28A1)
      IBUF(15)=IX
      WRITE(IVDU,10)(IBUF(I),I=1,28)
      K1=14
      K2=16
      ISW=0
      DO 200 J=1,10
      L1=K1+1
      L2=K2-1
      IBUF(L1)=IBLAN
      IBUF(L2)=IBLAN
      IBUF(K1)=IX
      IBUF(K2)=IX
      GOTO (300,310,320),ISHAD
300   DO 400 I=L1,L2
400   IBUF(I)=IX
      GOTO 320
310   IF(ISW)410,410,420
410   ISW=1
      GOTO 300
420   ISW=0
      DO 430 I=L1,L2
430   IBUF(I)=IBLAN
320   WRITE(IVDU,10)(IBUF(I),I=1,28)
      K1=K1-1
      K2=K2+1
200   CONTINUE
      DO 900 I=K1,K2
900   IBUF(I)=IX
      WRITE(IVDU,10)(IBUF(I),I=1,28)
      DO 910 I=3,28
910   IBUF(I)=IBLAN
      DO 920 J=1,3
920   WRITE(IVDU,10)(IBUF(I),I=1,28)
      RETURN
      END
```

```
FTN,B
      SUBROUTINE VDU(IFUN,IPOS1)
      IF(IFUN)900,900,100
100   IF(IFUN - 10)110,900,900
110   GOTO (1,2,3,4,5,6,7,8,9),IFUN
C
1     I=14B
      GOTO 200
2     I=1B
      GOTO 200
3     I=32B
      GOTO 200
4     I=12B
      GOTO 200
5     I=6B
      GOTO 200
6     I=25B
      GOTO 200
7     I=7B
      GOTO 200
8     I=20B
      J=IPOS1/10
      J1=J*10
      J2=IPOS1-J1
      IPOS1=J*16+J2
      GOTO 300
9     I=13B
      GOTO 300
C
200   WRITE(2,10)I
10    FORMAT(A1,"←")
      RETURN
300   WRITE(2,11)I,IPOS1
11    FORMAT(2A1,"←")
900   RETURN
      END
      END$
```

```
FTN,B
      PROGRAM TESTL
      DIMENSION INPUT(3),INTYP(3),IPARA(3),IFEED(3)
      DIMENSION KEYS(10,3),IPICS(10,3)
      COMMON ITIME,INTER,IN,IOUT,ISW
      IPRIN=2
      KEYBD=1
      INTER=1
      WRITE(IPRIN,10)
10    FORMAT(///," TEST FOR LANG",/)
      WRITE(IPRIN,11)
11    FORMAT(" GIVE LABEL & CLASS PER KEY")
      DO 100 I=1,10
      WRITE(IPRIN,12)I
12    FORMAT(I3," ←")
      READ(KEYBD,13)(KEYS(I,J),J=1,3)
13    FORMAT(2A2,I1)
100   CONTINUE
C
      WRITE(IPRIN,14)
14    FORMAT(//," GIVE KEYS PER PICTURE")
      DO 110 I=1,10
      WRITE(IPRIN,12)I
      READ(KEYBD,15)(IPICS(I,J),J=1,3)
15    FORMAT(3I2)
110   CONTINUE
C
      WRITE(IPRIN,16)
16    FORMAT(//," TO A ?, GIVE EXPT CONDITION & PICTURE NO")
      WRITE(IPRIN,17)
17    FORMAT(" THEN GIVE INPUT")
      CALL TIMER
C
200   WRITE(IPRIN,18)
18    FORMAT("? ←")
      READ(KEYBD,*)IEXP,ISLID
      READ(KEYBD,*)(INPUT(I),I=1,3)
      DO 210 I=1,3
      IPARA(I)=IPICS(ISLID,I)
      J=INPUT(I)
210   INTYP(I)=KEYS(J,3)
     ITIME=0
C
      CALL LAN1(INPUT,INTYP,IPARA,IEXP,IFEED,IM,IO,IW)
C
      IT=ITIME
      K=1
      DO 620 J=1,3
      I=IFEED(J)
      IF(I)622,622,625
622   IPARA(K)=20130B
      K=K+1
      IPARA(K)=54130B
```

```
      GOTO 620
625   IPARA(K)=KEYS(I,1)
     K=K+1
      IPARA(K)=KEYS(I,2)
620   K=K+1
C
      GOTO (650,651,652,653),IEXP
651   IF(IO)632,632,641
652   IF(IW)633,633,642
653   IF(IO)654,654,643
654   IF(IW)634,634,643
632   WRITE(IPRIN,28)(IPARA(J),J=1,6)
      GOTO 500
633   WRITE(IPRIN,29)(IPARA(J),J=1,6)
      GOTO 500
634   WRITE(IPRIN,30)(IPARA(J),J=1,6)
      GOTO 500
28    FORMAT(" GOOD WORD ORDER: ",6A2)
29    FORMAT(" GOOD WORD USAGE: ",6A2)
30    FORMAT(" GOOD DESCRIPTION: ",6A2)
C
641   WRITE(IPRIN,31)(IPARA(J),J=1,6)
      GOTO 500
642   WRITE(IPRIN,32)(IPARA(J),J=1,6)
      GOTO 500
643   WRITE(IPRIN,33)(IPARA(J),J=1,6)
      GOTO 500
31    FORMAT(" WORD ORDER SHOULD BE: ",6A2)
32    FORMAT(" WORD USAGE SHOULD BE: ",6A2)
33    FORMAT(" CORRECT DESCRIPTION SHOULD BE: ",6A2)
C
650   WRITE(IPRIN,34)(IPARA(J),J=1,6)
34    FORMAT(" YOUR DESCRIPTION WAS: ",6A2)
C
500   WRITE(IPRIN,35)IM,IO,IW,IT
35    FORMAT(" IM IO IW TIME",/,4I3)
      GOTO 200
      END
      END$
```

