Copyright is owned by the Author of the thesis. Permission is given for a copy to be downloaded by an individual for the purpose of research and private study only. The thesis may not be reproduced elsewhere without the permission of the Author.

SOME THEORETICAL STUDIES OF INTELLIGENCE:

EXTENSIONS AND TESTS OF ASPECTS OF PIAGET'S MODEL

A THESIS PRESENTED IN PARTIAL FULFILMENT
OF THE REQUIREMENTS FOR THE DEGREE
OF MASTER OF ARTS
IN PSYCHOLOGY AT
MASSEY UNIVERSITY

K.E. VOGES

1979

To

Heather

ABSTRACT

This study attempts to integrate two particular aspects of Piaget's theory, seriation of weight behavior and the equilibration model, with the recently developed information processing approaches to psychology, derived from the development and proliferation of the computer. It is felt that computer based technologies will be more widely used in the future in psychology, to help solve a variety of common problems. A major contemporary problem that the computer may help to overcome, is that of trying to integrate the great mass of experimental data that has been collected over the years of psychological investigation.

The paper has a number of major sections, a preliminary section discussing methodology, a section concerned with single subject performance on a weight seriation task, and a section concerned with Piaget's theoretical model, equilibration. A review is given of a range of modern approaches to psychology, including computer-based approaches and systems science. As well, the details of the methodologies of computer simulation, systems science, and single subject performance models are considered.

The actual experimental work deals with the development of a production system based on a study by Baylor and Gascon (1974), which deals with the seriation of a number of weights. Computer programs were written, in PASCAL to facilitate portability, to simulate the weight seriation

behavior of children in the age ranges of approximately 6 to 9 (Stages 1, 2 and 3). As well as validating the findings of Baylor and Gascon, a further approach to weight seriation problems was discovered for the Stage 1 child (this approach has been termed "Opposite-placement of couples" to complement Baylor and Gascon's "Juxtaposition of opposites". These programs are presented in the paper, along with examples of the print out obtained, and a comparison between these and the children's protocols were made. Only children exhibiting Stage 1 and Stage 3 behavior were considered in detail, as it was felt that the Stage 2 behavior was too complex and required more extensive observation before actual modelling of the behavior should be attempted.

Piaget's equilibration model is reviewed and discussed, with emphasis being placed on its relationship to cybernetics. As well, other mechanisms of the transition from one stage to another are considered, specifically the models of Klahr and Wallace (1976) and Pascual-Leone (1970).

It was concluded that the models embedded in the programs give an accurate simulation of the various types of behavior observed through the age range discussed. Various extensions of the work are also considered, relating to the integration of other aspects of cognitive and motivational aspects into the model. It is felt finally that it is important to extend work in the area of representation of the environment to enable the program to have something to manipulate. Piaget's work already has an information processing flavour, through his development of the concept of operations, but there is hardly any work based on the processing of representations of the environment. This helps give a direction to future research.

PREFACE

This study stems from an interest in Piagetian theory associated with operations, equilibration and cybernetics, and the challenge to relate some of these concepts with aspects of computer simulation. I believe the computer has an important, indeed vital role to play in the future of theoretical psychology. After reading many authors who are convinced of the truth of this belief, I feel that my attempt to acquire some knowledge in this area has been worthwhile. Psychology can make from the transition from a qualitative descriptive science to a quantitative, analytical one. If the models produced are to serve any useful purpose they must yield predictions that can be tested against experimental data. Because psychological systems are generally very complex, realistic models capable of solution could not have been formulated in the precomputer era. Now computational difficulties should no longer be a limiting factor in the development of a quantitative theoretical psychology. The only limitation now lies in our ability to formulate hypotheses about the operation of psychological systems, and to develop new mathematical and experimental techniques to test them.

I would like to take this opportunity to thank Dr. Don McAlpine for his understanding and help during his supervision of this work. Conversations with Don are

always (for me) an intellectually stimulating experience, and I thank him for those experiences. I would also like to thank Lindsay Groves from the Computer Science department for putting up with my abysmal ignorance concerning the workings of computer programs, and for giving me a considerable amount of his time and knowledge to increase my understanding of the process. Finally, I would like to thank my wife, for her patience, and for her support, without which this thesis would not have been completed.

K.E. Voges
November, 1979

CONTENTS

Preface

Table of Contents

List of Figures

PART I PRELIMINARIES

CHAPTER 1 A STATEMENT OF THE PROBLEM 2

CHAPTER 2 SOLUTIONS AND APPROACHES 5

1. Solutions 5
1.(1). Language 5
1.(2). Pictorial 6
1.(3). Mathematical 7

2. Approaches 9
2.(1). Computer-based 9
2.(1).(a) Information-processing approaches to psychology 9
(i) Advantages 9
(ii) Disadvantages 10
2.(1).(b) Computer simulation 12
2.(2). Systems Science 14

CHAPTER 3 METHODOLOGY 17

1. Computer Simulation 17

1.(1). Data collection 17

1.(2). Developing and programming the model 18

1.(3). Analysis of results 20

2. Systems Science 21

2.(1). Analysis 22

2.(2). Root definition of relevant systems 23

2.(3). Conceptualization 24

2.(4). Comparison and definition of possible changes 24

2.(5). Design and implementation 24

2.(6). Appraisal 25

3. Single Subject Performance Models 26

4. Conclusion 30

PART II THE PERFORMANCE MODELS OF A NUMBER OF INDIVIDUAL SUBJECTS

CHAPTER 4 EXPERIMENTAL STUDIES: PIAGET'S WORK 35

1. Grouping V 36

2. Characteristics of Stages 37

3. Piaget's Work 38

CHAPTER 5 EXPERIMENTAL STUDIES: LENGTH : SERIATION 41

CHAPTER 6 EXPERIMENTAL STUDIES: WEIGHT SERIATION 45

Baylor's Studies 46

1. Stage 1 46

2. Stage 2 49

3. Stage 3 52

4. Concluding Remarks 55

CHAPTER 7 A REPORT ON THE PROGRESS OF THE PRESENT STUDY 56

1. Programming Aspect 56

2. Experimental Aspect 59

2.(1). Subjects 59

2.(2). Procedure 60

2.(3). Results 61

3. Conclusions 63

3.(1). Stage 1 63

3.(2). Stage 3 65

PART III GENERAL MODELS OF DEVELOPMENT

CHAPTER 8 PIAGET'S TRANSITION MECHANISM: EQUILIBRATION 76

1. Cybernetics 77

2. Equilibration 79

3. The Relationship Between Cybernetics and Equilibration 82

PART IV CONCLUSION

CONCLUSIONS 86

PART V APPENDICIES AND REFERENCES

APPENDIX I STAGE 1A WEIGHT SERIATION PROGRAM 91

APPENDIX II STAGE 1B WEIGHT SERIATION PROGRAM 102

APPENDIX III STAGE 1 (ALTERNATIVE) WEIGHT SERIATION PROGRAM 113

APPENDIX IV STAGE 3 WEIGHT SERIATION PROGRAM 124

APPENDIX V STAGE 3 (UPGRADED) WEIGHT SERIATION PROGRAM 140

APPENDIX VI OTHER TRANSITION MECHANISMS 156

REFERENCES 160

LIST OF FIGURES

1. Relationship between theory and model. 13

2. Relationship between model and object system. 14

3. A model for models. 15

4. Systems science methodology. 22

5. Guidelines for an analysis of the problem situation. 23

6. A summary of the systems methodology. 25

7. Operating rules of a production system. 27

8. Anticipation and performance in seriation. 39

9. Base strategy for Stage 1 weight-seriation behavior. 47

10. Production rules for Stage 1(A and B) weight seriation behavior. 48

11. Base strategy for Stage 2 weight-seriation behavior. 49

12. Production rules for Stage 2 weight seriation behavior. 51 - 52

13. Base strategy for Stage 3 weight-seriation behavior ("Locate Heaviest"). 53

14. Base strategy for Stage 3 weight-seriation behavior ("Insertion"). 53

15. Production rules for Stage 3 weight seriation behavior. 54

16. Decision flow of program indicating initial goals and conditions for the production system (Stage 1). 58

17. Decision flow of program indicating initial goals and conditions for the production system (Stage 3). 59

18. Diagram showing experimental set up, viewed from top. 60

19. Simplified example of print out for simulation of Stage 1A behavior (from Appendix I). 64

20. Protocol for Tor (6;8) on a weight seriation task. 65

21. Protocol for Kevin (6;10) on a weight seriation task. 66

22. Simplified example of print out for simulation of Stage 1 Alternative behavior (from Appendix III). 67

23. Protocol for Samuel (7;11) on a weight seriation task. 68

24. Simplified example of print out for simulation of Stage 3 behavior (from Appendix IV). 69 - 71

25. Protocol for Warwick (8;4) on a weight seriation task. 72 - 74

PART I

PRELIMINARIES

CHAPTER 1

A STATEMENT OF THE PROBLEM

In many respects this thesis has as its main concern a methodological problem, outlined at different times by different theorists. It represents a growing dissatisfaction with a science that seems to be becoming an encyclopedia of known facts rather than an interpreter and organizer of these facts. This dissatisfaction was crystallized in an article by Newell (1973), in which he considers a very real problem emerging in psychological research, or rather two interrelated problems. The first concerns a lack of direction in research. Newell lists 59 different phenomena (by no means a final or exhaustive list) with which experimental psychology deals, and is pessimistic about any finality being reached in any study. 'Matters simply become mudier and mudier as we go down through time. Thus, far from providing the rungs of a ladder by which psychology gradually climbs to clarity, this form of conceptual structure leads rather to an ever increasing pile of issues, which we weary of or become diverted from, but never really settle' (Newell, 1973, pp.288 - 289). We seem to consider research as another consumable commodity. The research is performed with little consideration for extension or integration with current theory. This is Newell's second problem. 'We never seem in the experimental literature to put the results of all the experiments

together.... We do...relate sets of experiments. But the linkage is extraordinarily loose. One picks and chooses among the qualitative summaries of a given set of experiments what to bring forward and juxtapose with the concerns of a present treatment' (Newell, 1973, p.298). The problem concerns an integration of the great mass of data that has been gathered over the last hundred years since psychology has been considered an experimental science.

Other theorists, such as Deutsch (1960) have commented on the same problem. In his book, Deutsch suggests that there are too many facts in psychology, and the task is now to interpret them. Lewin was also aware of this problem. In a far-ranging paper (Lewin, 1965, p.201) he states: 'The need for a closer fusion of the various branches of psychology demands tools which permit better integration.' He was also aware of the problems with such an approach. 'We will produce but an empty formalism, if we forget that mathematization and formalization should only be done to the degree that the maturity of the material under investigation permits at a given time' (Lewin, 1965, p.201). One of his major concerns in the presentation of his theory was to bring together the total field of psychology in a logically consistent manner. He believed such a goal would be impossible without adequate mathematization. If a theory could combine into one logical system of known facts which previously could be tested only by separate theories, this particular theory would have great value as an organizational device. Lewin was of the opinion that concepts of topology would prove 'fruitful for representation in every field of psychology' (Lewin, 1965, p.215).

In many scientific fields, the concentration over the past several decades has been on analytical fact-finding, and experimental approaches in highly specific areas. This has been useful in helping to develop knowledge and to understand the details of specific but limited subjects. At some time however, there should be a period of synthesis, reconciliation and integration, so that the analytical and fact-finding elements are unified into broader, multi-

dimensional theories. 'There is evidence that every field of human knowledge passes alternatively through phases of analysis and fact finding to periods of synthesis and integration. Recently systems theory has provided this framework in many fields - physical, biological and social' (Beishon & Peters, 1972, p.16).

In the remainder of this section I shall consider some attempted solutions to the problems I have outlined, as well as the general methodological approach to be followed in this study. As will be seen, my main emphasis is towards what may be loosely termed a systems approach to the conceptualization of human cognitive activity. This new approach, following on from the theoretical work of Von Bertalanffy (1951), is being considered more and more in various physical and social sciences, and it is time that such an approach was investigated for its application to psychological issues. Coupled to this is the invention and proliferation of the high speed general purpose digital computer. This is I feel a significant step towards a very necessary tool designed to "put it all together", and make use of the immense wealth of data that has been gathered. A number of writers (Reitman, 1965; Apter, 1970) have considered the invention of the computer to be as important an event for psychology as the invention of the microscope was for biology. The exciting implications of this comparison have yet to be fully realized.

CHAPTER 2

SOLUTIONS AND APPROACHES

1. Solutions

1.(1). Language

Obviously the longest used and, at the present time, most important mode of communication used by psychologists is language. The majority of psychological theories, from psychotherapeutic techniques to speculations about the structure of the mind, are presented in words. By themselves, such theories are imprecise. Theories set down entirely in words are wide open to misinterpretation because the referents of the words are determined by the experience and training of the reader. For example, a lot of people would consider that Piaget has something important to say. But, except for broad agreement over general principles and experimental findings, psychologists may find some difficulty in agreeing about exactly what Piaget is saying. Such a problem varies depending on the lucidity of the individual theorist. Inhelder is aware of the problem and states: 'We are somewhat disturbed by the fact that the replication of our experiments does not always show a sufficient understanding of Piagetian theory on the part of the authors of these new works. We are of course the first to admit that such understanding is not easy to acquire, especially since this form of psychology is

closely linked to a certain form of epistemology ' (Forward to Modgil and Modgil, 1976, p.7).

One of the biggest problems with theories set down entirely in words 'is that the referents for the words are not externally objectified' (Reitman. 1965, p.8). The communicative aspects of words function by calling up ideas in the reader. Such ideas are entirely dependent on the experience and training of each individual approaching the written work. The problems involved in ensuring some relation between the reader's and the writer's ideas are awe-inspiring, and remain essentially unsolved.

An attempted solution to this dilemma has been the use of operational definitions, the tying of constructs to measures. The considerable interest in the strategy of operational definition which developed among psychologists a generation ago owed much of its impetus to the difficulties of building adequate theories entirely of words. Many at the time seemed to think it reasonable to hope that the foundations of psychology might be made both precise and communicable by tying concepts to standardized measures and manipulations. The approach has been successful in some areas, for example, operational constructs of intelligence, but in other areas, for example, measures of personality, the approach has been less successful.

At best however such an approach enables us to objectify presence, absence, and amount of whatever it is we are dealing with. By tying the acceptability of a concept to the immediate availability of an operational measure for it, there is no room left for unattached theorizing. While this approach has produced some useful measures, it leads to the eventual stopping of any real progress. In recent years, psychology has become less dependent on immediate operationality, and there is more interest in the structure and implications of theories cast in more formal languages. Such solutions will be considered in the following sections.

1.(2). Pictorial

The limitations of a purely language-based theory

presentation have now been considered. Other forms of theory presentation, such as pictures or diagrams, can provide common referents for relatively simple concepts and propositions, but they are of little help in clarifying a theorist's ideas (for himself or others) when related to complex structures or changes over time. Lewin (1936) is an example of a theorist who used pictures and diagrams to illustrate and present his theories, as well as using extensive mathematical presentations.

Miller (1964) dismisses Lewin's mathematical work as "merely discursive", and as an enrichment of English syntax. He suggests further that readers who approach Lewin's work expecting a mathematical theory will be disappointed. This appears to have occurred with Estes et al (1954, pp.341 - 342), who critized Lewin's work as follows:

> Where most learning theorists consider successful predictions of behavior to be the principle test of a good theory, Lewin's system requires only an "adequate", according to the theorist, description and symbolic representation of a situation.... By carrying over into psychology some of the verbal phraseology of physical field theories without the mathematics, Lewin has given us a facsimile of a field theory which resembles the real article in much the same way that a masterpiece of taxidermy resembles a live animal.

Lewin did not however, regard his mathematics as something to be "tested" by experiments; it was more a convenient language for expressing his highly spatial, Gestalt conception of psychological and social processes. Such a discursive method is not entirely useless and can lend a certain brevity to an account, and can frequently reveal analogies that may be missed without such a linguistic transformation.

1.(3). Mathematical

The attempt to apply mathematical models to psychological problems has a long history, a history which can be extended back at least to Bernoulli's paper in 1738 (in

Miller, 1964) and if one wishes to push the argument, even back to Aristotle. Miller (1964) gives four relatively distinct uses of mathematics in psychology. These are the discursive, normative, functional and structural uses. The discursive mode uses mathematical notation as a convenient extension of natural language. Normative applications use mathematics to discover or to demonstrate the most efficient way to attain certain goals, which unfortunately (from the point of view of the psychological theorizer) do not always coincide with what humans actually do. Functional descriptions are widely used in the physical sciences in stating the so-called "laws of nature" and this has perhaps been the most widely used type of mathematical presentation in psychology to the present day. The final use, that of structural description, is of most relevance to the course we are pursuing here, as will be seen in the chapters on Piagetian theory. Miller (1964, p.6) describes it as 'the use of mathematical notation to represent the reduction in degrees of freedom that must occur in any structured domain.'

From a psychological viewpoint, a structure is seldom obvious or available for inspection. Nevertheless, many theorists have insisted that the structure is there, and have tried to use mathematics to characterize it. The structure of the observed phenomena is generally analogous to the way that the mathematical symbols are related. Usually a system of symbols is constructed or borrowed whose structure is isomorphic with the structure discovered by the psychologist in his empirical investigations. Miller (1964) delineates two distinct levels of structure involved in a psychological theory. The subject constructs (often tactily) a model to guide his behavior, and then the psychologists constructs his model of what the subject has done. 'One task for the theorist is to make explicit the model that the subject is using; another is to construct a model that accounts for why and how he tries to use it. Thus the completed theory may contain two very different kinds of structure' (Miller, 1964, p.224). Mathematical

presentations of structures can be considered in spatial terms (as for example in a factor analysis) or in non-spatial terms, as Piaget has used.

2. Approaches

2.(1). Computer-based

The invention of the computer has lead to a host of technologies which are directly relevant to the problem of presenting complex psychological theories. I shall look at two important computer-based technologies, that of the information-processing approach, and the use of computer simulation techniques to produce models.

2.(1).(a) Information-processing approaches to psychology

From a psychological viewpoint, the stimulus is treated as information that is operated on, or processed, in different ways at different stages within the organism. In the course of this information processing it is assumed that changes take place in the nature of the stimulus information. Reitman (1965) considers information-processing in psychology as a 'label for a general approach to the study of psychological activity' (1965, p.1). This approach emphasizes the functional properties of thought and the things that this thought achieves. It encourages us to investigate in detail the systems of cognitive structures, elementary psychological processes, and higher order strategies involved in human behavior.

(i) Advantages

Perhaps the main advantage in such an approach to the study of cognitive processes is the degree of precision such an approach entails. The methods and research techniques developed in conjunction with these theories enable the psychologist to set down his ideas about cognitive processes as precisely as he chooses. It should be accepted that, if psychology is to take advantage of this chance to become a precise science, these ideas must be presented precisely. Before the development of these information-processing theories, the only tools of

communication available to psychologists apart from mathematics, were those ancient vehicles of words and pictures. Earlier sections have considered the limitations of these to the presentation of theories.

Hebb's theories could be considered as beyond the modes of communication that he had available when he first presented them. It is now possible, because of information-processing techniques, to carry out an analysis of cognitive processes at a level beyond anything possible at the time Hebb worked out his theory. As Reitman (1965, p.9) states: 'Information-processing techniques enable us to set down our ideas about central processes and structures objectively, precisely, and in formal languages dealing directly with association structures and the manipulation of complex relations among symbols.'