```
FTN,B
      PROGRAM TESTF
      DIMENSION IVAL(3)
      IVDU=2
      KEYBD=1
      IVAL(1)=1
      IVAL(2)=27
      IVAL(3)=53
      DO 100 I=1,3
      J=IVAL(I)/10
       J1=J1*10
      J2=IVAL(I)-J1
      J=J*16+J2
      IVAL(I)=J
100   CONTINUE
110   WRITE(IVDU,10)
10    FORMAT(" TYPE OF FIGURE: 1=TRIANGLE, 2=SQUARE, 3=DIAMOND ?←")
      READ(KEYBD,*)ITYPE
      WRITE(IVDU,11)
11    FORMAT(" SHADING: 1=FULL, 2=HALF, 3=NONE ?←")
      READ(KEYBD,*)ISHAD
      WRITE(IVDU,12)
12    FORMAT(" NUMBER OF FIGURES ?←")
      READ(KEYBD,*)INO
      IPOSN=14B
      WRITE(IVDU,13)IPOSN
13    FORMAT(A1)
      DO 200 I=1,INO
      IPOSN=13B
      J=1
      WRITE(IVDU,14)IPOSN,J
14    FORMAT(2A1)
      IPOSN=IVAL(I)
      GOTO (210,220,230),ITYPE
210   CALL TRIAN(ISHAD,IPOSN)
      GOTO 200
220   CALL SQURE(ISHAD,IPOSN)
      GOTO 200
230   CALL DIAMO(ISHAD,IPOSN)
200   CONTINUE
      GOTO 110
      END
      END$
```

```
FTN,B
      PROGRAM TESTV
     DIMENSION MESS(80)
100   WRITE (2,10)
10    FORMAT(" MESSAGE(0) OR GIVE FUNCTION NO(I) : ←")
     READ(1,*)I
      IF(I)200,200,300
200   READ(1,11)(MESS(J),J=1,80)
11    FORMAT(80A1)
      GOTO 100
300   IF(I-8)400,500,500
400   J=I

     CALL VDU(I,J  )
      GOTO 100
500   WRITE(2,12)
12    FORMAT(" POSITION : ←")
      READ(1,*)J

     CALL VDU(I,J  )
      GOTO 100
      END
      END$
```

Programs from Abromovitz (1972)

The experiment involved the conditioning of a key-press, Left or Right, to a colour stimulus, Red or Green respectively. Some eighteen degrees of colour saturation were used, ranging from fully saturated red through light red through neutral (grey) to light green and saturated green. A tone, High or Low, was paired with Red and Green respectively; in an experimental trial, the tone would come on for a certain time, then the colour would be presented, and the subject would have to respond with the correct key-press as quickly as possible. The reaction time was monitored, and the subject was given feedback about his responses on a variable ratio schedule. Errors were indicated by a buzzer, while correct responses

were indicated by a set of chimes. Every few trials, the usual tone-colour combination was reversed, leading to an error in a well-conditioned subject.

The program was required to present all the various stimuli, monitor the responses, and provide the feedback. A number of different time intervals were involved, and the schedules of reinforcement could be varied independently. Provision was made to calculate the pay due to a subject, given the value of a correct response and the penalty of a wrong response. The number of trials in a block, and the number of blocks in an experimental session, were specifiable. Early and late responses required apecial attention. Comprehensive performance data was accumulated for each block of trials.

Since this was one of the earliest control programs, the techniques employed were somewhat crude. In particular, most of the experimental I/O was carried out by an Assembler subroutine OUTK. Two other minor subroutines, WAIT and LIGHT, were used in the way their names suggest; the first initiated a pause between experimental blocks and then gave a warning signal at the start of the next block, while the second turned on a light in the experimental room, which was usually completely dark.

```
FTN,B
      PROGRAM PHDH
      DIMENSION  IEARL(18,2),IRONG(18,2),ILATR(18,2),SUM(18,2)
      DIMENSION  SQS(18,2),IRAN(600),IREV(200),IDATE(10),NAME(6)
      DIMENSION  IEAR(5,2),IRON(5,2),ILAT(5,2),SUMS(5,2),SQSS(5,2)
      DIMENSION  NUMBS(18,2),NUMB(5,2)
      COMMON  ITIMT,ITIMC,ITIMF,ITIMP,ITIMS,IER1,IER2,IER3,
     1 ISTIM,IFREV,IOPS,IFENN,IFENR,IFEPN,IFEPR,IFERF,
     2 ILATE,NERNN,NERNR,NERPN,NERPR,ITIMW
C
      NEXT = 1
500   WRITE (2,10)
10    FORMAT (4/,20X,"ABRAMOVITZ - PHD.H",//)
      WRITE (2,11)
11    FORMAT (/,"DATE : ←")
      READ (1,12) (IDATE(I),I=1,10)
12    FORMAT (10A2)
      WRITE (2,40)
40    FORMAT (/,"TAPE(1) OR TTY(2) : ←")
      READ (1,*) I
      GO TO (92,93),I
92    READ (5,*) (NAME(I),I=1,6)
      READ (5,*) IBLCT,IBLCN,ISESN
      READ (5,*) ITIMT,ITIMC,ITIMF,ITIMP,ITIMS,ITIMW
      READ (5,*) NFEED,INEGN,INEGR,IPOSN,IPOSR,INEGP
      READ (5,*) ALOW,AMID,HIGH
      READ (5,*) IFPAY
      GO TO (94,95),IFPAY
95    READ (5,*) IBFWD,IBASE,IERP,ISPD
      READ (5,*) AVER,SDEV
94    CONTINUE
      GO TO 90
93    WRITE (2,13)
13    FORMAT (/,"SUBJECT : ←")
```

```
      READ (1,12) (NAME(I),I=1,6)
      WRITE (2,14)
14    FORMAT (/,"BLOCT BLOCN SESSN")
      READ (1,*) IBLCT,IBLCN,ISESN
      WRITE (2,15)
15    FORMAT (/,"TIMET TIMEC TIMEF TIMEP TIMES TIMEW")
      READ (1,*) ITIMT,ITIMC,ITIMF,ITIMP,ITIMS,ITIMW
      WRITE (2,16)
16    FORMAT (/,"NFEED PNEGN PNEGR PPOSN PPOSR +OR-F")
      READ (1,*) NFEED,INEGN,INEGR,IPOSN,IPOSR,INEGP
      WRITE (2,17)
17    FORMAT (/,"DIGITIZER LIMITS : L,M,H : ←")
      READ (1,*) ALOW,AMID,HIGH
      WRITE (2,18)
18    FORMAT (/,"PAYOFFS - NO(1), YES(2) : ←")
      READ (1,*) IFPAY
      GO TO (90,91),IFPAY
91    WRITE (2,21)
21    FORMAT ("B/FWD BASIC ERROR SPEED")
      READ (1,*) IBFWD,IBASE,IERP,ISPD
      WRITE (2,28)
28    FORMAT ("MEAN & SDEV SPECS : ←")
      READ (1,*) AVER,SDEV
90    WRITE (2,19)
19    FORMAT (/,"STARTING TRIAL NUMBER : ←")
      READ (1,*) NTRL
      GO TO (510,520),NEXT
C
510   J = 1
      DO 100 I = 1,IBLCN
      K = I*IBLCT
      READ (5,20) (IRAN(L),L=J,K)
20    FORMAT (2513)
      J = K+1
100   CONTINUE
      J = IBLCN/IBLCT
      I = J*IBLCT
      IF (IBLCN - I) 87,87,88
88    J = J+1
87    K = 1
      DO 89 I = 1,J
      L = I*IBLCT
      IF (IBLCN - L) 86,85,85
86    L = IBLCN
85    READ (5,20) (IREV(N),N=K,L)
      K = L+1
89    CONTINUE
      NEXT = 2
C
520   DO 110 I = 1,2
      DO 110 J = 1,18
      IEARL(J,I) = 0
      IRONG(J,I) = 0
```

```
      ILATR(J,I) = 0
      NUMBS(J,I) = 0
      SUM(J,I) = 0.
      SQS(J,I) = 0.
      IF (J - 5) 105,105,110
105   IEAR(J,I) = 0
      IRON(J,I) = 0
      ILAT(J,I) = 0
      NUMB(J,I) = 0
      SUMS(J,I) = 0.
      SQSS(J,I) = 0.
110   CONTINUE
C
      INIT = NTRL
      NTRL = NTRL - 1
      JERR = 0
      NERNN = 1
      NERNR = 1
      NERPN = 1
      NERPR = 1
      ICNT = 0
     NER = 0
      SUME = 0.
      BSUM = 0.
     BSQS = 0.
     IBER = 0
   IPAYA = 0
     IEARN = IBASE
      DIVR = NFEED*IBLCT
      WRITE (2,25)
25    FORMAT (///)
      PAUSE
C
C
530   DO 200 I = 1,IBLCN
     DO 190 J = 1,IBLCT
C
     K = (I-1)*IBLCT + J
      ISTIM = IRAN(K)
      STIM = ISTIM
      IF (IREV(I) - J)120,130,120
130   IFREV = 1
      GO TO 140
120   IFREV = 0
140   IOPS = 1
      IFERF = 1
      IF (ISTIM) 141,145,145
141   IOPS = ISTIM + 3
      IPOS = IOPS
      GO TO 180
145   IF(STIM - ALOW) 146,146,150
146   IOPS = 2
```