Furthermore, information-processing approaches have uses in direct experimentation as well as in purely theoretic considerations. If unexpected results occur in an experiment, one is unable to say whether the difficulty is in the theory, the ancillary assumptions, or both. In an information-processing model, one can state, manipulate, and deduce implications from theories in a way that is at once sure, unambiguous, and independent of operations such as measures, manipulations and conditions relating the theory to data on human behavior.

(ii) Disadvantages

Before it is concluded that I am offering information-processing approaches as a panacea to cure all our theoretical ills, it should be pointed out that there are a number of disadvantages to such an approach. It may as yet turn out that certain human information-processing functions or complexes of functions cannot be modeled successfully in a computer language. In this case, it would appear that such an approach may be concerned with a relatively special and limited conception of human cognitive activity. Another problem may be that computers are too "good" to model human processes exactly. Unless we can find a way to program a computer to overlook certain (perhaps relevant)

data, be distracted easily, show symptoms of fatigue and other human traits, any models we produce may remain rather stylized. Some more basic problems are rather overwhelming. For instance, the sheer volume of a computer program designed to undertake complex behavior is a deterrent to such an approach. For example, a version of Newell, Shaw and Simon's General Problem Solver covers more than one hundred pages (Newell, 1962; In Reitman, 1965). This leads to a greater distance of communication between the information-processing model and the reader. A further problem is the verification of a model's isomorphism with the system it is trying to simulate.

Lamendella (1975) distinguishes three different levels of theories which could be considered to be modeling some aspect of an organism. 'Each of these three subtypes of theory have different goals, are constructed on the basis of different evidence, and model the organism in a different sense' (Lamendella, 1975, p.8). The "lowest" level is the behaviorally isomorphic theory, which at a minimum has to model the observed behavior. The internal processes used to obtain such behavior are of no importance. All that matters is the degree of isomorphism between the observed and modelled behavior. Lamendella states: 'I doubt that any one would want to claim that the structures and processes of a behaviorally isomorphic theory of man would constitute a theory of human cognitive processes' (1975, p.10). The next level, which is a subset of behaviorally isomorphic theories, are the psychologically isomorphic theories, in which the functional components of such a theory are equivalent to the functions actually involved when a person performs a certain behavior. As well as this, the functional components are interrelated in the same way as the functional components of a person's cognitive system are related. Lamendella states a problem remaining in this level of model building:

> The internal structure of the functional components of a psychological isomorphism theory have no constraints other than their capacity for performing

> the function. Even if the theory has one component corresponding to every one psychological component (assuming a one-to-one isomorphism were desired), the internal operations of the components could not hope to reflect the way people process information while ignoring the physiological facts entirely. (Lamendella, 1975, p.11)

This leads into the final level of theory building. Lamendella also believes that cognitive psychology may end up as an 'unstructured heap of superficial, unrelatable experimental data' (1975, p.12). He believes we need to incorporate physiological facts into the mainstream of psychology, leading to a psycho-physically isomorphic type of theory, containing the insights of physical description theories and psychologically isomorphic theories. With the construction of information-processing theories, the researcher needs to examine what sort of model he is interested in, and to examine his complete model to see what it is he has modeled.

Despite these difficulties, the possible benefits of such an approach are large, and this approach should be pursued. Of course, information-processing models can be used carelessly and imprecisely, but, if used intelligently can aid a researcher to develop and communicate his theory of complex cognitive processes with elegance and precision.

2.(1).(b) Computer simulation

The first computer-based technology which has been considered is that of the information-processing approach. This approach contains within it an underlying assumption of the structure of the human brain, namely that it is an information-processing mechanism. The second technology to be considered could be described as broader-based, in that it does not have this underlying assumption. This is the technique of computer simulation. Barton (1970, p.5) defines simulation as 'the dynamic execution or manipulation of a model of an object system for some purpose.' An object system is simply the system we wish

to study (i.e. the "real world"). Obviously in psychological research, this "real world" is not as clearly defined as it may be in other "harder" sciences. The considerations of Lamendella (1975), discussed in the previous section, are relevant here. We need to decide what level of the human organism we are attempting to model, whether the objective observable behavior, or a more hypothetical cognitive structure. DeGreene (1976, p.119) describes this problem when he states: 'An analog based on only two variables - input and output - has the lowest degree of isomorphism, and the underlying function may be vastly dissimiliar. Such an oversimplified model is most useful as a representation of a subsystem, which is then linked to other simplified subsystems. The lack of precision and detail should not transcend the subsystems.' Simulations are basically analogies. They can range from physical operating devices with definite shapes to block diagrams, figures, and computer programs.

We need at this point to make a distinction between the uses of the terms "theory" and "model", Barton (1970, p.26) defines a model as 'a constructed specific expression of a theory of one or more hypotheses.' This relationship is shown in Figure 1.

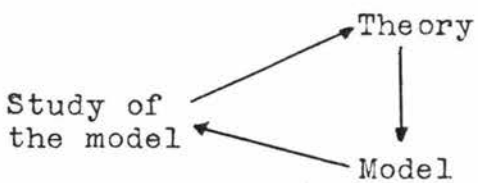

Figure 1. Relationship between theory and model. (From Barton, 1970, p.27)

A theory can be considered as a set of general or abstract principles as they exist at a particular moment in time. Theories are purposely broad, as they try to apply over many specific cases, and communicate generalities. A given theory may lead to almost an unlimited number of different models and applications. Figure 2 shows the relationship between theories, models, and the object system.

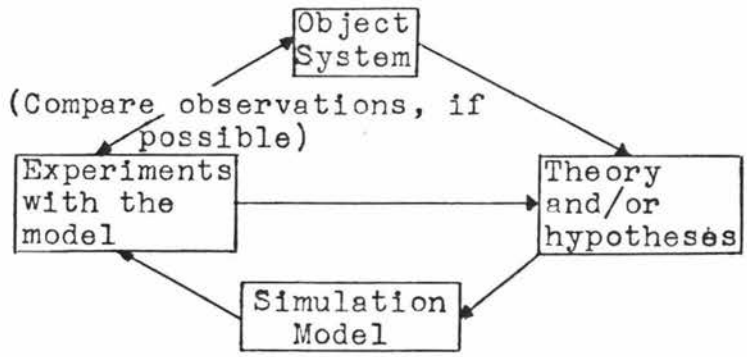

Figure 2. Relationship between model and object system. (From Barton, 1970, p.28)

A more elaborate model of this relationship can be seen in Figure 3. In discussing this Haggett (1976) considers that the use of the term model has at least three different uses in everyday language. These are as a noun, implying a representation; as an adjective, describing an ideal; and as a verb, used to demonstrate something. Ackoff, Gupta and Minas (1962) have suggested that we incorporate part of all three meanings when transferring to scientific usage. In model building we are creating an idealized representation of reality in order to demonstrate certain of its properties.

Realistic models can be constructed fairly easily for physical systems, but not for complete biological, psychological and social systems, in which real system relationships are obscure, the actual number of variables may not be known, and quantification poses a formidable problem. Modeling and simulation have an advantage for these systems however, in that they allow detachment of the observer from the system he is studying and therefore reduce personal bias. A model is made necessary by the shear complexity of reality. It conveys not the whole truth about the real world, but a useful and comprehensible part of it.

2.(2). Systems Science

A new approach, the so-called systems approach, is being considered in all areas of science. A system can be considered as an arbitrarily demarcated section of the real

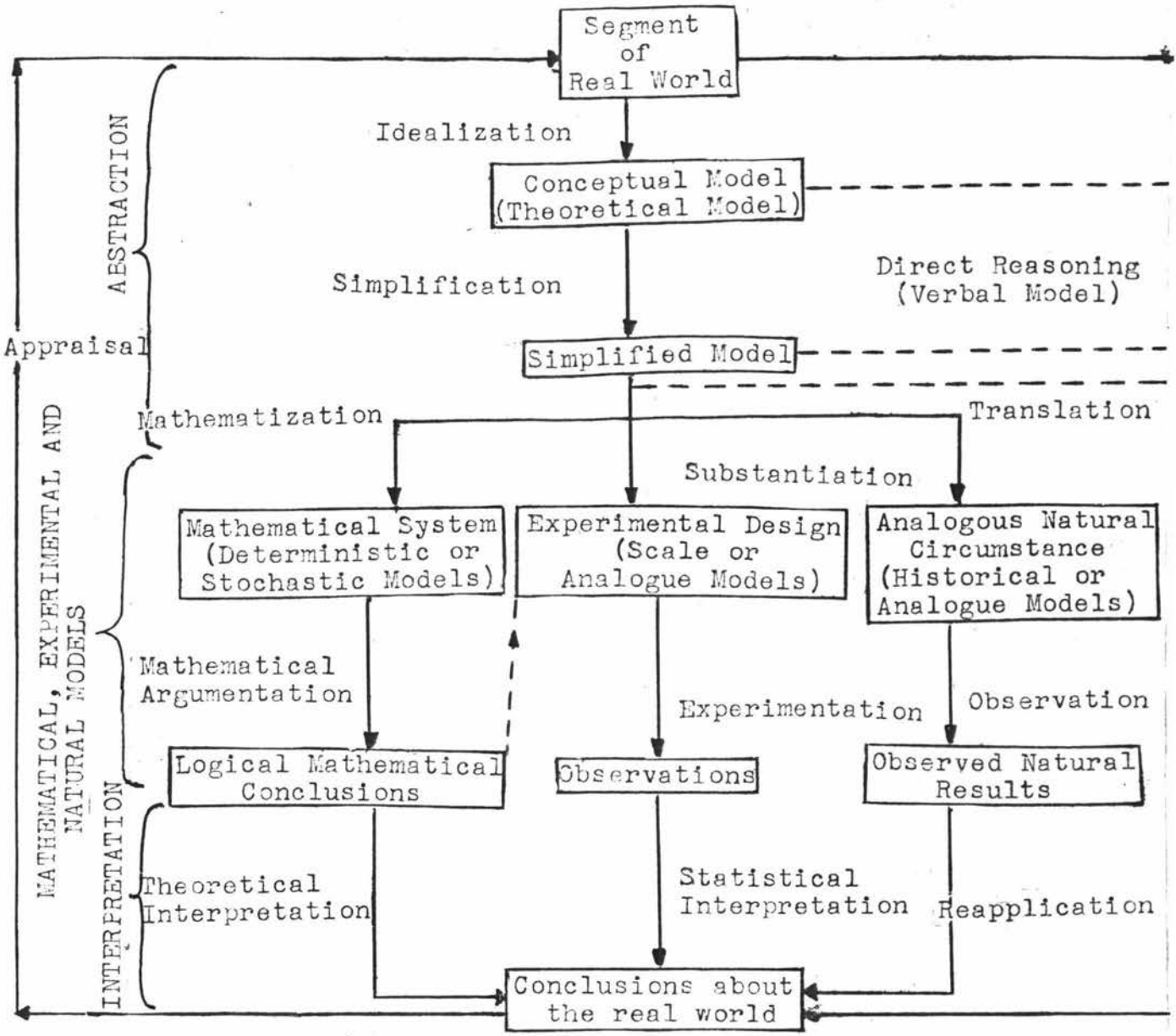

Figure 3. A model for models.
(From Haggett, 1976 p.316)

world which has some common functional connections. Because this approach has been used in a wide variety of disciplines, people coming from different backgrounds and bases have interpreted the concepts in different, or even contradictory ways. In general terms the systems viewpoint means looking at situations, topics, and problems as a complex of interacting parts which can be divided into systems, sub-systems, and sub-sub-systems. Identification of these various systems leads to an examination of the relationship between them. Specifically the flows of

influences, energy or information among and within the system involved and the routes these take, can be studied. 'Systems thinking and the systems approach is now a growth area' (Beishon & Peters, 1976, p.8). Ideas from this approach appear in different guises in cybernetics, systems engineering, general systems theory, operations research, systems analysis, computer systems, and many other fields.

As considered earlier, it seems we need a general approach to psychology to make sense of the great wealth of data that has been accumulated. DeGreene (1976) believes that systems theory provides such an approach. He states: 'This approach could serve as a launching point for a comprehensive theory of human behavior.... This theory must incorporate the input-throughput-output paradigm, and it must be expressible in terms of the modification or processing of inputs, probably in terms of information processing' (DeGreene, 1976, p.131). He delineates two main ways of looking at systems, generalization across systems, and simulation. I have already considered simulation in some detail. As for generalization, it is a common basic principle of systems theory that different systems may have much in common. 'It is a striking phenomenon, the significance of which is hardly understood, that laws which are structurally identical or isomorphic apply to fundamentally different fields' (Von Bertalanffy, 1951, p.305).

DeGreene (1976) outlines some properties of systems. A system is usually in equilibrium, and such an equilibrium is dynamic, not static; there is evidence of change over time, in the form of growth or decay; one subsystem may play a dominant role in the behavior of the system, leading to dominance or centralization; many systems are open, exchanging information, energy, or materials with their environment; and finally there is a process of differentiation, the formation over time of new constituents from old. Reading these properties, one is struck by the similarity between this conceptualization and Piaget's interpretation of the child and his behavior. This point will be pursued in depth in a later section.

CHAPTER 3

METHODOLOGY

1. Computer Simulation

We now need to consider the methodology of developing a simulation program. This could be divided into three broad basic steps. They are firstly, data collection, secondly a development and programming of the model, and finally, an analysis of the results.

1.(1). Data collection

In the initial stage dealing with data collection, there are a number of aspects not dealing specifically with the gathering of data. We need initially to determine the objectives of the simulation. We commence by examining a particular behavioral process and making observations of the phenomena. The actual operating realities of the system need to be considered, through these initial phases. These are, the representation the Ss use of the environment, a description of the general categories of behavior, and an identification of the rules or heuristics which interact with the environmental representation to make up a behavioral process.

'If a computer program is to simulate human behavior, it must have available an environment to work upon' (Feldman, 1971, p.185 - 186). We could ascertain the nature of the environment either by examining the S's references to it, or by generating an ideal environment that we think

the S would require to engage in his behavior. We need to consider the elements of the environment that the S feels are relevant to his task. As well as this, we need to consider what labels the S attaches to them. Types of relationships that the S forms between the objects in the environment also need to be considered. As well as spatial, the temporal relationships need to be considered - that is, do relationships between elements change over time?

Secondly, we need to consider qualitatively different kinds of behavior that the S exhibits. These categories of behavior can in general be described in terms of what the S says he is trying to do, i.e. his goals. These broad behavior patterns can define the framework within which the specific information processors are built. Not only observable aspects of behavior, but also cognitive behavior needs to be considered here.

Finally, we need to try and establish what rules and heuristics the S uses to operate on the information he has received. This may require some work and consideration of a large aspect of the S's observed behavior.

There are a number of possible methods of proceeding when we are faced with the problem of data gathering. Some researchers indicate a preference for gathering large quantities of data before simulation is attempted, others for after the model is developed. There is, of course, a need to use the data in a balanced way for information before the development of the models and for verification afterwards. Overriding all this, the researcher needs to be aware during the whole data collection process of the basic question of whether a complete and detailed (to the point of being operational) model is warranted and desirable.

1.(2). Developing and programming the model

Constructing a model of complex thought processes involves the breaking down of these processes into simple elements. There are a considerable number of factors to be considered. Gullahorn and Gullahorn (1971, p.353)

state:

> In computer modeling, one must develop a symbolic representation of both the experimental situation and the theoretical system. Neither task is trivial. Researchers simulating human chess play, for example, must develop a means of representing the chessboard, the pieces, and the operating rules concerning legal moves. As a seperate and additional problem, they must program hypotheses concerning human strategies in playing.

The model initially can be designed conceptually, beginning with a general verbal and mathematical description which leads to the conception of a flow chart, showing the logic of the interactions and flow of information through the system. This flow chart need show only the elements and flows of interest. This production of a flow chart is a crucial part of the whole operation.

Once the flow chart has been constructed, we need to select a language and program the model. There are a number of criteria needed for the final choice of the programming language used (Sammet, 1969), including its availability on the computer used, its general form of construction, and its ease and use of interpretation, etc. Newell and Simon (1971, p.154) describe the process as 'try to write a computer program that you think would simulate the human protocol ... realize the program on a computer, and determine what behavior path it would follow when confronted with the same problem as the human subjects.' A model needs to actually operate because 'there is no guarantee that a simulation model is logically sufficient or accurate until it has actually been implemented' (Starbuck and Dutton, 1971, p.18). The process does not end with the initial programming as the program needs to be run, tested, and "debugged".

There are two types of programs which can be constructed. These are the descriptive, in which the objective is to state how behavior <u>does</u> occur given an <u>internal</u> criter-

ion, and the prescriptive, in which the objective is to state what behavior should occur given an external criterion (Dutton and Briggs, 1971).

At our present stage of knowledge, any model described in a running program will have both psychological and nonpsychological processes. A psychological process can be considered to parallel the process used by S in generating a certain output from a certain input. A nonpsychological process does not have this isomorphism. This is necessary because there are occasions when E is unable to identify what the S is doing, or even if this is possible sometimes lack of programming techniques make it difficult or even impossible to model the desired process (Laughery and Gregg, 1971). These processes help "fill out" the model and need to be distinguished from the psychological processes. The nonpsychological processes indicate an area of attack to E, both in his understanding of S and of programming in general. The overall goal of this model construction is expressed in this analogy by Newell and Simon (1971, p.154):

> To explain a behavior path by a program is quite analogous to explaining the path of a planetary system by a system of differential equations. The differential equations predict what will happen next ... as a function of the exact state of the system at the beginning of the interval. The program determines what the mechanism will do next as a function of its exact state at the moment, this state being dependent in turn, on the previous history of the system and its current environment.

1.(3). Analysis of results

The final step in this process is to test the theory by computer simulation. The program is executed on a computer and the machine's performance is compared with the S's performance. The E measures and analyses the outputs obtained from the simulation model. Results are often

verified by comparing decisions made by a computer with those made humanly, given the same input data. In order to provide an adequate test, the behavior with which the computer output is compared must, of course, be different from the behavior which served as the basis for the model's construction. What actual degree of isomorphism is required between the computer's and the S's behavior is still a matter for debate. Such considerations are beyond the scope of this paper, but they have been touched on briefly earlier (see page 11).

Finally, as a result of this comparison between the two types of behavior, the model is adjusted and reconstructed to allow for any discrepancies observed. This process is continued until the E is satisfied with the result (if ever).

2. Systems Science

There are many different approaches to a methodology for systems thinking to be found, probably almost as many as there are definitions of systems thinking itself. I shall try to present a reasonably general account here, in a 'precise yet vague' form (Checkland, 1976, p.51), which enables ideas and actions to emerge, but which does not reduce to no more than a technique. Many approaches concentrate on an initial generation of a large number of alternative descriptions of the system, which is followed by a decision on the most appropriate based on the criteria of the objectives and the need the system is to fulfil. Even within a single objective, many alternatives are possible, so other factors for making a decision need to be taken into account. Jenkins (1976) gives an approach which emerges from his definition of a system as a complex grouping of components for which there is an overall objective. He selects the system which will solve the problem, places it in a systems hierarchy and defines the required objectives. This chosen system is designed, implemented and then reappraised in operation.

Checkland (1976) gives a nine-stage structure to a

systems methodology. This is presented in Figure 4. Only the first four stages are of direct relevance to our needs and will be considered in more detail. The remaining stages are more concerned with the use of the methodology in the solving of real world problems (e.g. management) and shall only be looked at briefly.

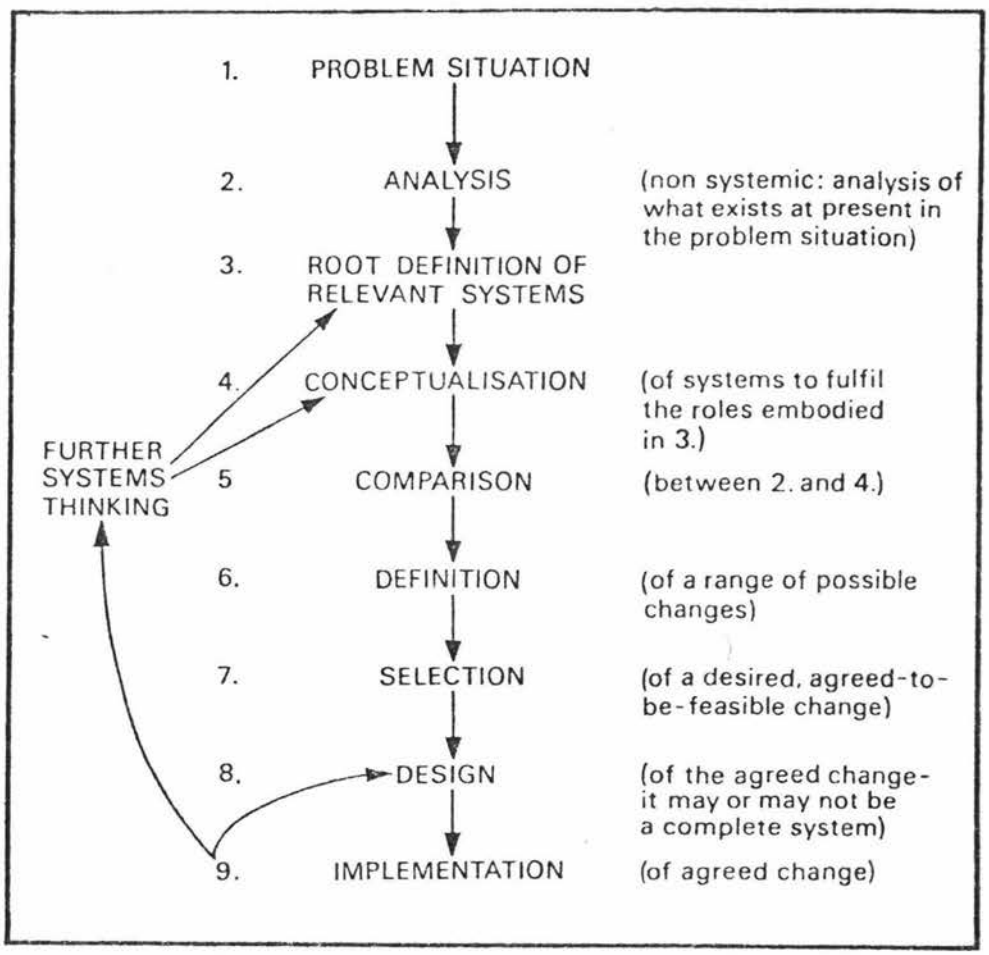

Figure 4. Systems science methodology.
(From Checkland, 1976, p.61)

2.(2). Analysis

The initial phase in the resolution of the problem situation is an analysis of it, a finding out of what actually exists at present. The situation is viewed as a hierarchy of systems. We try to build up a rich picture of the situation through three main aspects, structure, process and "climate". The structure is explained in terms of the actual physical design, the hierarchy developed, reporting structures, and patterns of communication. The processes are concerned with the basic activities, which

could be considered as planning, implementation, monitoring the effectiveness and external effects, and correcting deviations. The "climate" is the relationship between the structure and the process. Figure 5 outlines the guidelines for an analysis of the problem situation.

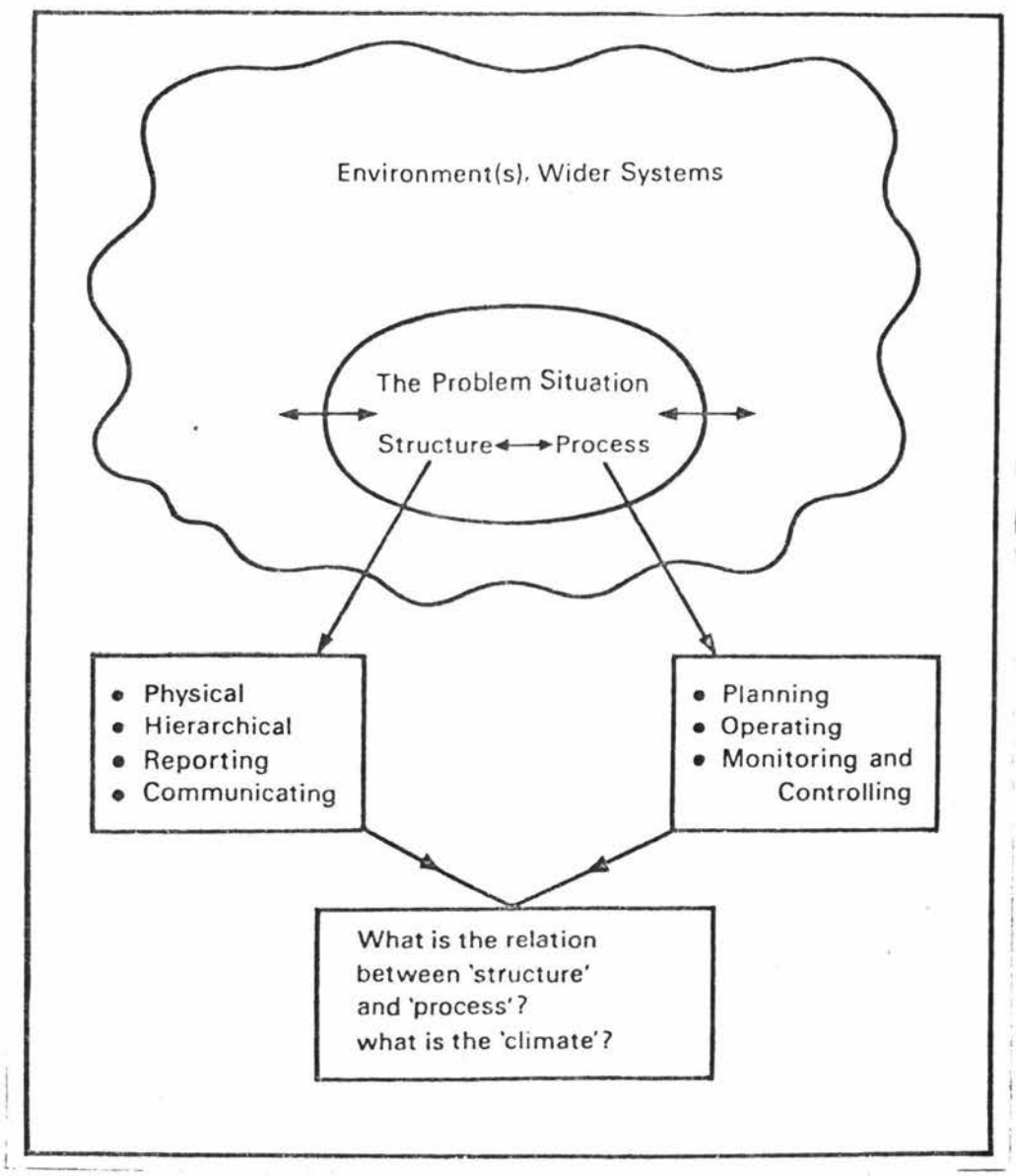

Figure 5. Guidelines for an analysis of the problem situation. (Checkland, 1976, p.62)

2.(2). Root definition of relevant systems

The root definition indicates the basic nature of the system or systems thought to be of relevance. It can be considered as a condensed representation of the system in its most fundamental form. It is only a part out of a

range of possibilities and can never be considered to be finally "correct". During the consideration of the problem situation, it may be worthwhile exploring the limitations of several root definitions. An example for a human system may be as an "information processing system", which leads to considerable implications. It is not the only root definition however. Something analogous to "self-actualizing systems" (Maslow, 1967, 1970) may also be posited, leading to different, but no less important, implications.

2.(3). Conceptualization

This leads into the actual building of conceptual models of the system which are relevant to the problem situation and for which root definitions have been derived. Something needs to be created which can be compared with the picture built up in the earlier analysis phase. This should be as objective a picture as possible of what is implied by the root definition. A way of building the model then needs to develop, perhaps by assembling the minimum number of activites required in a correct sequence. This model is then compared with a more formal system (if available) to ensure that the conceptualization is not deficient in some basic aspects.

2.(4). Comparison and definition of possible changes

A comparison may be performed at any number of levels between the conceptualization and what actually exists in the problem situation. It is done to reveal possible changes leading to an "improvement" of the problem situation. These changes are then defined and then certain are selected as being necessary and feasible.

2.(5). Design and implementation

Design refers here to the creation of some modification within the system, which is then implemented.

2.(6). Appraisal

Checkland describes this as 'a dignified way of saying "Start Again" '(1976, p.72). It emphasises the fact that the whole methodology is flexible and subject to continual assessment and change. Nothing should be considered as being in its "final form". More sophisticated root definitions and/or conceptualizations may emerge, resulting in a reconsideration of the problem. A summary of the whole methodological approach can be found in Figure 6.

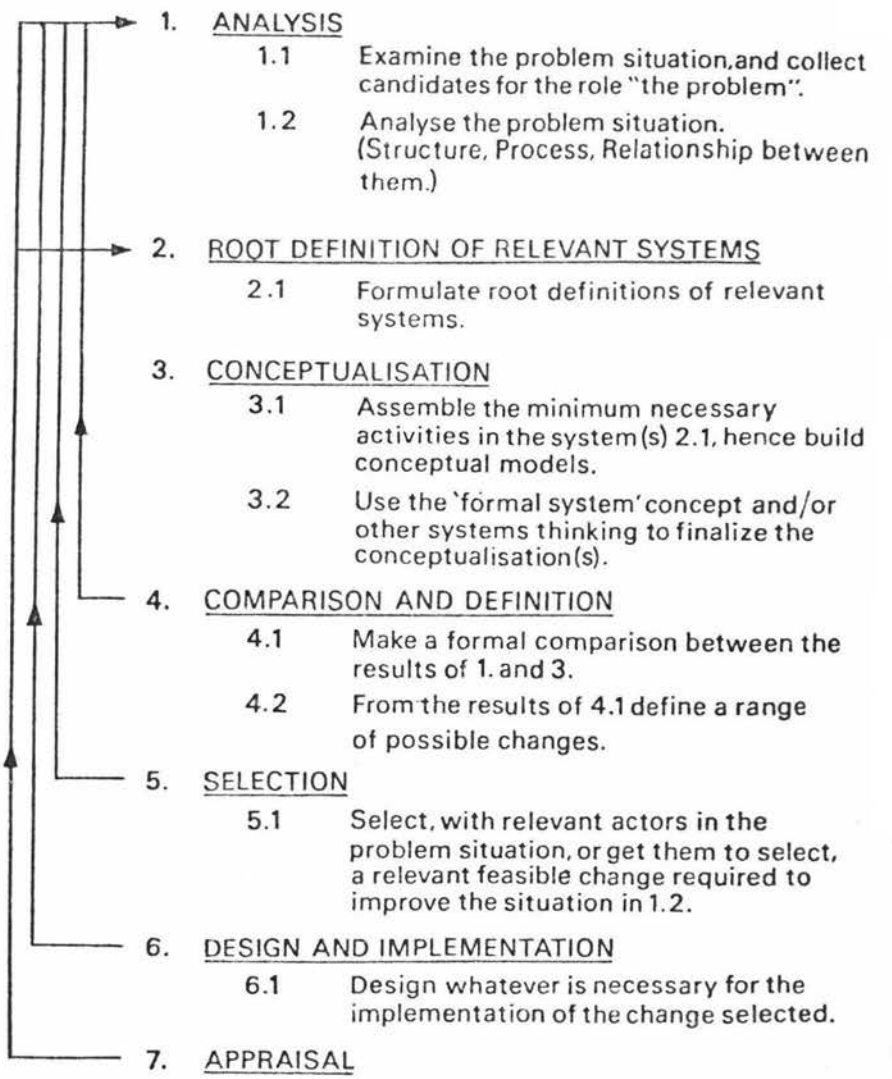

Figure 6. A summary of the systems methodology. (From Checkland, 1976, p.73)

3. Single Subject Performance Models

I now wish to look at a particular methodology for constructing performance models for single subjects. When the S is confronted with a certain situation, it is assumed that he deals with that situation by moving through a finite set of states of knowledge towards a predetermined goal. In order to move through this set of states of knowledge, the S sets certain goals and applies operators to reach those goals. At any point the choice of what to do next is dependent on the S's production system. The basic element of this system is the production.

Each production can be considered as a rule which has two parts, a condition part (C), and an action part (A). The rules are written in the form C $\longrightarrow$ A. The condition part refers to symbols in the short-term memory (STM) that represent goals and knowledge elements existing in the S's knowledge state (Klahr & Wallace, 1976). The condition part is evaluated by tests that can be applied to the states of knowledge. The right hand side of the expression is the action part, which is a set sequence of one or more operators. These operators consist of transformations on the STM, including the generation, interruption, and satisfaction of goals, the modification of existing elements, and the addition of new elements. The condition side of a production can be considered as a pattern of conditions of evocation, which when matched, evoke the appropriate action component.

The system operates in the control structure where each production is considered in turn. The current state of knowledge is matched against the pattern on the left side of the C $\longrightarrow$ A pair (the production). If the condition is satisfied for the current state, then the associated action on the right side is then evoked and performed, which transforms the state of knowledge by some change which occurs in immediate memory. The process then begins anew at the top by trying to match each new state of knowledge with the first C $\longrightarrow$ A pair, and the process continues. If no condition is satisfied, the processing

terminates unless some new information is obtained. The immediate memory has a goal stack as well as a collection of elementary assertions about symbols. Knowledge of paths followed and knowledge states to which return is possible are not represented. Khlar (1973, p.152) has described the simple operation of the system through a series of rules. This is presented in Figure 7 in the form of a flowchart.

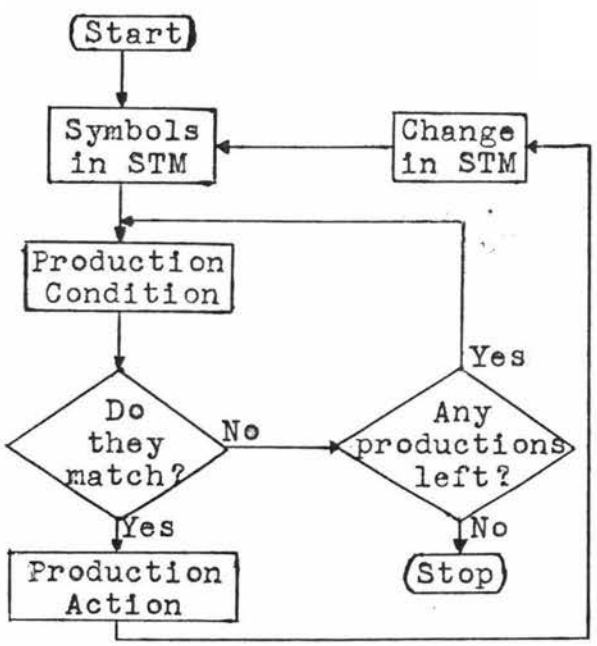

Figure 7. Operating rules of a production system. (Created from information in Klahr (1973) and Klahr & Wallace (1976))

It is important to note the hierarchical nature of the processes. As the system works through a series of production rules in turn, a hierarchy develops within the model. The importance of this type of hierarchy is that it is not fixed in advance, but grows in response to the interplay with the environment, and shows some of the

flexibility and transferability that seems to characterize human higher mental processes.

An ordered set of these productions put together is called a production system. In actual use a model is posed in the form of a production system. The system 'forms a complete process if it is iteratively applied to each new knowledge state that is generated by this action' (Waterman & Newell, 1971, p.289). The model is written as a series of conditional statements each expressing what to do under specified conditions. The act of executing the various tests on these productions is implicit. In any case where more than one condition is true, the one higher in the list takes priority (Newell & Simon, 1972). The model is used to predict 'which operators are to be performed at each step, hence what knowledge state is to be attained at each step, and thus what things relevant to the task can be exhibited by the subject at each point' (Newell, 1973a, p.33). Production systems constitute a family of computational and logical systems much studied in computer science (see Minsky, 1967; Hopcraft & Ullman, 1969). Members differ considerably in the details of the conditions, actions, control structures and the data types on which they work. Concepts equivalent to production systems occur in symbolic logic (Post, 1943) and linguistics (Chomsky, 1957).

The exact size of the production system does not seem to be an important descriptive parameter. It is certainly important that it be small enough to allow it to be discovered and analyzed. However, additional productions come into use and into view with every new aspect of significant variety in the task environment. The point at which analysis stops is dictated by the frequency with which particular productions are evoked. The S undoubtedly has other productions in long term memory (LTM) which are not always evoked when he is concerned with a particular problem space. A relatively small production system is permitted by closure of the problem space. This closure occurs by restricting the area considered to one specific task or aspect of behavior.

It is worth making the point here that, in addition to considering production systems as a methodology for constructing information-processing models of human behavior, Newell and Simon (1972) also consider the concept of production systems as a theory of the actual structure of the human information-processor. They state (1972, pp.803 - 804):

> We confess to a strong premonition that the actual organization of human programs closely resembles the production system organization .... We cannot yet prove the correctness of this judgement, and we suspect that the ultimate verification may depend on this organization's proving relatively satisfactory in many different small ways, no one of them decisive.

In support of this claim they outline a number of features, some of which are briefly summarized below. The full set with detailed explanations can be found in Newell and Simon (1972, pp.804 - 806). The features include: the information about an information processing system (IPS) encoded in a production system is homogeneous; each production is independent; a production has a strong stimulus-response flavor; productions seem to represent meaningful components of a total psychological process and not just odd fragments; the working memory for a production system is the STM which fits the functional definition; a production system may offer a viable model of the LTM.

Of importance during the development of a model is the choice of language that is used to report the model under consideration. While Klahr and Wallace (1976) state their models in the language PSG (created by Newell, 1972), it is worth noting that this is not the only representation for stating production systems and that even a specific language such as PSG allows wide latitude for the final configuration of specific assumptions about the human IPS. Two recent studies of the development of seriation skills in young children have used production

systems in quite distinct forms. Both Young (1973, in Klahr & Wallace, 1976) and Baylor and Gascon (1974) (considered in more detail in Part II) used production systems to model different levels of performance on seriation tasks; Baylor and Gascon developed a production system language specifically designed for seriation models (Gascon & Baylor, 1973). Both of these approaches are reviewed by Klahr (1976). A paper by Newell (1973b) provides a basic introduction to production systems and to the use of PSG, as well as showing a fairly complete production system model of the mental scanning work of Sternberg (1970). An extensive general treatment of production systems, independent of any programming language, can be found in Newell and Simon (1972). Yet another production system representation and language is utilized by Waterman's (1974) adaptive production systems.

The production system format seems to have developed as the preferred form for specifying an information-processing system. Newell and Simon (1972) have considered this format as one of the most important features of the information-processing approach to psychology since its inception in the early 1950's.