```
      IPOS = 5
      GO TO 180
150   IF (STIM - AMID) 151,151,155
151   IOPS = 2
      IPOS = 4
      IFERF = 0
      GO TO 180
155   IF (STIM - HIGH) 156,156,157
156   IPOS = 4
      IFERF = 0
      GO TO 180
157   IPOS = 3
      GO TO 180
180   IF (IFERF) 160,160,165
165   IF (NERNN - INEGN) 166,167,168
166   IFENN = 0
      GO TO 170
167   IFENN = 1
      GO TO 170
168   NERNN = 1
      GO TO 165
170   IF (NERNR - INEGR) 171,172,173
171   IFENR = 0
      GO TO 175
172   IFENR = 1
      GO TO 175
173   NERNR = 1
      GO TO 170
175   IF (NERPN - IPOSN) 176,177,178
176   IFEPN = 0
      GO TO 181
177   IFEPN = 1
      GO TO 181
178   NERPN = 1
      GO TO 175
181   IF (NERPR - IPOSR) 182,183,184
182   IFEPR = 0
      GO TO 160
183   IFEPR = 1
      GO TO 160
184   NERPR = 1
      GO TO 181
160   CALL OUTA
C
      ISTIM = ISTIM + 3
     IFREV = IFREV + 1
      NTRL = NTRL + 1
     IF (IER1) 205,205,210
210   IEARL(ISTIM,IFREV) = IEARL(ISTIM,IFREV) + 1
     IEAR(IPOS,IFREV) = IEAR(IPOS,IFREV) + 1
     JERR = JERR + 1
```

```
      NER = NER + 1
      IBER = IBER + 1
      GO TO 240
205   IF (IER3) 215,215,220
220   ILATR(ISTIM,IFREV) = ILATR(ISTIM,IFREV) + 1
     ILAT(IPOS,IFREV) = ILAT(IPOS,IFREV) + 1
     JERR = JERR + 1
      NER = NER + 1
      IBER = IBER + 1
      GO TO 240
215   ATE = ILATE
      NUMBS(ISTIM,IFREV) = NUMBS(ISTIM,IFREV) + 1
     NUMB(IPOS,IFREV) = NUMB(IPOS,IFREV) + 1
     SUM(ISTIM,IFREV) = SUM(ISTIM,IFREV) + ATE
     SQS(ISTIM,IFREV) = SQS(ISTIM,IFREV) + ATE**2
     SUMS(IPOS,IFREV) = SUMS(IPOS,IFREV) + ATE
     SQSS(IPOS,IFREV) = SQSS(IPOS,IFREV) + ATE**2
      SUME = SUME + ATE
      BSUM = BSUM + ATE
      BSQS = BSQS + ATE**2
      IF (IER2) 240,240,230
230    IRONG(ISTIM,IFREV) = IRONG(ISTIM,IFREV) + 1
      IRON(IPOS,IFREV) = IRON(IPOS,IFREV) + 1
    NER2 = NER2 + 1
      JERR = JERR + 1
240   CONTINUE
190   CONTINUE
C
      ICNT = ICNT + 1
      IF (ICNT - NFEED) 300,310,300
310   ICNT = 0
      CALL LIGHT
C
      GO TO (355,330),IFPAY
330   ER = NER
      SUME = SUME / (DIVR-ER)
      SUME = (AVER-SUME)/SDEV
      IF (SUME) 331,331,335
335   ISUME = SUME
      IF (ISUME - 3) 340,340,341
341    ISUME = 3
340   IEARN = IEARN + ISUME*ISPD
331    SUME = 0.
      NER = 0
      IF (JERR - 2) 345,345,350
345   IEARN = IEARN + (IERP/(JERR+1))
350   IF (IPAYA - IEARN) 351,355,355
351   IPAYA = IEARN
      GO TO (352,353),INEGP
352   WRITE (2,22) NTRL,JERR,IPAYA
22    FORMAT (3(I5,1X))
```

```
      GO TO 360
353   GERR = JERR
      GERR = 100. *(DIVR-GERR)/DIVR
      WRITE (2,31) NTRL,GERR,IPAYA
31    FORMAT (I5,2X,F5.1,2X,I4)
      GO TO 360
355   GO TO (356,357),INEGP
356   WRITE (2,22) NTRL,JERR
      GO TO 360
357   GERR = JERR
      GERR = 100. *(DIVR-GERR)/DIVR
      WRITE (2,31) NTRL,GERR
      GO TO 360
360   JERR = 0
      WRITE (4,23) NTRL
23    FORMAT (I6)
C
      DO 370 J = 1,2
      DO 370 K = 1,5
      WRITE (4,24) NUMB(K,J),SUMS(K,J),SQSS(K,J),IEAR(K,J),
     1 IRON(K,J),ILAT(K,J)
24    FORMAT (I4,2F15.2,3I6)
      NUMB(K,J) = 0
      SUMS(K,J) = 0.
      SQSS(K,J) = 0.
      IEAR(K,J) = 0
      IRON(K,J) = 0
      ILAT(K,J) = 0
370   CONTINUE
      WRITE (4,29)
29    FORMAT (" ")
C
C
    IF (NTRL-INIT-ISESN+1) 300,400,400
C
300   CONTINUE
200   CONTINUE
      GO TO 530
C
C
400   WRITE (2,25)
     GO TO (420,430),IFPAY
430   I = IBFWD + IEARN
      WRITE (2,26) IBFWD,IEARN,I
26    FORMAT (/,5X,I4," CENTS B/FWD",/,"PLUS ",I4," CENTS",/,
1 2X,"=  ",I4," CENTS TO DATE")
420   TRLS = NTRL - INIT - IBER + 1
      BSQS = SQRT ((BSQS-(BSUM**2/TRLS))/(TRLS-1.))
      BSUM = BSUM/TRLS
      WRITE (2,30) BSUM,BSQS,TRLS
30    FORMAT (/,"MEAN RT ",F8.2,"  SDEV RT ",F8.2,"  OVER ",
     1 F7.1," TRIALS")
```