4. Conclusion

The three areas we have considered show some overlap of concepts and approaches. In particular, the representation of the environment by Ss discussed in the computer simulation section, and the problem space of the single subject performance models may have rather a lot in common. It also may well be that the simple elements discussed in the former section have been identified as the productions of the latter section. Other points need to be made. In the systems science approach the need for a large number of approaches to be considered gives a direction to any specific experimental research, with a view to determining which behavior is the most isomorphic with the actual behavior being studied. At a more basic level, any number

of root definitions (see page 23) can be posited for man, and it is here that his complexity lies, as any one of them may give some indication of his nature, but none on their own could be considered as a complete description. On a more immediately useful level, it may well turn out that a system which involves production rules will have within it a hierarchy of subsystems, with some higher order productions leading to whole production subsystems which are introduced and executed whenever a particular environment or type of information is recieved.

When Piaget's so-called clinical method is considered more closely, there is some similarity between his careful and extensive exploring of a Ss thought processes and the intensive studies undertaken when one S's behavior is recorded and simulated through the use of information-processing models. Piaget developed his clinical method from his work at the Binet laboratory in Paris when he decided it was useful for the study of thought. His work is structured in order to isolate variables of thought processes, aiming to learn precisely how one individual thinks about any one problem. Having gathered all the information he can from one individual, he relates his findings to his theoretical structure. The clinical method can be considered as a general framework for questioning the S, which allows for flexibility in the way the E questions the S in order to determine what the S is thinking. This questioning is very delicate and subject to several kinds of error. The E may talk too much and thereby suggest answers to the child. Or, the E may not talk enough and fail to pose the questions necessary for determining the child's meaning. As well as this, many non-verbal tests need to be incorporated into the interview to allow for the child's non-verbal thinking. Any verbal work required always refers to actual concrete objects that the S has before him, or answers may be obtained in the form of manipulations of these concrete objects. Braine (1962) has critized Piaget's clinical method as being too verbal and therefore inadequate. However, many of Piaget's

earlier tentative findings have been substantiated in more carefully controlled experiments. 'Piaget proceeds with the assumption that a detailed investigation of any small sample of a species will yield basic information inherent to all members of that species' (Maier, 1965, p.82), and it would appear that this assumption has been validated.

Piaget has explicated his eclectic methods of research and theory construction. He proposes two essential approaches, much akin to those utilized in the nuclear sciences (Piaget, 1954). They are, firstly, the step-by-step analysis based upon an investigation of cause and effect, which forms a network characterized by hierarchical relationships and combined connections; and secondly, the analysis of implications by considering both the field as a whole and the coordination of its parts. In mathematics this is comparable to the group, and in logic, to the propositional operation. Emperical investigations and the inductive mode of reasoning is akin to the first approach, while reasoning by logic and constructing hypotheses by deduction is similar to the second. Symbolic logic, Piaget stresses, can be applied scientifically as a research tool as can the more commonly used statistical techniques.

Piaget's concentration on specific aspects of the child's development at different times can be considered in terms of the closure of the problem space (e.g. conservation, seriation, classification, etc.). Within each of these spaces, he has intensively and fairly exhaustively examined the process and its parameters. Researchers working on an information-processing explanation of the processes also tend to concentrate on one problem space. Baylor's studies with others looked at seriation (Baylor and Gascon, 1974; Baylor & Lemoyne, 1975); and Klahr and Wallace's looked at conservation (Klahr & Wallace, 1970, 1973).

Finally, it would be of some value if a single computer language could be devised which would enable researchers to present their diverse models in a single format, with a view to some form of integration of these problem spaces. A lot of fundamental questions, such as the representation used

by the Ss in solving these problem spaces would need to be answered however before much agreement on a common language could be reached. Boulding's (1956) conception of the "image" as the base that behavior is dependent on may be a useful focussing tool for the problem. The constructing of such a language may well be another focussing tool for looking more closely at the whole range of human behavior and describing it in more explicit terms.

PART II

THE PERFORMANCE MODELS OF A NUMBER OF INDIVIDUAL SUBJECTS

CHAPTER 4

EXPERIMENTAL STUDIES: PIAGET'S WORK

The groupings of logical operations are important structures in Piagetian theory - they are the structural centerpiece of the concrete-operational period. Piaget asserts that these logico-mathematical structures are intended to model thought structures.. Unfortunately Piaget does not give any explicit explanation of exactly how these structures can come to be translated into actual human behavior. Most researchers seem to bypass the model when it comes to actual experimental approaches. 'When, however, one looks at a child's actual performance on one of Piaget's tasks, the structural (logical) descriptions turn out to be of little use in accounting for the detailed sequence of steps the child goes through in doing the task, whether he solves it correctly or not' (Baylor & Gascon, 1974, p.2).

Flavell considers the structural model as a 'general context or frame of reference for looking at behavior, rather than a precise and detailed statement of model-to-behavior isomorphisms' (1963, p.189). There appears to be three main purposes in Piaget's groupings (Flavell, 1963, p.190). They are firstly a precise structural description of "ideal" cognition in certain logical operations, secondly they form a framework for considering global qualities of the pre-operational stage, and thirdly they can be used to

investigate more specific intellectual attainments.

1. Grouping V

In this discussion, I shall look specifically at Grouping V (addition of assymmetrical relations), of which the core operation is seriation. Grouping V is concerned with asymmetrical relations, in which some ordering occurs. This order is transitive (i.e. A $<$ B and B $<$ C implies A $<$ C), and occurs in a specific direction (i.e. A $<$ B, $\neq$ A $>$ B). The important aspects of this grouping may be set out formally as follows (from Piaget, 1949):

Let O, A, B, C, D ... be a collection of entities.

Let $\longrightarrow$ be a transitive asymmetrical relation.

Let a, a', b', c' ... label the ordered differences indicated by $\longrightarrow$.

We can therefore form a series:

$$O \xrightarrow{a} A \xrightarrow{a'} B \xrightarrow{b'} C \xrightarrow{c'} D \quad \ldots\ldots$$

$$\xrightarrow{\quad b \quad}$$

$$\xrightarrow{\quad\quad c \quad\quad}$$

$$\xrightarrow{\quad\quad\quad d \quad\quad\quad}$$

The series has the following properties:

1. Transitivity

   e.g. $(O \xrightarrow{a} A) + (A \xrightarrow{a'} B) \equiv (O \xrightarrow{b} B)$

2. Composition (closure)

   e.g. $(O \xrightarrow{a} A) + (A \xrightarrow{a'} B) + (B \xrightarrow{b'} C) = (O \xrightarrow{c} C)$

3. Associativity

   e.g. $(a + a') + b' = a + (a' + b')$

4. Special Identities

   a. tautology $a + a = a$

   b. resorption $a + b = b$

5. Reversibility

   The inverse of an ordered difference relation $(A \xrightarrow{a'} B)$ is the reciprocal $(B \xrightarrow{a'} A)$.

6. General Identity - equivalence relation

e.g. $\xrightarrow{a} + \xleftarrow{a} = \xrightarrow{o}$

2. Characteristics of Stages

Piaget (1977) outlines a series of levels in the development of the ability to seriate. Level I is divided into three sections. At level IA(i), the child simply arranges the objects in twos (e.g. one big and one small) and cannot coordinate the pairs. At level IA(ii), the same thing occurs but in threes or fours instead of twos. Level IB results in construction of the series but only after considerable trial and error and subsequent corrections. A child however at level IIA manages the series immediately by looking for the smallest (or largest) of the elements, then the smallest (or largest) of the remainder, and so on.

Piaget (1977) distinguishes between the child's action in performing the behavior, and the child's conceptualization of it. For Piaget the action precedes the conceptualization, or at best they proceed simultaneously. The child's consciousness (cognizance, in Piaget's terminology) of his conceptualization becomes progressively more developed. In level IA this amounts to an awareness of dichotomies (big and little) and trichotomies (big, middle and little), which remain at the same level as the actual action, and which in the early stages amount to arranging in pairs and later arranging in threes and fours. At this stage the specific actions dominate, and the child's attention is concentrated on the results of the action.

At level IB, the child's conceptualization is at the level of "labelling". This relates to the particular individual objects without any explanation of the relationships among them, or coordination of specific actions. The child is likely to use such labels as "a big-middle sized one" etc. Actions at this level usually result in the construction of an actual series, but only after trial and error, that is, when the child actually sees his mistake.

At level IIB, the coordination between the actions dominate, and the child uses a systematic method to complete the series. The child must simultaneously select elements smaller than the preceding one, and larger than the following one, or vice versa. The child's conceptualization now includes the relationship between the actions, with consideration being paid to internal and central mechanisms, to the coordinations themselves.

3. Piaget's Work

The number of available studies in the area of seriation seem to be relatively small (Modgil, 1974; Modgil & Modgil, 1976). The Geneva studies have demonstrated that children below the age of four or five do not have the concept of seriation even when presented with a small number of objects. Piaget and Szeminska (1941, in Inhelder and Piaget, 1964) studied the development of length seriation initially and found the stages outlined above. (See also Piaget, 1952). They used ten small rods with others added later for insertion in the original series. Inhelder and Piaget state that 'children do not use the configuration of the whole until a comparatively late age' (1964, p.248). Younger children (5 to 7 years) need to compare two differences by measuring one pair against the other. Older children (9 to 10 years) can tell at once whether the differences are equal or unequal by taking the configuration of the series into account (i.e. whether the differences are equal, or if there is some relationship between each member of the series, resulting in either a positively or negatively accelerating parabola across the top of the sticks). Inhelder and Piaget (1964) also consider the difference between visual and tactile perception of the objects to be seriated. They found no great difference between the two modes of presentation. The other major area they considered was that of obtaining the children's "anticipations" of what the series would be like, before actually constructing it, by drawing the expected series.

There were three possible responses. Firstly, failure in anticipation, secondly a global anticipation, in which there was a correct pencil drawing or a correct seriation of size in a coloured drawing, but there was no correspondence between the colours drawn and that of the real objects. Finally, an analytic anticipation was obtained, in which there was an exact correspondence of colours and sizes of the drawing and the objects, and a correct seriation of the drawing sizes. The results obtained are presented in Figure 8, which is drawn from results published in Inhelder and Piaget (1964).

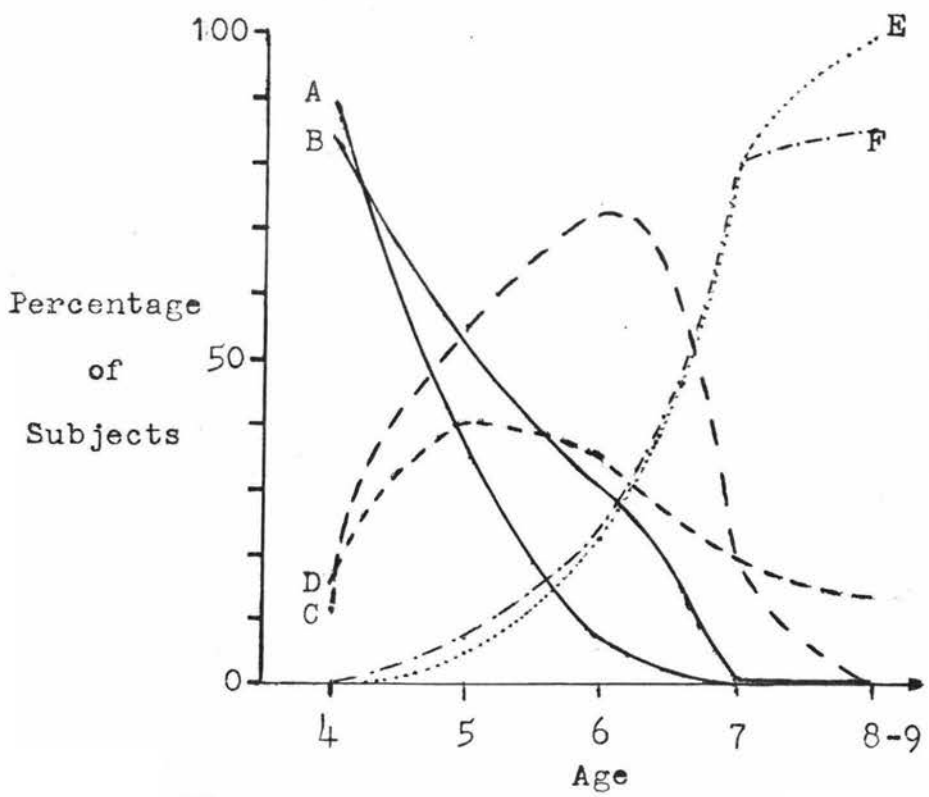

Figure 8. Anticipation and performance in seriation.
(Constructed from data in Inhelder & Piaget, 1964, p.253)

A - Failure in anticipation.
C - Global anticipation.
E - Analytic anticipation.

B - Failure in seriation.
D - Success by trial and error.
F - Operational seriation.

Piaget's solution to the apparent paradox of the child being able to anticipate the seriation before being able to perform it is quite simple. An actual performance of seriation implies reversibility - the S must be able to realize that, for example, an element E is longer than the elements already in the series (D, C) at the same time as it is shorter than the elements remaining (F, G). When drawing a child needs only to consider one direction of the size variable, and does not have to compare pairs. 'The relation he imposes is one-directional and therefore irreversible. The anticipation expressed in these drawings is an incomplete anticipatory schema because it is pre-operational' (Inhelder and Piaget, 1964, p.257). It would appear that these anticipations are a different phenomena to the conceptualization (discussed earlier), which appears to follow the development of action.

CHAPTER 5

EXPERIMENTAL STUDIES: LENGTH SERIATION

One of the earliest non-Piagetian studies of length seriation was that of Braine (1959), which included an experiment on transitivity of length using a largely non-verbal testing procedure. The Ss were rewarded with candy for correctly choosing the longer (or shorter) of two uprights. After observing that A > B and B > C, they were tested for their ability to infer that A > C, where the difference was imperceptible. Braine (1959) in this way distinguished between the use of perception (i.e. can see A > C) and the use of inference (i.e. A > B and B > C implies A > C) in reaching a decision. This procedure resulted in eliciting transitive inferences at considerably earlier ages than Piaget had shown. Braine (1959, p.5) states that 'Piaget's diagnosis of a child's ability to perform the inferences basic to measurement is based entirely on his measuring behavior.... It seems ... that a child's measuring behavior may reflect his knowledge of measuring technique as much as it does his ability to reason logically about length.'

The monograph however has an 'importance far beyond its immediate empirical contributions' (Flavell, 1963, p.388n). Many of Piaget's tasks seem to demand a higher level of verbal comprehension by the child than the level of con-

ceptual understanding they are designed to measure. The child sometimes needs an understanding of certain words to accomplish the task, which he may not have. Braine argues that Piaget may frequently underestimate the child's basic intellectual capacities by using assessment techniques which do not differentiate between these capacities and associated verbal abilities, and which fail to ensure an adequate motivation to perform.

Smedslund (1963) believed that Braine's study had a number of weaknesses because he failed to eliminate important variables of nontransitive hypotheses, which lead to equivocal findings. Smedslund attempted to eliminate these methodological weaknesses and concluded that his data supports Piaget's original findings against Braine's. Braine (1964) argued against this criticism by pointing out some of Smedslund's (1963) methodological weaknesses involving the Ss task requirements. Smedslund used the Muller-Lyer illusion in his study and Braine argues that the study 'must be reinterpreted as a determinination of the approximate age at which children will, in a particular set of circumstances involving measurement, spontaneously interpret the question "Which is longer?" as a question of real rather than apparent length' (Braine, 1964, p.801). In Smedslund's (1965) reply to these criticisms it was generally agreed by both writers that the data obtained was ambigious and needed updating. Some clarification of the development of transitivity was needed as this is an important component in the development of successful seriation behavior.

An attempt was made by Siegel (1972) to establish that children as young as three years can manipulate the concept of seriation when the verbal requirements of the task are minimized. This study was a natural extension of her previous study (Siegel, 1971), which also explored the concepts of seriation, with a minimum amount of verbal instruction being involved. In this study Siegel remarks that the full understanding of ordinal position precedes a full comprehension of seriation. 'The understanding of the concept of addition develops later than the concepts of

seriation and ordination' (Siegel, 1971, p.357). Siegel's (1972) study was defined as a study of 'the child's ability to learn to choose a particular position, for example, smallest, in a two, three, or four object series' (1972, p.135). The results overall indicated that the end positions of the series were significantly easier to identify than the inner positions. She also claimed that children of three years of age can learn a seriation task when the verbal requirements of the task are minimized. When the children were asked to identify the middle member of a set of three or the next smallest of a set of four, there was a marked increase in the failure rate. It should be pointed out that Siegel used at the most four objects and Inhelder and Piaget (1964) report that at age four, 47% of the Ss were capable of small uncoordinated series of up to three or four in length. This would seem to indicate that in actual fact Siegel's results confirm Inhelder and Piaget's findings and that the main difficulty seemed to be in different conceptions of what was meant by seriation.

Gollin, Moody and Schadler (1974) argued that Siegel's (1972) tasks were relational learning tasks in that solution of the problem required an understanding of the size relationships among the whole population of nine possible members from which Siegel drew her sets of two, three and four members. They set out to determine if young children could learn to select reliably the appropriate cue provided that they were asked to point to the smallest member of a given set. They conclude: 'The solution of relational size problems in the context of ordinal series was facilitated when children were directed to locate a terminal member (the smallest) of the series. In contrast, subjects who were instructed to locate a non-terminal member without prior designation of a terminal member were unable to achieve solution of the problem' (Gollin, Moody & Schadler, 1974, p.101).

This brief look at a specific series of controversies in the length seriation literature indicates the large number of confused variables and methodological errors that

have lead to a large part of the disagreement. The methodological considerations which have been touched on here include the use of non-verbal directions and an adequate motivation to perform. The concept of transitivity needs to be seperated from seriation itself, and this is perhaps best obtained by the variation of the size of the series. The effect of the size of the series also needs to be investigated as a factor on its own. Another area of interest is the effect of using the location of a terminal member as an aid to subsequent seriation. Sinha and Walkerdine (1978) report some interesting results using different approaches to Piagetian-type studies, particularly in the areas of nonverbal directions and motivation, which they feel are more in line with the child's approach to the problem situation. A systematic study with these considerations in mind needs to be designed and run to give some clarity to this whole area.

CHAPTER 6

EXPERIMENTAL STUDIES: WEIGHT SERIATION

The studies outlined in the previous chapter have considered seriation of length. O'Reilly and Steger (1970) used weight seriation to investigate the child's ability to perform judgements in context, in the presence of an anchor. Their results demonstrate that the ability to seriate along a stimulus dimension was linked to the ability to utilize a frame of reference. This would be in agreement with Gollin, Moody and Schadler's (1974) study (cited earlier), where locating the end member of a series could be considered as utilizing a frame of reference. Smedslund's (1963a) suggestion that seriation facilitates transitivity was supported by O'Reilly and Steger (1970) since their anchor-type method demanded seriation. However Smedslund (1963a) reported that in general none of his Ss, aged 5 to 7 years, formed relative judgements of the weights he used. This was not the case in O'Reilly and Steger's (1970) study for the Ss could not only order the weights but shifted their whole order of judgement when given an anchor. Wadsworth (1971) reports that length seriation occurs first, around age 7, and weight seriation occurs later, at around age 9. This contention was questioned by O'Reilly and Steger, who found that children of age six and above could order the weights. My own

experiments have certainly found children of age eight capable of at least Level IIB behavior (see Chapter 7), that is, seriation.

Baylor's Studies

Baylor and Gascon (1974) outline an information processing model of the behavior of children in weight seriation tasks. They divided the development into three stages, with labelling different from that employed by Piaget (see earlier section). In general Stage 1 refers to Piaget's level IA(i), Stage 2 refers to level IA(ii), and Stage 3 appears to refer to both level IB and level IIA. I shall consider Baylor and Gascon's approach in some detail, as it throws light on both the use of an information processing approach to theory construction, and a detailed understanding of the development of a child's seriation behavior.

1. Stage 1

Stage 1 in the development of seriation behavior is referred to as the "juxtaposition of couples". Children at this level typically weigh two blocks at a time and place each ordered pair side by side in their final position. The level is further divided into stages 1A and 1B. In stage 1A, a child given an odd number of blocks will place the final single block on the balance and weigh it by itself. In stage 1B, the child anticipates the consequences of placing the block on the balance but does not in fact do so. At both levels, the result of the action is to place this final block in the "most heavy" position in the series constructed.

The base strategy for this stage as outlined by Baylor and Gascon (1974) is presented in Figure 9. The base strategy is outlined as a series of goals. The top goal is of course SERIATE. There are two subgoals. The level 2 ("lower") goal is WEIGH. This leads to the level 1 goal, FINDHEAVYLIGHT and is satisfied when the child removes the weights from the balance. The starred bracket, *[,

indicates that this section of the base strategy is repeated as many times as is necessary to complete the SERIATE goal.

SERIATE
  *[FINDHEAVYLIGHT
    -WEIGH
    -SATISFIED WEIGH
  SATISFIED FINDHEAVYLIGHT]
SATISFIED SERIATE

Figure 9. Base strategy for Stage 1 weight-seriation behavior. (Baylor and Gascon, 1974, p.8)

The production rules (see page 26) required to translate this base strategy into actual behavior are presented in Figure 10. The first two rules apply when the SERIATE goal is active (P1 initiates activity and P2 terminates it). The next three rules apply when the FINDHEAVYLIGHT goal is active. P3 places a block on the balance when there is none there, P4 moves a single block to the heavy end of the series (occurs when the child weighs one block by itself), and P5 moves the heavy block followed by the light block, when the two blocks have been weighed. The rules P6 and P7 occur when the WEIGH goal is active. They both enable blocks to be moved to the balance, either two blocks (P6), or one block when only one remains (P7).

These seven rules apply specifically for Stage 1A behavior. The final two rules in Figure 2 (P8 and P1') are necessary when considering Stage 1B. P8 considers the effect of an SERIATE goal when there is only one block left. This enables the child to place the last block directly in the heaviest position of the series without weighing it. This rule results in rule P1 being changed to the form of rule P1'. Baylor and Gascon state that this set of production rules 'simulates perfectly the details of an actual ... protocol' (1974, p.18). The model would therefore appear to be a valid one of the seriation behavior of

a Stage 1 child. While it may model external behavior, modelling of the S's understanding and more "internal" levels of the information processing mechanism is not as likely. See page 11 for a discussion of the different levels of simulation.

P1 SERIATE
(PO > 0) ⇒ (SET-GOAL FINDHEAVYLIGHT)

P2 SERIATE
(PO = 0) ⇒ (SATISFIED SERIATE)

P3 FINDHEAVYLIGHT
(PB = 0) ⇒ (SET-GOAL WEIGH)

P4 FINDHEAVYLIGHT
(PB = 1) ⇒ (MOVE PB HEAVY TO PF RIGHT)
(SATISFIED FINDHEAVYLIGHT)

P5 FINDHEAVYLIGHT
(PB = 2) ⇒ (MOVE PB HEAVY THEN PB LIGHT TO PF RIGHT)
(SATISFIED FINDHEAVYLIGHT)

P6 WEIGH
(PO > 1) ⇒ (MOVE PO RIGHT THEN PO RIGHT TO PB)
(SATISFIED WEIGH)

P7 WEIGH
(PO = 1) ⇒ (MOVE PO RIGHT TO PB)
(SATISFIED WEIGH)

P11 SERIATE
(PO > 1) ⇒ (SET-GOAL FINDHEAVYLIGHT)

P8 SERIATE
(PO = 1) ⇒ (MOVE PO TO PF RIGHT)
(SATISFIED SERIATE)

Figure 10. Production rules for Stage 1(A and B) weight seriation behavior. (Baylor and Gascon, 1974, p.14)
(Note: PO - original position; PB - balance position; PF - final position.)

Baylor and Gascon's program was written in a special computer language called BG (see Gascon and Baylor, 1973), which was designed for this task. It gives a clear indication of the steps taken to reach the final goal. However, as this language is not available locally, I rewrote the program, initially in FORTRAN and later in PASCAL to enable it to be run locally. There was some loss of clarity and elegance, but the actual simulation of the behavior has been as effective as the original.
Chapter 7 outlines the procedure followed and the current status of this work.

2. Stage 2

In Stage 2 the defining strategy is that called "Juxtaposition of subseries". Once again the stage can be divided into parts A and B, dependent on the length of the subseries formed. Stage 2A children succeed on most three-block problems, whereas Stage 2B children are able to solve at least fifty percent of four-block problems correctly.

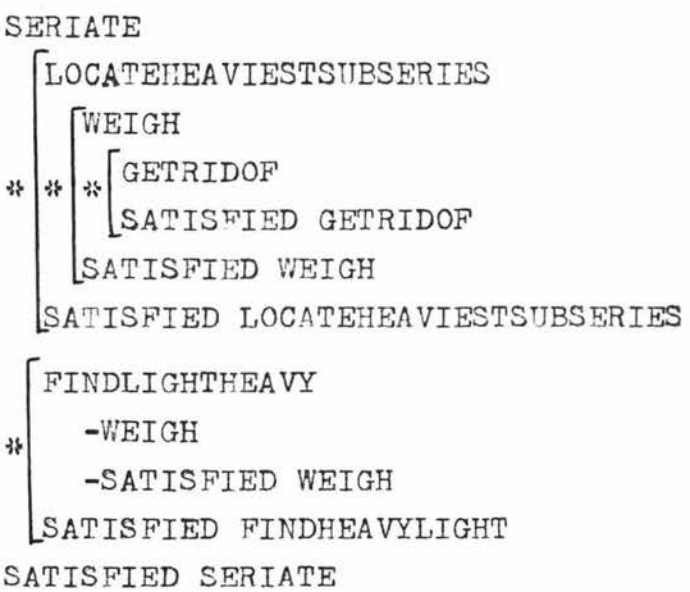

Figure 11. Base strategy for Stage 2 weight-seriation behavior. (Baylor and Gascon, 1974, p.8)

The base strategy for this stage is presented in Figure 11 and shows that the LOCATEHEAVIESTSUBSERIES goal leads to an additional goal, which is stated as GETRIDOF. This involves removing the lighter block from the balance until the heaviest of a subseries is found. This particular base strategy leads to a juxtaposition of two subseries, either partially or completely ordered, without any coordination between the two. A checking strategy FINDHEAVYLIGHT, a return to Stage 1 behavior, is evoked and leads to a juxtaposition of light-heavy couples in the final position.

The production rules for this stage are presented in Figure 12. The first three rules apply when SERIATE is active (P1 initiates the activity and P2 terminates it). In the special situation where there are no blocks left in the original position, but still one left in the final position needing to be checked (PF NEXT), then the goal FINDLIGHTHEAVY is initiated. Production rules P4 to P7 are evoked when the active goal is LOCATEHEAVIESTSUBSERIES. From the model, exactly which production is evoked depends on the number of blocks in the balance (PB), the number in the position where the current subseries is being formed (PF1), and whether there is a heaviest block so far to be checked against any other blocks (LASTHEAVIER). This is obvious at an experimental level. A subject is going to approach a task differently depending on the initial conditions. Production rules P8 and P9 apply when the FINDLIGHTHEAVY goal is active and depend entirely on whether there are weights on the balance or not (PB). If there is, P8 occurs, and if not the goal is set at WEIGH (P9). Production rules P10 to P13 result in blocks being weighed on the scale (the WEIGH goal is active). All four of these rules achieve their goal and apply at different stages of the process. P10 weighs two blocks from the final position (PF) where there are none left in the original position (PO); P11 takes one from PF and one from PO, when only one remains in PO; P12 takes two blocks from PO when there are no blocks on the scale, and P13 is evoked when there is already one block on the scale. The final production, P14, is evoked when the GETRIDOF goal is evoked (after P6).

P1 SERIATE
(PO > 0) ⇒ (SET-GOAL LOCATEHEAVIESTSUBSERIES)

P2 SERIATE
(PO = 0)&(PF NEXT = NIL) ⇒ (SATISFIED SERIATE)

P3 SERIATE
(PO = 0)&(PF NEXT =T) ⇒ (SET-GOAL FINDLIGHTHEAVY)

P4 LOCATEHEAVIESTSUBSERIES
(PB < 2) ⇒ (SET-GOAL WEIGH)

P5 LOCATEHEAVIESTSUBSERIES
(PB = 2)&(PF1 = 0) ⇒ (LET LASTHEAVIER = PB HEAVIER; MOVE PB HEAVIER THEN PB LIGHTER TO PF1)

P6 LOCATEHEAVIESTSUBSERIES
(PB = 2)&(PF1 > 0)&(LASTHEAVIER = T) ⇒ SET-GOAL GETRIDOF

P7 LOCATEHEAVIESTSUBSERIES
(PB = 2)&(PF1 > 0)&(LASTHEAVIER = NIL) ⇒ (LET NEXT = PF RIGHT; MOVE PB LIGHTER THEN PB HEAVIER TO PF1 LEFT; TRANSFORM PF1 TO PF; SATISFIED LOCATEHEAVIESTSUBSERIES)

P8 FINDLIGHTHEAVY
(PB = 2) ⇒ (LET NEXT = PF LEFTOFHOLE; MOVE PB HEAVIER THEN PB LIGHTER TO PF HOLE; SATISFIED FINDLIGHTHEAVY)

P9 FINDLIGHTHEAVY
(PB < 2) ⇒ (SET-GOAL WEIGH)

P10 WEIGH
(PB = 0)&(PO = 0) ⇒ (MOVE PF NEXT AND PF LEFTOFHOLE TO PB; SATISFIED WEIGH)

P11 WEIGH
(PB = 0)&(PO = 1) ⇒ (MOVE PO RIGHT AND PF1 RIGHT TO PB; SATISFIED WEIGH)

P12 WEIGH
(PB = 0)&(PO > 1) ⇒ (MOVE PO RIGHT AND PO RIGHT TO PB; SATISFIED WEIGH)

P13 WEIGH
(PB =1) ⇒ (MOVE PF1 LASTHEAVIER TO PB; LET LASTHEAVIER = NIL; SATISFIED WEIGH)

P14 GETRIDOF
(LASTHEAVIER = T) ⇒ (MOVE PB LIGHTER TO PF1; SATISFIED GETRIDOF)

Figure 12. Production rules for Stage 2 weight seriation behavior. (Baylor and Gascon, 1974, p.20)

There are a larger number of variables in the Stage 2 behavior compared to the Stage 1 behavior. This is due to the fact that part of the final position (PF) is used to form a current subseries (PF1) which is then transformed into PF when the subseries has been completed. LASTHEAVIER refers to a memory which remembers whether the heaviest block in a subseries has been weighed with all the potentially heavier candidates. A hole is created in PF when blocks are moved from there back to PB, and this explains the HOLE-based instructions in Figure 12.

This particular production system outlines one possible seriation approach. There are many others. Stage 2 to appear to be the hardest one to model due to this great variety of approaches. This may relate to the fact that the child's seriation behavior is in a state of transition from the simple two-place behavior of Stage 1 to the complete seriation behavior of Stage 3.

3. Stage 3

In Stage 3, the most common strategy is called "Locate Heaviest", and has the same structure as the LOCATEHEAVIESTSUBSERIES presented in Figure 11, except that it now refers to the whole series, and not simply an isolated subseries. Other strategies are possible, such as a less common "Insertion", and an idiosyncratic "Heavy-light Sieve" (Baylor and Gascon, 1974), and further research may reveal some others. Only the base strategies for the

"Locate Heaviest" and the "Insertion" strategy are presented here. The former is shown in Figure 13 and the latter in Figure 14.

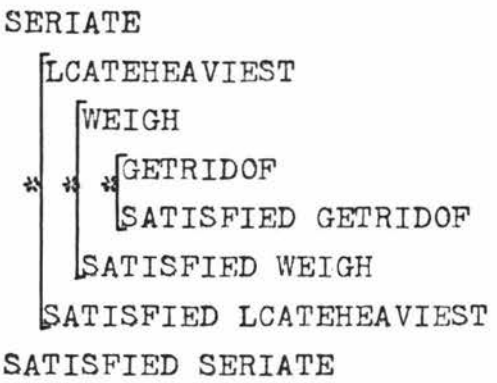

Figure 13. Base strategy for Stage 3 weight-seriation behavior ("Locate Heaviest"). (Baylor and Gascon, 1974, p.8)

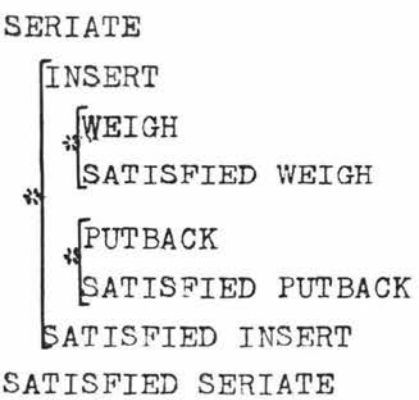

Figure 14. Base strategy for Stage 3 weight-seriation behavior ("Insertion"). (Baylor and Gascon, 1974, p.8)

As the "Locate Heaviest" strategy is the most common, only the production rules based on this particular base strategy will be considered here. These are outlined in Figure 15. An important addition is the intermediate position (PI). While the S is looking for the heaviest block, the lighter blocks are discarded into a new position, PI. When the heaviest block has been found, this PI reverts back to the original position (PO). The seriation starts again if blocks still remain.

P1 SERIATE
(PO > 1) ⇒ (SET-GOAL LCATEHEAVIEST)

P2 SERIATE
(PO = 1)&(PB = 1) ⇒ (SET-GOAL LCATEHEAVIEST)

P3 SERIATE
(PO = 0)&(PI > 0) ⇒ (TRANSFORM PI TO PO)

P4 SERIATE
(PO = 0)&(PI = 0)&(PB = 1) ⇒ (MOVE PB TO PF RIGHT; SATISFIED SERIATE)

P5 LCATEHEAVIEST
(PB < 2) ⇒ (SET-GOAL WEIGH)

P6 LCATEHEAVIEST
(PB = 2)&(PO > 0) ⇒ (SET-GOAL GETRIDOF)

P7 LCATEHEAVIEST
(PB = 2)&(PO = 0) ⇒ (MOVE PB HEAVIER TO PF RIGHT; SATISFIED LCATEHEAVIEST)

P8 WEIGH
(PB = 0) ⇒ (MOVE PO RIGHT THEN PO RIGHT TO PB; SATISFIED WEIGH)

P9 WEIGH
(PB = 1) ⇒ (MOVE PO RIGHT TO PB; SATISFIED WEIGH)

P10 GETRIDOF
(MOVE PB LIGHTER TO PI; SATISFIED GETRIDOF)

Figure 15. Production rules for Stage 3 weight seriation behavior. (Baylor and Gascon, 1974, p.25)

The first four rules apply when the SERIATE goal is active. If there is more than one block in PO, the LCATEHEAVIEST goal is set (P1); if there is one block in PO and one on the balance (PB), the LCATEHEAVIEST goal is again set (P2); if there are no blocks left in PO but some in PI, PI becomes PO (P3); and if no blocks remain in PO

or PI, the SATISFIED SERIATE state is reached (P4). The next three rules occur when the LCATEHEAVIEST goal is active. If there are less than two blocks on the balance, the goal is set at WEIGH (P5); if the balance is full and there are still weights in PO, the goal is set at GETRIDOF (P6); and if the balance is full and no blocks remain in PO, the LCATEHEAVIEST goal is satisfied (P7). Productions P8 and P9 occur when the WEIGH goal is active, and depend on whether there are no weights in PB (P8) or one weight (P9). P10 occurs when the GETRIDOF goal is active.

4. Concluding remarks

The approach followed by Baylor and Gascon (1974) has been presented in some detail as it forms the basis for both the experimental work and the computer program development which is outlined in the following chapter. Although the details of the production systems for each stage appear fairly complex, the base strategies from which they derive are relatively simple. Baylor and Gascon (1974) believe that a child will not carry in his head such a specific representation as the production systems indicate. This appears to contradict Newell and Simon's hypothesis (see page 29). This may be due in part to the fact that the computer requires a more specific representation of the environment than does the child. This is a major problem in the field and one that is receiving some attention (see Bobrow and Collins, 1975). However, they do state that 'a child is postulated to come into the experiment with the components of a particular base strategy in his head. In order to convert a base strategy into actual behavior, it has been suggested that he has a complier (or interpreter) available that enables him to construct a specific production system that carries out the base strategy' (Baylor & Gascon, 1974, pp.38 - 39). In spite of these reservations, this production system approach has proved successful at simulating the behavior of individual children, as will be seen in the following pages.

CHAPTER 7

A REPORT ON THE PROGRESS OF THE PRESENT STUDY

The practical thrust of this study has proceeded along two main aspects. Firstly there has been the development of a series of programs, which could be run locally, capable of simulating the various stages of the target behavior. Secondly there has been the experimental work with children to verify the findings of Baylor and Gascon. I shall consider each of these aspects in turn in the following two sections and in the third section discuss the relationship of the two aspects.

1. Programming Aspect

The development of the simulation programs which have been an important component of this study was carried out on a Burroughs B6700 computer using the high-level programming language PASCAL. This language was chosen for a number of reasons. It is a very powerful language having many features which make it useful for this type of work; secondly, as it is based on the principles of the increasingly important "structured programming" approach, it will become more widely used in the future; and lastly, its present use in a post-graduate Artificial Intelligence course has ensured its development at this installation.

The production systems of Baylor and Gascon (1974),

which have been outlined in some detail in Chapter 6, were used as the basis for the development of the programs. Programs were developed for Stage 1A (Appendix I) and Stage 1B (Appendix II). A program was also written to simulate an alternative type of Stage 1 behavior, not covered by Baylor and Gascon, to be discussed in the next section, and this program is presented in Appendix III.

Stage 2 behavior was not simulated for the purposes of this study because of its complexity. Furthermore, more comprehensive studies of the type of behavior exhibited in this level should be done before time is spent on attempting to model it. Appendix IV gives a program for the simulation of Stage 3 behavior based directly on Baylor and Gascon's work. Appendix V gives an upgraded version of this program, which contains some small refinements.

All the programs produced are divided into five sections. Firstly the declaration section, where the components of the program are declared. Following this there is a collection of basic "housekeeping" procedures relating to the manipulation of a goalstack, data input, initialization, and printing of data. Thirdly, procedures to move the weights from one place to another during the simulation are presented. Fourthly the production system is shown, giving the initial conditions followed by the appropriate action. Finally, there is the main program, which initiates the whole process and controls it through calling various procedures, depending on the current goal.

Figure 16 shows a decision flow for Stage 1A behavior, indicating the appropriate goals and initial conditions which (according to the model) are sufficient to trigger off the productions and subsiduary goals. This decision flow reflects the control of the program contained in the production system. Figure 17 shows the comparable decision flow for Stage 3.