```
      WRITE (2,27) INIT,NTRL,(IDATE(I),I=1,10),(NAME(I),I=1,6)
27    FORMAT (/,"TRIALS FROM ",I6," TO ",I6," ON ",10A2,
     1 " WITH ",6A2,///)
      DO 410 I = 1,2
      DO 409 J = 1,18
      IF (NUMBS(J,I)) 411,411,412
412   TRLS = NUMBS(J,I)
      SQS(J,I) = SQRT ((SQS(J,I)-(SUM(J,I)**2/TRLS))/(TRLS-1.))
      SUM(J,I) = SUM(J,I)/TRLS
411   K = J-3
      WRITE (2,41) K,NUMBS(J,I),SUM(J,I),SQS(J,I),IEARL(J,I),
     1 IRONG(J,I),ILATR(J,I)
41    FORMAT (I3,2X,I5,2X,2F15.2,3I6)
409   CONTINUE
      WRITE (2,32)
32    FORMAT (/)
410   CONTINUE
      WRITE (2,25)
      CALL WAIT
      GO TO 500
      END
      END$
```

```
ASMB,R,B
       NAM OUTA
       ENT OUTA
*
       COM TIMET,TIMEC,TIMEF,TIMEP,TIMES
       COM ER1,ER2,ER3,STIM,IFREV,OPS
       COM IFENN,IFENR,IFEPN,IFEPR,IFERF
       COM LATEN,NEGN,NEGR,POSN,POSR
*
TBG    EQU 10B
INR    EQU 14B
OUTR   EQU 15B
*
TIME   BSS 1
TONEC  BSS 1
COLRC  BSS 1
IFRES  BSS 1
RESP   BSS 1
*
BUZZ1  OCT 20
BUZZ2  OCT 40
BUZZ3  OCT 200
*
*
OUTA   NOP
        ISZ OUTA
       CLF 0B
       CLA,INA
       OTA TBG
       CLB
       STB IFRES
       STB ER1
       STB ER2
       STB ER3
       STB LATEN
*
       LDA STIM
       SSA

       JMP SATUR
       CLB,INB
       RBR
       ALF,ALF
       ALF,RAR
       ADA 1B
*
TONES  LDB IFREV
       SZB
       JSB REVER
*
RETN   LDB OPS
       RBL,RBL
```

```
       STB TONEC
       ADB ØB
       STB COLRC
*
PAUS   CLA
       LDB TIMEP
       JSB TIMER
       LDB TIMET
       LDA TONEC
       OTA 1B
       OTA OUTR
       CLA,INA
       JSB TIMER
       LDA IFRES
       SZA
       JMP EROR1
       LDB TIMEC
       LDA COLRC
       OTA 1B
       OTA OUTR
       CLA,INA
       JSB TIMER
       LDA IFRES
       SZA,RSS
       JMP EROR3
       LDA RESP
       CPA OPS
       JMP ERO2P
       JMP ERO2N
*
*
SATUR ADA =D3
       LDB IFREV
       SZB
       JSB REVER
       JMP RETN
       STB OPS
       JMP RETN+1
*
REVER NOP
       ISZ REVER
       LDB OPS
       ADB =D-1
       SZB
       ADB =D-2
       ADB =D2
       JMP REVER,I
*
TIMER NOP
       CMB,INB
       STB TIME
     STC TBG,C
```

```
    SZA
        JSB INPUT
        SFS TBG
        JMP *-1
        STC TBG,C
        ISZ TIME
        JMP *-6
        JMP TIMER,I
*
INPUT NOP
        LIB INR
        SZB,RSS
        JMP INPUT,I
     CPB =B4
        JMP STOPE
        LDA IFRES
        SZA
        JMP INPUT,I
          ISZ IFRES
        STB RESP
       LDA TIME
        ADA TIMEC
       STA LATEN
     BLF,BLF
        MIB 1B
        OTB 1B
        JMP INPUT,I
*
EROR3 CLA
        STA LATEN
        INA
        STA ER3
        JMP *+3
*
EROR1 CLA,INA
        STA ER1
        LDA BUZZ1
        OTA OUTR
        MIA 1B
        OTA 1B
        CLA
        LDB TIMEF
        JSB TIMER
        JMP RETRN
*
ER02P LDA IFERF
        SZA,RSS
        JMP RETRN
        LDA IFREV
        SZA,RSS
        JMP POSNO
```

```
       ISZ POSR
        LDA IFEPR
        SZA,RSS
       JMP RETRN
        LDA BUZZ3
        JMP ERON
*
POSNO ISZ POSN
       LDA IFEPN
       SZA,RSS
       JMP RETRN
       LDA BUZZ3
       JMP ERON
*
ERO2N ISZ ER2
       LDA IFERF
       SZA,RSS
       JMP RETRN
       LDA IFREV
       SZA
       JMP NEGRE
       ISZ NEGN
       LDA IFENN
       SZA,RSS
       JMP RETRN
       LDA BUZZ2
       JMP ERON
*
NEGRE ISZ NEGR
       LDA IFENR
       SZA,RSS
       JMP RETRN
       LDA BUZZ2
*
ERON   OTA OUTR
       MIA 1B
       OTA 1B
       CLA
       LDB TIMEF
       JSB TIMER
       JMP RETRN
*
STOPE CLC TBG,C
       CLA
       OTA OUTR
       CMA
       OTA 1B
       HLT 77B
       JMP OUTA+2
*
RETRN CLC TBG,C
       CLA
```