Finally, it is worth noting the slight modification of the Stage 3 behavior to make it more closely simulate the seriation behavior. In Appendix IV (page 128), the procedure PBTOPI accomplishes the same result as Baylor and

Gascon's program and results in an ordered placement of the blocks from the balance pan to the intermediate position. In Appendix V (page 144), the procedure PRTOPI has been upgraded with the addition of a random number component, giving a more realistic simulation of the behavior. A few other minor changes are also necessary with the addition of this component. For this reason it was decided to present the whole program in full, even though it is substantially identical to Appendix IV.

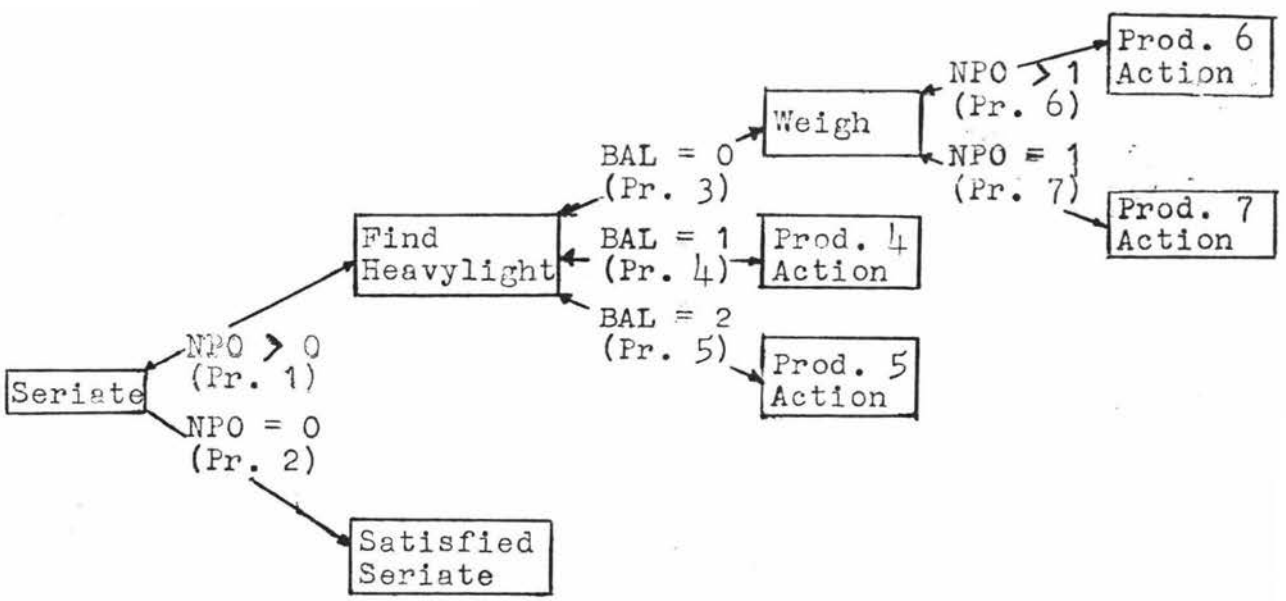

Figure 16. Decision flow of program indicating initial goals and conditions for the production system (Stage 1).

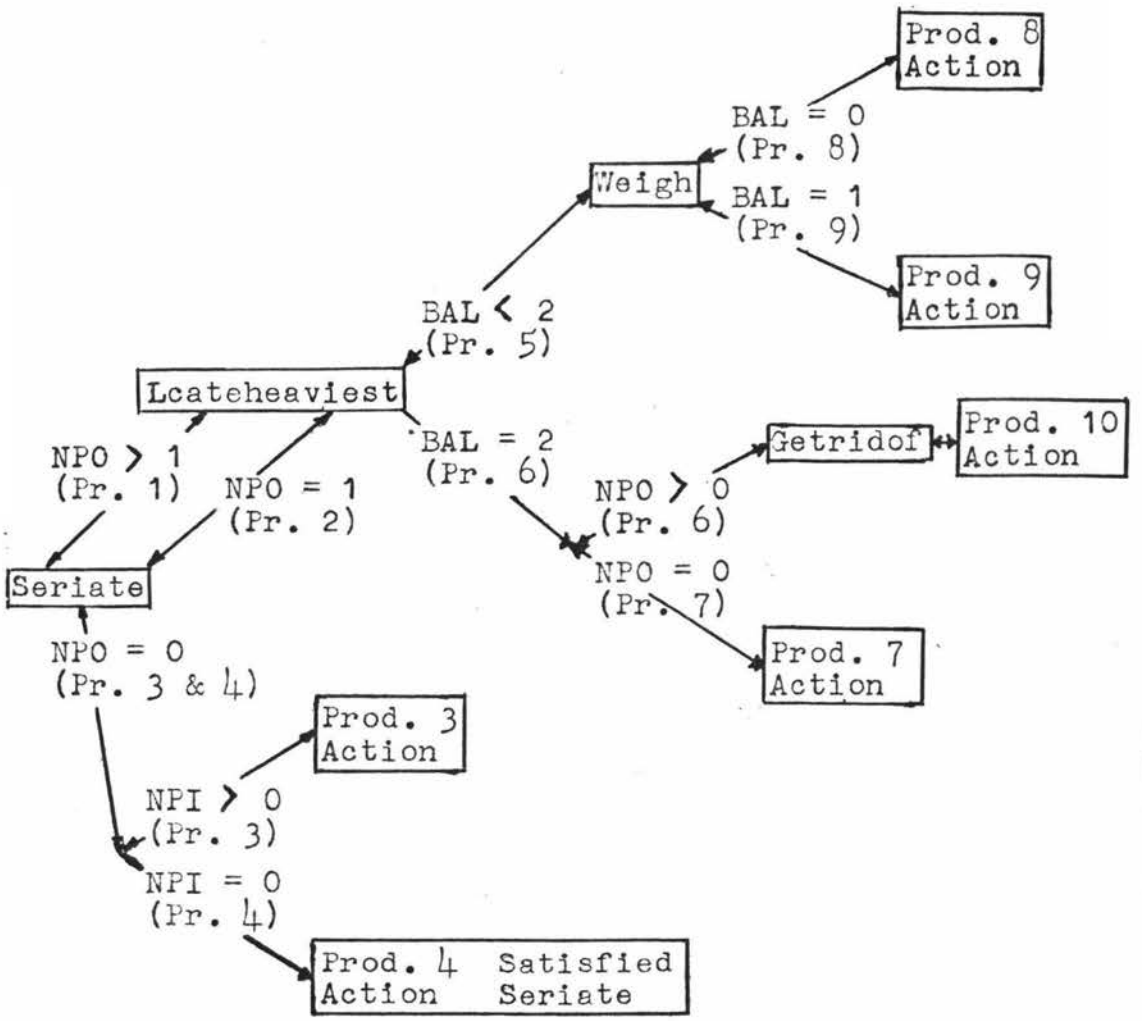

Figure 17. Decision flow of program indicating initial goals and conditions for the production system (Stage 3).

2. Experimental Aspect

2.(1). Subjects

Twenty children, ranging in age from 6;4 to 9;5, were presented with a weight seriation task to solve which was similar to that used by Baylor and Gascon (1974). Protocols of their performances were collected but only four of these are presented in detail. The children were selected from the P3 to S2 classes of a local primary school. As this study was only interested in looking in detail at Stage 1 and Stage 3 performance, any results which were not of

either type were rejected for the purposes of the study. The children were tested until twenty showing either Stage 1 or Stage 3 behavior had been found.

2.(2). Procedure

Each child (S) was tested individually, seated opposite the experimenter (E) at a table. A Stoelting balance with two pans was used, one of which moved up and the other down when weights were placed on the balance. This helped to accentuate the difference between the two weighs. The weights were virtually identical containers, each with a letter written on the top and bottom sides, for the purpose of identification. The letter names and actual weights of each block were: S = 182.7g; Q = 181.8g; M = 180.8g; B = 180.0g; A = 179.0g; V = 178.1g; O = 177.2g. Two different orders of presentation were used. These were BASQOMV and AQVBMOS. Figure 18 shows a plan of the experimental setup, showing the positions of the original and final positions, and the balance, in relation to S and E. A brief pretest was given to determine if the child could understand which was the heavier of two containers

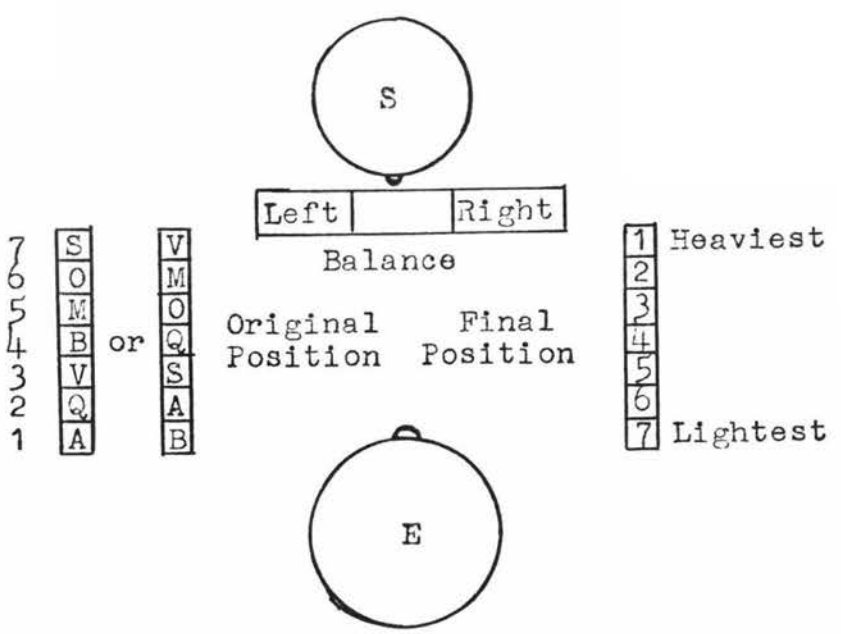

Figure 18. Diagram showing experimental set up, viewed from top.

presented.

The instructions given were practically identical to those given in Baylor and Gascon (1974, p.3). These were:

> See, I have lots and lots of yoghurt cartons here. My cartons are all the same size but they do not weigh the same. Out of all these blocks what I want you to do is to find me the heaviest, then the block that weighs a little bit less, then the block that weighs still a little bit less, and so on until the least, least, least heavy of all the blocks.
>
> Now there is one rule in my game: You can weigh the blocks as often as you think it's necessary, but you cannot put more than two blocks on the scale at the same time because there are only two places.
>
> Okay, do you understand what I want? I want you to find which one of the blocks is the heaviest, then the one that weighs a little bit less, and so on until the least, least, least heavy of all the blocks. And then when you are sure, put the heaviest block here. Because it's the heaviest, you don't have to move it as far. Then put the one that weighs a little bit less here; and so on until the least, least, least heavy block here. Do you understand? Can you repeat back to me what I just asked you?

2.(3). Results

Of the twenty children tested, eleven were found to have recognizable Stage 3 behavior of a type that is consistent with the model being presented here. Their ages ranged from 7;9 to 9;5. There were 6 boys and 5 girls. The remaining nine exhibited what was recognized as Stage 1 behavior. Their ages ranged from 6;4 to 7;11. There were 4 boys and 5 girls. It should be noted that these ages cannot be considered as representing any information concerning the age range of the various stages, as this was not the purpose of the study.

Of the nine children studied exhibiting Stage 1 behavior

only three showed the stage 1A behavior described by Baylor and Gascon (1974). Only one of these (Tor 6;8) exhibited behavior which exactly matched the computer simulation (see Tor's protocol on page 65). The other two (Sharon 7;4 and Kevin 6;10) did not match exactly the computer prediction (see Kevin's protocol on page 66). This was due to differences in the manner of removing blocks from the original position. These two children did not do this in a systematic way. None of the children studied exhibited the Stage 1B behavior described by Baylor and Gascon (1974).

The remaining 6 children showed behavior that was not found by Baylor and Gascon's study. This I have termed the "Opposite placement of couples". The same weighing procedure is carried out, in that the children take two blocks from the original position and place them on the balance pans. However, when the blocks are removed, the heaviest goes to the heavy end and the lightest to the light end of the final position series. In this way the series is built up from both ends. With an odd number of blocks, the final block goes into the center position. In Baylor and Gascon's Stage 1 formulation, they state the last weight is thought of as the heaviest 'perhaps because of the rapidity of descent of the pan of the scale holding the solitary block' (1974, p.17). However, their subjects, and the children studied here placed their last block with no consideration given to this fact, either in the center or at the light end. This behavior may indicate that the child is more effected by the perceptual Gestalt property of closure than with the logico-mathematical properties of highest weight. See Samuel's protocol on page 68 as an example of the behavior exhibited by this alternative strategy. Only 3 of the children studied exhibited behavior that exactly matched the Stage 1 alternative program (Samuel 7;11, Cherie 6;11 and Julian 6;4). The other three (Maree 6;4, David 7;1 and Lisa 6;7) removed blocks randomly from the original position.

Seven of the eleven children who exhibited Stage 3 behavior used the "Locate heaviest" strategy. The remaining four used idiosyncratic methods. Their protocols were not followed up, as different simulation programs would have needed to be devised. Of the seven children, none were exactly modelled by the program, due to the fact that the blocks removed from the balance pans and placed in the intermediate position were placed in an unstructured heap and not in any ordered way, as is done by the program. Even with the modified Stage 3 program (see Appendix V) there is no way that an exact copy can be guaranteed, due to the influence of chance. All of the eleven children from this stage removed the blocks from the original position in a systematic manner.

3. Conclusions

3.(1) Stage 1

As mentioned in the previous section, a few children at the Stage 1A level seriated the blocks in the manner outlined by Baylor and Gascon (1974) which they titled "Juxtaposition of couples". The computer program developed to simulate this behavior is shown in Appendix I. The print out for a typical run is also included here, and is shown in a clearer form in Figure 19. Compare this with the result obtained for Tor (6;8), which is presented in Figure 20. It can be seen here that the program simulates the child's behavior exactly and that they both end up with identical final positions.

However, once again compare Figure 19 with the result recorded for Kevin (6;10) which is shown in Figure 21. The simulation breaks down because of the child's removal of the blocks from the original position, which in the model is done in an orderly manner from right to left. Kevin has taken his blocks from the original position in a random manner. This would be a relatively simple procedure to incorporate into the model, as it only requires the addition of a degree of randomness into the selection procedure, as has been done in the PBTOPI

procedure for Stage 3 (see page 144).

The other common approach found in this study, which is not mentioned by Baylor and Gascon, is the "Opposite placement of couples". A computer program was also developed to simulate this behavior and is shown in Appendix 3, along with a typical print out. The print out is also shown in a more immediately readable form in Figure 22. A typical result for a child's performance (Samuel (7;11)) is shown in Figure 23 and once again it can be seen that the two protocols are identical. Other children studied also took blocks from the original position in a random manner, in a similar way to that described for Stage 1A. No detailed example has been given

Active Goal	Production	Original Position	Balance L	Balance R	Final Position
Seriate	01	4512736	-	-	-------
Findheavylight	03	4512736	-	-	-------
Weigh	06	45127--	6	3	-------
Findheavylight	05	45127--	-	-	36-----
Seriate	01	45127--	-	-	36-----
Findheavylight	03	45127--	-	-	36-----
Weigh	06	451----	7	2	36-----
Findheavylight	05	451----	-	-	3627---
Seriate	01	451----	-	-	3627---
Findheavylight	03	451----	-	-	3627---
Weigh	06	4------	1	5	3627---
Findheavylight	05	4------	-	-	362715-
Seriate	01	4------	-	-	362715-
Findheavylight	03	4------	-	-	362715-
Weigh	07	-------	4	-	362715-
Findheavylight	04	-------	-	-	3627154
Seriate	02	-------	-	-	3627154

Figure 19. Simplified example of print out for simulation of Stage 1A behavior (from Appendix I).

of this random removal for the alternative Stage 1 strategy. The final weight is placed in the center of of the final position array. As mentioned earlier, this may be more a property of Gestalt closure, as a gap is usually left in the middle as the blocks are placed in their final position.

3.(2). Stage 3

The Stage 3 strategy considered in detail is the "Locate heaviest" strategy described earlier (see pages 52 to 55). The program developed to simulate this behavior is presented in Appendix IV, together with a typical printout. The printout is also presented in Figure 24,

Active Goal	Production	Original Position	Balance L	Balance R	Final Position
Seriate	01	BASQOMV	-	-	-------
Findheavylight	03	BASQOMV	-	-	-------
Weigh	06	BASQO--	V	M	-------
Findheavylight	05	BASQO--	-	-	MV-----
Seriate	01	BASQO--	-	-	MV-----
Findheavylight	03	BASQO--	-	-	MV-----
Weight	06	BAS----	O	Q	MV-----
Findheavylight	05	BAS----	-	-	MVQO---
Seriate	01	BAS----	-	-	MVQO---
Findheavylight	03	BAS----	-	-	MVQO---
Weigh	06	B------	S	A	MVQO---
Findheavylight	05	B------	-	-	MVQOSA-
Seriate	01	B------	-	-	MVQOSA-
Findheavylight	03	B------	-	-	MVQOSA-
Weigh	07	-------	B	-	MVQOSA-
Findheavylight	04	-------	-	-	MVQOSAB
Seriate	02	-------	-	-	MVQOSAB (3627154)

Figure 20. Protocol for Tor (6;8) on a weight seriation task.

on pages 69 to 71. Warwick (8;4) gives a very clear indication of this strategy. His protocol is presented in Figure 25, on pages 72 to 74.

The discrepancies between the model and Warwick's observed behavior result from the random way in which the subject assigns blocks to PI after weighing them, compared to the systematic method used by the program. As described earlier (see pages 57 to 58) a modification was developed to deal with this, and this is found in Appendix V.

It is also worth noting here a point which shows the limitations of this stage of seriation behavior, which intiially appears very elegant and logical. On one

Active Goal	Production	Original Position	Balance L	Balance R	Final Position
Seriate	01	BASQOMV	-	-	-------
Findheavylight	03	BASQOMV	-	-	-------
Weigh	06	BA-QO-V	M	S	-------
Findheavylight	05	BA-QO-V	-	-	SM-----
Seriate	01	BA-QO-V	-	-	SM-----
Findheavylight	03	BA-QO-V	-	-	SM-----
Weigh	06	-A--O-V	Q	B	SM-----
Findheavylight	05	-A--O-V	-	-	SMQB---
Seriate	01	-A--O-V	-	-	SMQB---
Findheavylight	03	-A--O-V	-	-	SMQB---
Weigh	06	-A-----	V	O	SMQB---
Findheavylight	05	-A-----	-	-	SMQBVO-
Seriate	01	-A-----	-	-	SMQBVO-
Findheavylight	03	-A-----	-	-	SMQBVO-
Weigh	07	-------	A	-	SMQBVO-
Findheavylight	04	-------	-	-	SMQBVOA
Seriate	02	-------	-	-	SMQBVOA (1324675)

Figure 21. Protocol for Kevin (6;10) on a weight seriation task.

occasion (when A was placed on the balance and remained there until B, the final weight in the current original position series, was weighed) the last weight placed on the balance was the heaviest, indicating that the weight that had been on the balance up to that point had been the heaviest, and would now become the next-to- heaviest. However the subject did not realize this, and proceeded to weigh all the weights again.