```
        OTA  OUTR
        OTA  1B
        STF  ØB
        JMP  OUTA,I
*
*
        END
```

```
ASMB,R,B
       NAM WAIT
       ENT WAIT
       COM SPACE(21),TIMEW
TIME   BSS 1
TONE   OCT 200
TBG    EQU 10B
OUTR   EQU 15B
*
WAIT   NOP
       ISZ WAIT
       CLF 0B
       LDA =B4
       OTA TBG
       LDA TIMEW
       CMA,INA
       STA TIME
       STC TBG,C
       SFS TBG
       JMP *-1
       ISZ TIME
       JMP *-4
       LDA TONE
       OTA OUTR
       STC TBG,C
       SFS TBG
       JMP *-1
       CLA
       OTA OUTR
       CLC TBG,C
       STF 0B
       JMP WAIT,I
       END
```

```
     ASMB,R,B
       NAM LIGHT
       ENT LIGHT
*
      COM SPACE(4),TIMES,S(17),TIMER
*
LITE   OCT 400
OUTR EQU 15B
TBG  EQU 10B
SAVEA BSS 1
*
LIGHT NOP
       ISZ LIGHT
       LDA LITE
       OTA OUTR
       OTA 1B
       CLA,INA
       OTA TBG
       LDA TIMES
       CMA,INA
       STA TIMER
       LDA IJSB
       STA TBG
       STC TBG,C
       JMP LIGHT,I
*
IJSB   JSB LINK,I
       ORB
LINK   DEF CONT
       ORR
*
CONT   NOP
       STA SAVEA
       STC TBG,C
       ISZ TIMER
       JMP CONT,I
*
       CLC TBG,C
       CLA
       OTA 1B
       OTA OUTR
       LDA SAVEA
       JMP CONT,I
*
       END
```

Programs from Marshall (1975)

These were quite straightforward programs, designed to create and then execute reinforcement schedules. They are well documented in Marshall (1975). Subroutines TIMER and RAND were used; TIMER performed the functions outlined elsewhere, while RAND provided a random digit for the variable reinforcement schedules.

```
FTN,B
      PROGRAM ANDY2
      DIMENSION VAL(200)
      COMMON IBASE,ITIME,IOUT,INPUT
      WRITE(2,10)
10    FORMAT(///," MEAN INTERVAL: ←")
      READ(1,*)AV
      WRITE(2,11)
11    FORMAT(" NO OF INTERVALS: ←")
      READ(1,*)NTERM
      TERM=NTERM
      X=1.0+ALOG(TERM)
      WRITE(2,12)
12    FORMAT(" COMPUTING")
     DO 100 I=1,NTERM
      Z=I
      Y=TERM-Z
      IF(Y)110,110,120
110   W=Y-(Y+1.0)*ALOG(Y+1.0)
      GOTO 130
120   W=X+Y*ALOG(Y)-(Y+1,0)*ALOG(Y+1.0)
130   VAL(I)=W*AV
100   CONTINUE
      WRITE(2,13)
13    FORMAT(" OUTPUT PLUG: ←")
      READ(1,*)IPLUG
      J=16001
      WRITE(2,18)
18    FORMAT(" RAND NO: ←")
      READ(1,*)I
```

```
      DO 140 K=1,I
      CALL RAND(J,X)
140   CONTINUE
     IBASE=2
     CALL TIMER
200   ITIME=0
      IOUT=0
      CALL RAND(J,X)
      X=TERM*X+1.0
      WRITE(2,17)X
17    FORMAT(" RAND= ",F8.1)
      I=X
      INTER=VAL(I)
      IF(INTER-600)200,200,205
205   WRITE(2,14)INTER
14    FORMAT(/," INTER= ",F8.1)
210   CONTINUE
      IF(INPUT)210,210,300
300   CONTINUE
      WRITE(2,15)
15    FORMAT(" INPUT")
330   CONTINUE
      IF(ITIME-INTER)330,310,310
310   IOUT=IPLUG
      ITIME=0
      WRITE(2,16)
16    FORMAT(" NOW")
320   CONTINUE
      IF(ITIME-25)320,320,200
      END
      END$
```

```
FTN,B
    SUBROUTINE RAND(I,X)
      J=I*899
      IF(J)10,20,20
10    J=J+32767+1
20    X=J
      X=X/32768.0
      I=J
      RETURN
      END
     END$
```

Programs from Oxtoby (1971)

These programs are anotated to shown their function. The purpose was to present a series of time intervals, marked by 'pips', to a subject, who was then required to decide whether the last time interval was of the same duration as the preceeding intervals. The series was created by the main program, EXP02, and presented by the Assembler routine PRES.

```
FTN,B
      PROGRAM EXP02
C  **
C  **  THIS PROGRAM PRESENTS A SEQUENCE OF
C  **  TIME INTERVALS TO A SUBJECT.
C  **  THESE SEQUENCES MAY HAVE FROM ONE
C  **  TO SEVEN TIME INTERVALS IN THEM, OF
C  **  WHICH THE LAST INTERVAL IS ALWAYS THE
C  **  "TEST" INTERVAL.   THE INTERVALS PRE-
C  **  CEEDING THE TEST INTERVAL HAVE THE SAME
C  **  LENGTH, CALLED THE BASIC TIME INTERVAL.
C  **  THESE SEQUENCES ARE ARRANGED IN ORDER,
C  **  SUCH THAT THEIR TEST INTERVALS RANGE
C  **  FROM A VALUE LARGER THAN THE BASIC INTERVAL
C  **  TO A VALUES SMALLER THAN THIS.   THERE
C  **  ARE NINE SUCH SEQUENCES IN SUCH AN ORDER
C  **  THE CONSTRUCTION OF THESE NINE TEST
C  **  INTERVALS IS SUCH THAT THE MIDDLE OR
C  **  5TH TEST IS EQUAL TO THE BASIC INTERVAL,
C  **  WHILE THE REST ARE SYMMETRICALLY SOME
C  **  MINIMUM TIME DISTANCE AWAY FROM THE
C  **  BASIC INTERVAL.   THIS MINIMUM DISTANCE
C  **  IS CALLED THE SERIES INCREMENT (IE A
C  **  SERIES COMPRISES NINE SEQUENCES).
C  **  FINALLY, THE TEST INTERVALS ARE ORDERED
C  **  BY THE ADDITION OR SUBTRACTION OF
C  **  MULTIPLES OF THE BASIC TIME INCREMENT
C  **
C  **  THE VARIABLES USED BY THE PROGRAM :
C
```

```
C**     INTER - THE VALUE OF THE BASIC TIME INTERVAL
C **              USUALLY 1000 OR 1500 MS
C **     INCRE - THE BASIC TIME INCREMENT.  IF INTER
C **              = 1000, ABOUT 10, IF = 1500, ABOUT 15
C **     IEXPT - SET = 1 TO INDICATE A NEW SETTING
C **              OF THE EXPERIMENTAL PARAMETERS NEEDED
C **     NUMI  - THE NUMBER OF PERIODS IN A SEQUENCE
C **              EXCLUDING THE TEST INTERVAL
C **     INCS  - THE SERIES INCREMENT.   THIS VARIES
C **              FROM 0 TO 70 MS
C **     IDIR  - THE DIRECTION OF THE PRESENTATION OF
C **               THE TESTS, EITHER FORWARD OR BACK-
C **              WARD.
C **     IFSAM - SET = 1 IF A SAMPLE OF THE SEQUENCES
C **           N IS TO BE PRESENTED TO S
C **     ISAM  - THE VALUES OF THE SAMPLE TESTS
C **
      DIMENSION ISAM(2),IADD(9),ITEST(2,9)
      COMMON IDIR,IFSAM,NUMI,INCS,IEXPT,ITIM(9)
C
      NUMT = 9
      IPRNT = 2
      IREAD = 1
      ITYPW = 2
      ITYPR = 1
C **
C **  REQUEST THE EXPERIMENTAL PARAMETERS ON
C **  THE TELETYPE, AND READ IN THEIR VALUES
C **
200   WRITE (IPRNT,10)
10    FORMAT (///," BASIC TIME INTERVAL, MS?")
      READ (IREAD,11) INTER
11    FORMAT (I5)
      WRITE (IPRNT,12)
12    FORMAT (/," BASIC TIME INCREMENT, MS?")
      READ (IREAD,11) INCRE
500   WRITE (IPRNT,13)
13    FORMAT (/,"NEW SERIES?")
     READ (ITYPR,20) IEXPT
20    FORMAT (I1)
      IF (IEXPT) 100,100,200
100   WRITE (ITYPW,21)
21    FORMAT ("NO OF PERIODS?")
      READ (ITYPR,20) NUMI
     WRITE (ITYPW,22)
22    FORMAT ("SERIES INCREMENT?")
      READ (ITYPR,23) INCS
23    FORMAT (I3)
     WRITE (ITYPW,24)
24    FORMAT ("DIRECTION?")
      READ (ITYPR,20) IDIR
      WRITE (ITYPW,25)
25    FORMAT ("SAMPLE?")
```

```
      READ (ITYPR,20) IFSAM
C **
C **  IF A PAIR OF SAMPLE SEQUENCES ARE REQUIRED
C **  REQUEST THEIR VALUES AND READ THEM IN
C **
      IF (IFSAM) 110,110,210
210   WRITE (IPRNT,14)
14    FORMAT ("SAMPLE DURATIONS?")
     READ (IREAD,15) ISAM(1),ISAM(2)
15    FORMAT (2I5)
C **
C **  TYPE "READY" ON THE TTY
C **
110   WRITE (IPRNT,16)
16    FORMAT (//," READY",/)
C **
C **  IF A SAMPLE, THEN ESTABLISH THE VALUES
C **  OF THE SEQUENCE'S INTERVALUES INTO "ITIM"
C **  AND CALL THE ASSEMBLER SUB-PROGRAM TO
C **  PRESENT THEM
C **
      IF (IFSAM) 300,300,400
400   DO 410 IS = 1,2
      DO 420 J = 1,NUMI
420   ITIM(J) = INTER
    ITIM(NUMI+1) = ISAM(IS)
      WRITE (ITYPW,26) IS
26    FORMAT (I1)
CA    CALL PRES
410   CONTINUE
C **
C **  TYPE ON THE TTY "END OF SAMPLE"
C **
      WRITE (ITYPW,27)
27    FORMAT (/,"END OF SAMPLE",/)
C
C **
C **  ESTABLISH THE SERIES TIME INTERVALS
C **  FROM THE EXPERIMENTAL PARAMETERS
C **
C
300   INUMT = NUMT/2
      JNUMT = INUMT + 1
      DO 310 I = 1,INUMT
      IADD(I) = INCS
      IS = I
      DO 320 J = 1,IS
320   IADD(I) = IADD(I) + INCRE
      K = I + JNUMT
      IADD(K) = -IADD(I)
```