From the preceding discussion, it can be seen that the programs developed simulate most areas of behavior perfectly, but there are still some discrepancies between the model and the observed behavior. The occurence of these

Active Goal	Production	Original Position	Balance L	Balance R	Final Position
Seriate	01	4512736	-	-	-------
Findheavylight	03	4512736	-	-	-------
Weigh	06	45127--	6	3	-------
Findheavylight	05	45127--	-	-	3-----6
Seriate	01	45127--	-	-	3-----6
Findheavylight	03	45127--	-	-	3-----6
Weigh	06	451----	7	2	3-----6
Findheavylight	05	451----	-	-	32---76
Seriate	01	451----	-	-	32---76
Findheavylight	03	451----	-	-	32---76
Weigh	06	4------	1	5	32---76
Findheavylight	05	4------	-	-	321-576
Seriate	01	4------	-	-	321-576
Findheavylight	03	4------	-	-	321-576
Weigh	07	-------	4	-	321-576
Findheavylight	04	-------	-	-	3214576
Seriate	02	-------	-	-	3214576

Figure 22. Simplified example of print out for simulation of Stage 1 Alternative behavior (from Appendix III).

discrepancies has been put down to chance, and appropriately dealt with. However further research may show that this apparently "random" behavior is under the control of some particular previously unconsidered parameter, and future programs may be able to incorporate this fact, and so move the program one step further to an accurate simulation of the external behavior.

Active Goal	Production	Original Position	Balance L	Balance R	Final Position
Seriate	01	BASQOMV	-	-	-------
Findheavylight	03	BASQOMV	-	-	-------
Weigh	06	BASQO--	V	M	-------
Findheavylight	05	BASQO--	-	-	M-----V
Seriate	01	BASQO--	-	-	M-----V
Findheavylight	03	BASQO--	-	-	M-----V
Weigh	06	BAS----	O	Q	M-----V
Findheavylight	05	BAS----	-	-	MQ---OV
Seriate	01	BAS----	-	-	MQ---OV
Findheavylight	03	BAS----	-	-	MQ---OV
Weigh	06	B------	S	A	MQ---OV
Findheavylight	05	B------	-	-	MQS-AOV
Seriate	01	B------	-	-	MQS-AOV
Findheavylight	03	B------	-	-	MQS-AOV
Weigh	07	-------	B	-	MQS-AOV
Findheavylight	04	-------	-	-	MQSBAOV
Seriate	02	-------	-	-	MQSBAOV (3214576)

Figure 23. Protocol for Samuel (7;11) on a weight seriation task.

Active Goal	Prod	Original Position	Balance L	Balance R	Intermediate Position	Final Position
Seriate	01	5264371	-	-	-------	-------
Lcateheaviest	05	5264371	-	-	-------	-------
Weigh	08	52643--	1	7	-------	-------
Lcateheaviest	06	52643--	1	7	-------	-------
Getridof	10	52643--	1	-	7------	-------
Lcateheaviest	05	52643--	1	-	7------	-------
Weigh	09	5264---	1	3	7------	-------
Lcateheaviest	06	5264---	1	3	7------	-------
Getridof	10	5264---	1	-	73-----	-------
Lcateheaviest	05	5264---	1	-	73-----	-------
Weigh	09	526----	1	4	73-----	-------
Lcateheaviest	06	526----	1	4	73-----	-------
Getridof	10	526----	1	-	734----	-------
Lcateheaviest	05	526----	1	-	734----	-------
Weigh	09	52-----	1	6	734----	-------
Lcateheaviest	06	52-----	1	6	734----	-------
Getridof	10	52-----	1	-	7346---	-------
Lcateaviest	05	52-----	1	-	7346---	-------
Weigh	09	5------	1	2	7346---	-------
Lcateheaviest	06	5------	1	2	7346---	-------
Getridof	10	5------	1	-	73462--	-------
Lcateheaviest	05	5------	1	-	73462--	-------
Weigh	09	-------	1	5	73462--	-------
Lcateheaviest	07	-------	-	5	73462--	1------
Seriate	03	73462--	-	5	-------	1------
Seriate	01	73462--	-	5	-------	1------
Lcateheaviest	05	73462--	-	5	-------	1------
Weigh	09	7346---	2	5	-------	1------
Lcateheaviest	06	7346---	2	5	-------	1------
Getridof	10	7346---	2	-	5------	1------
Lcateheaviest	05	7346---	2	-	5------	1------
Weigh	09	734----	2	6	5------	1------
Lcateheaviest	06	734----	2	6	5------	1------
Getridof	10	734----	2	-	56-----	1------
Lcateheaviest	05	734----	2	-	56-----	1------
Weigh	09	73-----	2	4	56-----	1------

Active Goal	Prod	Original Position	Balance L	Balance R	Intermediate Position	Final Position
Lcateheaviest	06	73-----	2	4	56-----	1------
Getridof	10	73-----	2	-	564----	1------
Lcateheaviest	05	73-----	2	-	564----	1------
Weigh	09	7------	2	3	564----	1------
Lcateheaviest	06	7------	2	3	564----	1------
Getridof	10	7------	2	-	5643---	1------
Lcateheaviest	05	7------	2	-	5643---	1------
Weigh	09	-------	2	7	5643---	1------
Lcateheaviest	07	-------	-	7	5643---	12-----
Seriate	03	5643---	-	7	-------	12-----
Seriate	01	5643---	-	7	-------	12-----
Lcateheaviest	05	5643---	-	7	-------	12-----
Weigh	09	564----	3	7	-------	12-----
Lcateheaviest	06	564----	3	7	-------	12-----
Getridof	10	564----	3	-	7------	12-----
Lcateheaviest	05	564----	3	-	7------	12-----
Weigh	09	56-----	3	4	7------	12-----
Lcateheaviest	06	56-----	3	4	7------	12-----
Getridof	10	56-----	3	-	74-----	12-----
Lcateheaviest	05	56-----	3	-	74-----	12-----
Weigh	09	5------	3	6	74-----	12-----
Lcatehaeviest	06	5------	3	6	74-----	12-----
Getridof	10	5------	3	-	746----	12-----
Lcateheaviest	05	5------	3	-	746----	12-----
Weigh	09	-------	3	5	746----	12-----
Lcateheaviest	07	-------	-	5	746----	123----
Seriate	03	746----	-	5	-------	123----
Seriate	01	746----	-	5	-------	123----
Lcateheaviest	05	746----	-	5	-------	123----
Weigh	09	74-----	6	5	-------	123----
Lcateheaviest	06	74-----	6	5	-------	123----
Getridof	10	74-----	-	5	6------	123----
Lcateheaviest	05	74-----	-	5	6------	123----
Weigh	09	7------	4	5	6------	123----
Lcateheaviest	06	7------	4	5	6------	123----
Getridof	10	7------	4	-	65-----	123----

Active Goal	Prod	Original Position	Balance L	Balance R	Intermediate Position	Final Position
Locateheaviest	05	7------	4	-	65-----	123----
Weigh	09	-------	4	7	65-----	123----
Locateheaviest	07	-------	-	7	65-----	1234---
Seriate	03	65-----	-	7	-------	1234---
Seriate	01	65-----	-	7	-------	1234---
Locateheaviest	05	65-----	-	7	-------	1234---
Weigh	09	6------	5	7	-------	1234---
Locateheaviest	06	6------	5	7	-------	1234---
Getridof	10	6------	5	-	7------	1234---
Locateheaviest	05	6------	5	-	7------	1234---
Weigh	09	-------	5	6	7------	1234---
Locateheaviest	07	-------	-	6	7------	12345--
Seriate	03	7------	-	6	-------	12345--
Seriate	02	7------	-	6	-------	12345--
Locateheaviest	05	7------	-	6	-------	12345--
Weigh	09	-------	7	6	-------	12345--
Locateheaviest	07	-------	7	-	-------	123456-
Seriate	04	-------	-	-	-------	1234567

Figure 24. Simplified example of print out for simulation of Stage 3 behavior (from Appendix IV).

Active Goal	Prod	Original Position	Balance L	Balance R	Intermediate Position	Final Position
Seriate	01	AQVBMOS	-	-	-------	-------
Lcateheaviest	05	AQVBMOS	-	-	-------	-------
Weigh	08	AQVBM--	S	O	-------	-------
Lcateheaviest	06	AQVBM--	S	O	-------	-------
Getridof	10	AQVBM--	S	-	O------	-------
Lcateheaviest	05	AQVBM--	S	-	O------	-------
Weigh	09	AQVB---	S	M	O------	-------
Lcateheaviest	06	AQVB---	S	M	O------	-------
Getridof	10	AQVB---	S	-	O--M---	-------
Lcateheaviest	05	AQVB---	S	-	O--M---	-------
Weigh	09	AQV----	S	B	O--M---	-------
Lcateheaviest	06	AQV----	S	B	O--M---	-------
Getridof	10	AQV----	S	-	O-BM---	-------
Lcateheaviest	05	AQV----	S	-	O-BM---	-------
Weigh	09	AQ-----	S	V	O-BM---	-------
Lcateheaviest	06	AQ-----	S	V	O-BM---	-------
Getridof	10	AQ-----	S	-	O-BMV--	-------
Lcateheaviest	05	AQ-----	S	-	O-BMV--	-------
Weigh	09	A------	S	Q	O-BMV--	-------
Lcateheaviest	06	A------	S	Q	O-BMV--	-------
Getridof	10	A------	S	-	OQBMV--	-------
Lcateheaviest	05	A------	S	-	OQBMV--	-------
Weigh	09	-------	S	A	OQBMV--	-------
Lcateheaviest	07	-------	-	A	OQBMV--	S------
Seriate	03	OQBMV--	-	A	-------	S------
Seriate	01	OQBMV--	-	A	-------	S------
Lcateheaviest	05	OQBMV--	-	A	-------	S------
Weigh	09	OQBM---	V	A	-------	S------
Lcateheaviest	06	OQBM---	V	A	-------	S------
Getridof	10	OQBM---	-	A	---V---	S------
Lcateheaviest	05	OQBM---	-	A	---V---	S------
Weigh	09	OQB----	M	A	---V---	S------
Lcateheaviest	06	OQB----	M	A	---V---	S------
Getridof	10	OQB----	M	-	-A-V---	S------
Lcateheaviest	05	OQB----	M	-	-A-V---	S------
Weigh	09	OQ-----	M	B	-A-V---	S------

Active Goal	Prod	Original Position	Balance L	Balance R	Intermediate Position	Final Position
Lcateheaviest	06	OQ-----	M	B	-A-V---	S------
Getridof	10	OQ-----	M	-	-ABV---	S------
Lcateheaviest	05	OQ-----	M	-	-ABV---	S------
Weigh	09	O------	M	Q	-ABV---	S------
Lcateheaviest	06	O------	M	Q	-ABV---	S------
Getridof	10	O------	-	Q	-ABV--M	S------
Lcateheaviest	05	O------	-	Q	-ABV--M	S------
Weigh	09	-------	O	Q	-ABV--M	S------
Lcateheaviest	07	-------	O	-	-ABV--M	SQ-----
Seriate	03	ABVM---	O	-	-------	SQ-----
Seriate	01	ABVM---	O	-	-------	SQ-----
Lcateheaviest	05	ABVM---	O	-	-------	SQ-----
Weigh	09	ABV----	O	M	-------	SQ-----
Lcateheaviest	06	ABV----	O	M	-------	SQ-----
Getridof	10	ABV----	-	M	------O	SQ-----
Lcateheaviest	05	ABV----	-	M	------O	SQ-----
Weigh	09	AB-----	V	M	------O	SQ-----
Lcateheaviest	06	AB-----	V	M	------O	SQ-----
Getridof	10	AB-----	-	M	-----VO	SQ-----
Lcateheaviest	05	AB-----	-	M	-----VO	SQ-----
Weigh	09	A------	B	M	-----VO	SQ-----
Lcateheaviest	06	A------	B	M	-----VO	SQ-----
Getridof	10	A------	-	M	----BVO	SQ-----
Lcateheaviest	05	A------	-	M	----BVO	SQ-----
Weigh	09	-------	A	M	----BVO	SQ-----
Lcateheaviest	07	-------	A	-	----BVO	SQM----
Seriate	03	BVO----	A	-	-------	SQM----
Seriate	01	BVO----	A	-	-------	SQM----
Lcateheaviest	05	BVO----	A	-	-------	SQM----
Weigh	09	BV-----	A	O	-------	SQM----
Lcateheaviest	06	BV-----	A	O	-------	SQM----
Getridof	10	BV-----	A	-	---O---	SQM----
Lcateheaviest	05	BV-----	A	-	---O---	SQM----
Weigh	09	B------	A	V	---O---	SQM----
Lcateheaviest	06	B------	A	V	---O---	SQM----
Getridof	10	B------	A	-	--VO---	SQM----

Active Goal	Prod	Original Position	Balance L	Balance R	Intermediate Position	Final Position
Lcateheaviest	05	B------	A	-	--VO---	SQM----
Weigh	09	-------	A	B	--VO---	SQM----
Lcateheaviest	07	-------	A	-	--VO---	SQMB---
Seriate	03	VO-----	A	-	-------	SQMB---
Seriate	01	VO-----	A	-	-------	SQMB---
Lcateheaviest	05	VO-----	A	-	-------	SQMB---
Weigh	09	V------	A	O	-------	SQMB---
Lcateheaviest	06	V------	A	O	-------	SQMB---
Getridof	10	V------	A	-	O------	SQMB---
Lcateheaviest	05	V------	A	-	O------	SQMB---
Weigh	09	-------	A	V	O------	SQMB---
Lcateheaviest	07	-------	-	V	O------	SQMBA--
Seriate	03	O------	-	V	-------	SQMBA--
Seriate	02	O------	-	V	-------	SQMBA--
Lcateheaviest	05	O------	-	V	-------	SQMBA--
Weigh	09	-------	O	V	-------	SQMBA--
Lcateheaviest	07	-------	O	-	-------	SQMBAV-
Seriate	04	-------	-	-	-------	SQMBAVO (1234567)

Figure 25. Protocol for Warwick (8;4) on a weight seriation task.

PART III

GENERAL MODELS OF DEVELOPMENT

CHAPTER 8

PIAGET'S TRANSITION MECHANISM: EQUILIBRATION:

The most fundamental stage-independent concept in Piaget's theory is that of the equilibration model. This comprises a basic general theoretical construction which can be imposed on the whole of the developmental sequence outlined. It can be considered as a 'global conception of what the whole developmental process and its successive structural products are about' (Flavell, 1963, p.238). The model is proposed as an attempt by Piaget to cope with the problem of a mechanism, or system of rules, to explain the transition from one stage to the next. Phillips (1969) considers Piaget's conception of equilibration as the inspiration for the theory in the first place, and which now remains as its overarching principle.

The model of transition used by Piaget was borrowed from cybernetics, which is probably 'the most influential of all disciplines in affecting interdisciplinary currents' (Gallagher, 1977, p.28). Piaget often cites Ashby's (1956) account of cybernetics which was written for workers in the biological sciences. Piaget describes the discipline as an 'explanation of how one mechanism can direct others, or itself, by means of transmissions and the retroactive or anticipatory effects of information given '(Piaget, 1971, p.61). Piaget has in a number of articles, considered the

relationship of cybernetics to his general theoretical position.(Piaget, 1942; Piaget, 1953).

1. Cybernetics

Before looking more specifically at Piaget's model of equilibration, I shall discuss in a general way what various authors have considered cybernetics to entail, before tying the concepts more closely to Piaget's work.

The computer simulation of behavior, which has been the main emphasis of Part II of this thesis, has to a large extent emerged as part of the broader context of the discipline of cybernetic. Wiener (1948) introduced the word to refer to a new inter-disciplinary subject that he felt was emerging and which took as its subject matter, control and communication systems of all kinds. Wiener's book was eclectic and contained discussions of normal and abnormal physiological processes, as well as information, communication and feedback per se. Wiener's definition of cybernetics was 'the study of control and communication in the animal and the machine.' In practice the field to which this refers is so wide (incorporating parts of engineering, mathematics, logic, computer science, linguistics, biology, physiology, medicine, anthropology, psychology, sociology and economics, among other subjects) that it is not surprising that there have been differences of emphasis and subsequently disagreement as to what the term refers. Apter (1970) suggests that the field of cybernetics 'is perhaps the same as that to which the term "systems research" is beginning to be applied' (1970, p.31). He further suggests that 'cybernetics is in a sense more an attitude than a strict subject-matter: an attitude characterised by a passion for precision and a contempt for ideas that cannot be expressed in effective form, a desire to search for common organizational features in purposeful systems superficially unlike one another, a feeling that organisms are essentially constructible .... This attitude has in turn stimulated the emergence of a host of sophisticated concepts and practical techniques' (Apter, 1970, p.45 - 46).

Glushkov outlines a different outlook on the subject when he states: 'Cybernetics can in general, be divided into two large fields which form the basis of the discipline. These are one containing the general theory of information processing, and the other the theory and structure of information processors'(Glushkov, 1966, p.v).

The very foundation of the discipline of cybernetics is based upon a systems approach. It is primarily concerned with information flows in complex systems. 'Athough cybernetics has been applied primarily to mechanistic engineering problems, its model of feedback, control and regulation has a great deal of applicability for biological and social systems as well' (Kast & Rosenweig, 1972, p.17). It can be thought of as an attempt to understand organisms through making analogies to machines. Pask (1961, p.11) gives the theme of cybernetics as 'how systems regulate themselves, reproduce themselves, evolve and learn.' Cybernetics is theoretical in the sense that it has traditionally emphasized the understanding of a given process for its own sake, rather than for any practical applications that may emerge. Guilbaud (1959, p.6) makes an interesting comparison: 'The position of cybernetics is rather analogous to that of statistics, which finds uses in extremely diverse fields ... and derives its name from one specialized application - the study of human societies or states - with which it never lost contact, despite its rise to independence.'

George (1965, p.2) considers cybernetics as 'a search for precision.' This can be achieved by either mathematical, information processing, verbal, or even hardware models. He also emphasises an important aspect of cybernetics as the concept of negative feedback. Cybernetics is as yet a young discipline, and, like applied mathematics, cuts across the entrenched specializations of the sciences. How far the implications of this new approach will effect advances in science, and psychology in particular, remains to be seen.

2. Equilibration

Piaget believes that the three factors of maturation, experience, and social interaction do not suffice to explain cognitive development. Mental behaviors are observed that suggest the child contributes an organizational, or regulating quality, to development. Equilibration means that there is a requirement for these three concepts to be in balance. The organism comes to each situation with its own cognitive structures, which may be as diverse as the infant's reflex action patterns or the adult's accomodated structures. The discrepancy between the new objects or experiences and established structures sets up an imbalance, termed disequilibrium. Piaget maintains that growth of knowledge is due to the conflict between structures and events, the imbalance between what the individual knows and new experiences. If the new experiences are neither too novel nor too familiar, they may be assimilated and may influence or change those existent structures. In this case we may say that the structures have accomodated to the new information. This conflict creates the transition from one stage to another. The process by which structures change from one state to another is called equilibration, and the result of that process is a state of equilibrium. What these first three factors have in common is that the individual is passive. Something is done to him; his physiological system matures, or he is presented with physical or linguistic material to absorb. But intellectual development is not this passive and Piaget finds it necessary to call upon the factor of the individual's own activity. 'An individual learns to see the world as coherent, as structured, to the extent that he acts upon the world, transforms it, and succeeds in coordinating these actions and transformations' (Duckworth, 1964, p.172).