```
310   CONTINUE
      IADD(JNUMT) = 0
      DO 340 I = 1,NUMT
      ITEST(1,I) = INTER + IADD(I)
      ITEST(2,I) = INTER - IADD(I)
340   CONTINUE
C  **
C  **  PRESENT THE TIME INTERVALS BY CALLING
C  **  THE ASSEMBLER SUB-PROGRAM "PRES"
C  **  AS EACH SEQUENCE IS PRESENTED, TYPE
C  **  OUT ON THE TTY ITS SEUQENCE NUMBER
C  **
      DO 350 I = 1,NUMT
      DO 360 J = 1,NUMI
360   ITIM(J) = INTER
      ITIM(NUMI+1) = ITEST(IDIR,I)
      WRITE (ITYPW,28) I
28    FORMAT (I1)
      CALL PRES
350   CONTINUE
C  **
C  **  AT THE END OF A SERIES, GO TO THE START
C  **
      GO TO 500
      END
      END$
```

```
ASMB,R,B
       NAM PRES
       ENT PRES
*
*   THIS ASSEMBLER SUB-PROGRAM SIMPLY PRESENTS
*   A NUMBER OF OUTPUT "PIPS" WHICH DELINEATE
*   THAT NUMBER OF TIME INTERVALS
*   THE VALUES OF THE TIME INTERVALS ARE
*   CONTAINED IN "ITIM", AND THE NUMBER OF
*   THESE INTERVALS ARE CONTAINED IN "NUMI"
*
*
       COM SPAC(2),NUMI,SPAS(2),ITIM(9)
TIM    DEF ITIM
TIMI   DEF ITIM
TIMEN  DEF ITIM+9
TBG    EQU 10B
OUTR   EQU 15B
INPUT  EQU 14B
TIME   BSS 1
*
PRES   NOP
*
*   MONITOR THE INPUT KEY FOR A TAP FROM
*   E TO INDICATE THAT THE TTY HAS BEEN
*   TURNED OFF, THEN PROCEED
*
       LIA INPUT
       SZA,RSS
       JMP *-2
```

```
       LDA PRES,I
       STA PRES
       LDA TIM
       ADA NUMI
       INA
       STA TIMEN      SET LIMITS OF TIMES
*
       CLF ØB
       CLA,INA
       OTA TBG        SET TBG 1MS INTERVAL
*
       LDA TIMI,I     LOAD CURENT TIME INTERVAL
       CMA,INA
       STA TIME       STORE ITS COMPLEMENT
*
       CLB
       CLA,INA
       OTA OUTR       TURN ON THE OUTPUT
*
       STC TBG,C      START THE CLOCK
*
CARRY SFS TBG
       JMP *-1
*
       STC TBG,C      RESTART THE CLOCK IMMEDIATELY
       INB
       CPB =D5Ø
       JSB OFF        ALLOW 5Ø MS BEFORE SWITCH OFF
*
       ISZ TIME       DECREMENT TIMER
       JMP CARRY
*
       CLA,INA
       OTA OUTR       OUTPUT EVENT IMMEDIATELY AFTER END
ISZ TIMI       OF LAST EVENT
       LDA TIMI,I
       CMA,INA
       STA TIME       STORE NEW INTERVAL
*
       LDB TIMEN
       CPB TIMI       CHECK FOR END OF SERIES
      JMP ENDS
*
       CLB
       JMP CARRY
*
*   THIS ROUTINE TURNS OFF AN OUTPUT EVENT
*
OFF    NOP
       CLA
```

```
        OTA OUTR
        JMP OFF,I
*
*
*   THE END OF THE SERIES
*   ALLOW 50 MS BEFORE SWITCH OFF
*
ENDS    LDA =D50
        CMA,INA
        STA TIME
        STC TBG,C
        SFS TBG
        JMP *-1
        ISZ TIME
        JMP *-4
        JSB OFF
*
*
*   MONITOR THE INPUT KEY AGAIN, FOR E'S
*   INDICATION THAT THE TTY HAS BEEN SWITCHED
*   ON.   PAUSE FOR 200 MS, THEN RETURN TO THE
*   MAIN CALLING PROGRAM
*
        LDA TIM
        STA TIMI
        LIA INPUT
        SZA,RSS
        JMP *-2
*
*
        LDA =D-200
        STC TBG,C
        SFS TBG
        JMP *-1
        INA
        SZA
        JMP *-5
*
      CLC TBG,C
STF 0
        JMP PRES,I
*
END
```