Piaget maintains that achieving a near equilibrium in a constantly changing situation is the goal of all human functions: biological, affective, and above all, cognitive. He defines equilibrium as a state in which 'all the virtual transformations compatible with the relationships of the

system compensate each other' (Piaget, 1957, p.41). Life itself is in a field of constant motion and 'intellectual operations proceed in terms of structures-of-the-whole. These structures denote the kinds of equilibrium towards which evolution in its entirety is striving at once organic, psychological and social, their roots reach down as far as biological morphogenesis itself' (Piaget, 1952a, p.256). Any state of equilibrium is entirely theoretical, and at least, only momentarily attainable. Piaget thinks that psychology should use the laws of equilibrium, which are primarily descriptions of the processes of change governing the states of equilibrium. Piaget further proposes that organisms tend toward equilibrium with the enviroment. The organism - whether a human being or some other form of life - tends to organize structure into coherent and stable patterns. His ways of dealing with the world tend towards a certain balance. He tries to develop structures which are effective in his interaction with reality. As Maier states (1965, p.90): 'Life is a progressive series of attempts to balance forms; each attempt rests upon previous acquisitions; and each - with increasing complexity - creates new forms which furnish in turn the foundations for later balance.'

Equilibration may also be conceptualized as the regulator of the relationship between assimilation and accomodation. It can be thought of as a mode or style of regulation. A given event may be assimilated or accomodated differently by different persons. Structures continually move toward a state of equilibrium, and when a state of relative equilibrium has been attained, the structure is sharper, more closely delineated, than it had been previously. But that very sharpness points up inconsistencies and gaps in the structure that had never been salient before. Each equilibrium state therefore carries with it the seeds of its own destruction, for the child's activities are thenceforth directed towards reducing those inconsistencies and closing those gaps.

Piaget (1977a) distinguishes three kinds of equilibrium.

Firstly there is the relationship between assimilation and accomodation. 'There is an equilibrium between the structures of the subject and the object; its structures accomodate to the new object being presented and the object is assimilated into the structures' (Piaget, 1977a, p.11). This is the fundamental form of equilibration. Secondly, there is an equilibrium among the subsystems of the S's schemes. Schemes are coordinated into subsystems, partial systems in relation to the total system of the S's knowledge. Systems can evolve at different rates, and as they evolve, there is a need for coordination of the two, or 'an equilibration of subsystems' (Piaget, 1977a, p.11). Thirdly, there has to be an equilibrium established between the parts of the S's knowledge and the totality of this knowledge. 'There is a constant differentiation of the totality of knowledge into the parts and an integration of the parts back into the whole. This equilibrium between differentiation and integration plays a fundamental biological role' (Piaget, 1977a, p.11).

Piaget gives two reasons for postulating this fourth factor of equilibration. The first relates to a need for coordination among the three other factors that have been mentioned. Secondly, 'in the act of knowing, the subject is active, and consequently, faced with an external disturbance, he will react in order to compensate and consequently he will tend towards equilibrium. Equilibrium, defined by active compensation, leads to reversibility. Operational reversibility is a model of an equilibrated system where a transformation in one direction is compensated by a transformation in the other direction' (Piaget, 1964, p.181). This has been restated more recently as 'in the construction of any operational or preoperational structure, a subject goes through much trial and error and many regulations that in a large part involve self-regulations. Self-regulations are the very nature of equilibration' (Piaget, 1977a, p.10).

To conclude, Piaget defines equilibration as 'the search for a better and better equilibrium in the sense of an

extended field, in the sense of an increase in the number of possible compositions, and in the sense of a growth in coherence' (Piaget, 1977a, p.12). We can see that even though it has not been possible to define the process of equilibration precisely, it is possible to get an overall general view of what Piaget considers a fairly basic process in cognitive development.

3. The Relationship between Cybernetics and Equilibration

Living systems are open systems in the cybernetic sense because they exchange information with their environment. If however the system accumulates information without subsequent organization, it will cease to exist an an independent system. The question is: How is it possible to have both stability and change?

The answer to this system lies in considering the system of equilibration as a series of feedback loops of increasing complexity between the living system and its environment. The essential aspect of such a process is progressive closure, which permits simultaneous extension of the environment. Such a system is only possible when the cognitive mechanisms function as specialized regulatory organs which control exchanges with the environment (Piaget, 1971, p.354).

The stability of this system is maintained by the achievement of equilibrium, which implies a conservation of the system. Operations are self-regulatory mechanisms and are therefore perfect regulators of the whole system. Piaget's equilibration system incorporates Von Bertalanffy's (1960) concept of equilibrium as a stable state in an open system. This equilibrium is characterized by both a stability and a mobility. (See the discussion between Von Bertalanffy and Piaget in Tanner and Inhelder (1960).)

Cognitive functions are specialized factors of self-regulation which control the exchanges underlying the entire system of behavior. Skinner's (1971) suggestion that mediating states of mind can be neglected betrays a lack of understanding of modern biology, which emphasises the role

of self-regulation. Piaget states: 'The notion of operation ... involves self-regulatory mechanisms. They are - in Ashby's sense, in his cybernetic terminology - the perfect regulations in that the outcome is anticipated before the act is actually carried out. The feedback, which at lower levels has incomplete reversibility, now becomes a feedback with perfect reversibility in the sense of inversion or reciprocity' (1977a, p.12). These self-regulatory mechanisms are 'important throughout every level of biological development' (Piaget, 1977a, p.9).

To understand equilibration finally, it is necessary to conceive of it as a process, leading to greater and greater probability that equilibrium will be reached. That this equilibrium will be reached is in the end assured by logical necessity. Furth outlines what is meant by this logical necessity:

> Piaget related equilibration to implication as an intrinsic form of causality within a total organization and to the lawfulness peculiar to dynamic systems as they are studied in cybernetics. Piaget likes to draw a parallel between the probability laws of equilibration and the transition from unpredictable to less predictable and ultimately logically necessary responses of strict implication. In this light the development of intelligence appears to an observer as a coordinated sequence of behavior such that a present stage is most likely, considering the immediately previous one, even though the intrinsic probability of its occurence, considered from levels far removed, appears only small. For Piaget, the concept of equilibration makes comprehensible why, in retrospect, alternatives or degrees of freedom for further evolution or development are progressively reduced with succeeding stages. (1969, p.207 - 208)

Piaget's theory of progressive stages of development is based on this notion of logical necessity. 'The transition from one stage to another is therefore an equilibration

in the most classical sense of the word. But since these displacements of the system are activities of the subject, and since each of these activites consist of correcting the one immediately preceding it, equilibration becomes a sequence of self-regulations whose retroactive processes finally result in operational reversibility. The latter then goes beyond simple probability to attain logical necessity' (Piaget, 1970, p. 725). The structures or systems of operations of the stages become necessary even though they are not so in the beginning. The seed to initiate and sustain this development is the process of equilibration.

PART IV

CONCLUSION

CONCLUSIONS

I have tried in this paper to consider Piaget's theory from two divergent directions. Firstly a very precise look at a specific area of children's behavior, and secondly a look at the general mechanisms of development and stage transition that Piaget proposes, specifically equilibration.

The programs shown in the Appendicies and discussed in Chapter 7 can be seen to model fairly accurately the limited area of observable behavior that has been investigated in this study. This particular target behavior was chosen because of its relative simplicity, but, it can be seen that even this results in a model of some complexity. To make a production amenable for observation, the problem are is restricted markedly. This is what has been done here in that a specific aspect of behavior, that of weight seriation, has been considered in some detail. Extensions to this work could involve enlarging the size of the production system to incorporate wider aspects of behavior. Furthermore, the procedures which have been based on a random model need to be investigated to discover any parameters controlling the behavior that could decrease the degree of randomness and increase the degree of prediction.

The representation of the problem space used in this study has been at a fairly simple level, involving arrays for the original and final positions and the balance. This simplicity of representation makes the problem a very useful

one for investigation. Compare it for instance with the complex representations needed for chess-playing programs.

Other important areas of possible research include a more intensive investigation of whether the production systems and base strategies, which have formed the basis of the model, are actually "in" the child ready to be brought to conceptual awareness. This has lead to criticisms being levelled at this type of model concerning whether the subject actually carries a "program" around in his head or not. An analogy is perhaps appropriate here. Twice in my education I have been introduced to different models of the structure of the atom. The latter one was considerably more sophisticated but neither are "correct" in any absolute sense of the word. They were both convenient fictions for communicating a model of atomic structure. In the same way the models presented here are convenient fictions for communicating a model of the mind's structure. No doubt they will become increasingly sophisticated, but they will never be completely correct models.

The consideration of general mechanisms of development has not been done in great detail. What was originally envisioned was a mathematically-based cybernetic system that would be capable of being run as a program, that could demonstrate the transition from one stage to another. Piaget's concept of equilibration remains a general model and would need to be more formalized before it could be used as a basis for any development. Grodins (1965) talks about a computer simulation of cybernetic systems in medicine and biology. With the biological foundation of Piaget's theory, his work could lay the groundwork for the simulation of Piaget's equilibration concept, conceived of as a cybernetic system.

Other transition mechanisms have been posited which can describe the child's movement from one stage to the next. These have been outlined in more detail in Appendix VI. Firstly there is an approach which considers the transition mechanism as a "higher order" production system which controls other production systems (Klahr and Wallace, 1976).

Secondly the work of Pascual-Leone (1970) has exciting possibilities in that he considers the determining factor in any particular stage to be the size of the central processor, or M-space. This ties neatly in with Piaget's concept of a "field of equilibrium". Elements of all these approaches may be useful in the final formulation of a transition mechanism. Appel and Goldberg's (1977) book illustrates that researchers are attempting to more closely understand the concept of equilibration and its applications.

How does the work that has been presented here tie in with the original problems expressed in Chapter 1? The first problem concerns a lack of direction in research. For a large body of researchers, Piaget's theory has represented a direction, a specific (if broad) area to direct their attention. The development of the child's cognitive ability is an important area, for both theoretical development and practical benefits. Furthermore Piaget believes that his theory represents a coherent whole, in spite of the surface diversity of its expression. This leads to the second problem Newell discusses - that of putting all the results of the various experiments together. It is here that the most exciting possibilities of the methods I have been advocating lie. The development of an efficient symbol processor, the computer, has given us a means of "putting it all together". For example, a model of motivational factors could be divised and superimposed over the seriation structure, leading to some predictions about the effects of interaction between the two. More complete environmental representations could lead to such factors as the relative speed of descent of the balance pans leading to a possible hypothesis about the difference between the weights. Developments of programs of other specific cognitive abilities may lead to further work on interaction effects. The phenomena being observed are complex, and complex models will be needed to simulate them.

Before the widespread use of the computer, the only material available for the psychologist to work on were animals and humans. The use of animals has considerable

limitations, eventually leading to a "ratamorphic" conception of psychology. Humans are satisfactory in most respects, but ethical limitations prevent excessive manipulation of variables to determine effects. Now we have available a new experimental field, whose theoretical degree of complexity may be similar to man's, and who can be experimented on with no ethical problems.

It would be naive to suggest that in a volatile science such as psychology, there should be quite distinct periods of analytic fact-finding followed by distinct periods of synthesis. The two should occur together. But it is important to realize that analytical research has gone on for a considerably long period of time. In conjunction with this continuing research, attempts should be made to synthesize a model of the "whole man". Computer science technology is almost at the point where preliminary attempts to do this are possible. It is worth noting that the level of software development in computer science in well in front of its application in psychology.

It is interesting to note that Piaget's theory already has a strong information processing orientation, with his emphasis on operations, but diverges very distinctly from this with his complete lack of emphasis on the processing of representations. The development of adequate representational models is an important area for further research, and may need to precede a more complete modelling of cognitive ability. The concepts of information processing utilized in this paper offer a powerful methodology for developing models of the human information processor. We are still some way from realizing the full implications of the statement that the development of the computer will be as significant an event for psychology as the development of the microscope was for biology.

PART V

APPENDICIES AND REFERENCES

APPENDIX I

STAGE 1A WEIGHT SERIATION PROGRAM

```
                CARD FILE := CARD.

PROGRAM STAGE1;

(*                                                    *)
(*                                                    *)
(*          STAGE 1A WEIGHT SERIATION                 *)
(*          ----- -- ------ --------                  *)
(*                                                    *)
(*                                                    *)


(* SIMULATION OF STAGE 1 WEIGHT SERIATION BEHAVIOR *)
(* INPUT IS A LIST OF WEIGHTS FOR ITEMS TO BE SERIATED *)
(* LIST MAY BE ANY LENGTH, UP TO 20, SEPERATED BY COMMAS *)

CONST
   MAX = 20;       (* MAX. NO. OF ITEMS WHICH CAN BE HANDLED *)
   GSXMAX = 10;    (* MAX. DEPTH OF GOAL STACK *)

TYPE
   GOALSET = (SERIATE, FINDHEAVYLIGHT,WEIGH, NONE);
   BALSTATE = (EMPTY, LEFT, RIGHT, BOTH);
   BALPOS = LEFT .. RIGHT;

VAR
   PO,PF : ARRAY [1 .. MAX] OF REAL;  (* ORIGINAL & FINAL POSITIONS *)
   NPO,          (* NO. OF ITEMS IN ORIGINAL POSITION *)
   NPF,          (* NO. OF ITEMS IN FINAL POSITION *)
   NORIG,        (* NO. OF ITEMS ORIGINALLY PRESENTED *)
   K : 0 .. MAX;
   BAL : BALSTATE;
   PB : ARRAY [BALPOS] OF REAL;  (* BALANCE PANS *)
   GOALSTACK : ARRAY [0 .. GSXMAX ] OF GOALSET;
   GSX : 0 .. GSXMAX;
```

003:0000:1 (repeated for each line)

MASSEY UNIVERSITY

```
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
004:0000:1
004:0001:3
004:0002:4
004:0004:0
004:0008:4
004:0009:2
004:0009:2
004:0009:2
003:0000:1
003:0000:1
003:0000:1
006:0000:1
006:0001:2
006:0003:0
006:0003:2
006:0008:5
006:000D:3
006:000E:1
006:000E:1
006:000E:1
003:0000:1
003:0000:1
007:0000:1
007:0002:4
007:0003:2
007:0004:0
007:0004:4
007:0004:4
007:0004:4
003:0000:1
003:0000:1
008:0000:1
008:0000:5
008:0000:5
008:0002:0
008:0008:3
008:000B:0
008:000C:1
008:000D:0
008:000E:2
008:0017:3
008:0018:2
008:0018:2
```

```
    (*                                  *)
    (*                                  *)
    (*        BASIC PROCEDURES          *)
    (*                                  *)
    (*                                  *)

PROCEDURE GOAL(NEWGOAL: GOALSET);
    (* ---- *)
  (* PUSH A NEW GOAL ONTO THE STACK *)
BEGIN
  IF GSX < GSXMAX THEN BEGIN
    GSX := SUCC(GSX);
    GOALSTACK[GSX] := NEWGOAL;
  END ELSE HALT;
END (* GOAL *);

PROCEDURE SUCCEED(OLDGOAL: GOALSET);
    (* ------- *)
  (* POP A GOAL OFF THE GOAL STACK *)
BEGIN
  IF GSX > 0 THEN BEGIN
    IF GOALSTACK[GSX] = OLDGOAL THEN
    GSX := GSX - 1
    ELSE HALT;
  END ELSE HALT;
END (* SUCCEED *);

PROCEDURE INITIALISE;
    (* ---------- *)
BEGIN
  GSX := 0;  GOAL(NONE);  GOAL(SERIATE);
  BAL := EMPTY;
  NPF := 0;
END (* INITIALISE *);

PROCEDURE INPUTOFDATA;
    (* ---------- *)
BEGIN
  NPO := 0;
  REPEAT
    NPO := NPO + 1;
    READ(INPUT, PO[NPO]);
  UNTIL EOF(INPUT);
  NPO := NPO - 1;
  NORIG := NPO;
  IF NPO > MAX THEN BEGIN
    WRITELN(OUTPUT, 'TOO MANY ITEMS', NPO, MAX);
    NPO := MAX;
  END;
```

```
    WRITELN(OUTPUT, 'STAGE 1 WEIGHT SERIATION');                    008:0018:2
    WRITELN(OUTPUT, '------------------------');                    008:001E:2
   WRITELN(OUTPUT);                                                 008:0024:2
   WRITELN(OUTPUT);                                                 008:0028:0
    WRITELN(OUTPUT, 'ORIGINAL POSITION:');                          008:002B:4
    FOR K := 1 TO NPO DO WRITE(OUTPUT, PO[K]);                      008:0031:4
    WRITELN(OUTPUT);                                                008:003D:3
    WRITELN(OUTPUT);                                                008:0041:1
    WRITELN(OUTPUT);                                                008:0044:5
END (* INPUTOFDATA *);                                              008:0048:3
                                                                    008:005E:4
PROCEDURE CURRENTFINALPOS;                                          008:005E:4
      (* ---------------- *)                                        008:005E:4
BEGIN                                                               003:0000:1
    WRITELN(OUTPUT, 'CURRENT FINAL POSITION:');                     003:0000:1
    FOR K := 1 TO NORIG DO WRITE(OUTPUT, PF[K]);                    00C:0000:1
    WRITELN(OUTPUT);                                                00C:0006:1
    WRITELN(OUTPUT);                                                00C:0012:0
    WRITELN(OUTPUT);                                                00C:0015:4
    WRITELN(OUTPUT);                                                00C:0019:2
END (* CURRENTFINALPOS *);                                          00C:001D:0
                                                                    00C:0020:4
PROCEDURE FINALPOS;                                                 00C:002C:4
      (* --------- *)                                              00C:002C:4
BEGIN                                                               00C:002C:4
    WRITELN(OUTPUT);                                                003:0000:1
    WRITELN(OUTPUT);                                                003:0000:1
    WRITELN(OUTPUT, 'FINAL POSITION:');                             00D:0000:1
    FOR K := 1 TO NPF DO WRITE(OUTPUT, PF[K]);                      00D:0003:5
    WRITELN(OUTPUT);                                                00D:0007:3
END (* FINALPOS *);                                                 00D:000D:3
                                                                    00D:0019:2
                                                                    00D:001D:0
                                                                    00D:0027:4
                                                                    00D:0027:4
                                                                    00D:0027:4
                                                                    00D:0027:4
                                                                    00D:0027:4
                                                                    00D:0027:4
                                                                    00D:0027:4
                                                                    00D:0027:4
                                                                    00D:0027:4
                                                                    00D:0027:4
                                                                    00D:0027:4
                                                                    00D:0027:4
                                                                    00D:0027:4
                                                                    00D:0027:4
                                                                    00D:0027:4
                                                                    00D:0027:4
                                                                    00D:0027:4
                                                                    00D:0027:4
```

MASSEY UNIVERSITY

```
00D:0027:4
00D:0027:4
00D:0027:4
00D:0027:4
00D:0027:4
00D:0027:4
00D:0027:4
00D:0027:4
00D:0027:4
00D:0027:4
00D:0027:4
00D:0027:4
00D:0027:4
00D:0027:4
00D:0027:4
00D:0027:4
003:0000:1
003:0000:1
003:0000:1
00E:0000:1
00E:000D:5
00E:0015:3
00E:0015:3
00E:0017:5
00E:0019:1
00E:0019:1
00E:001A:2
00E:001A:2
00E:001C:3
00E:001E:4
00E:001E:4
00E:001E:4
003:0000:1
003:0000:1
003:0000:1
00F:0000:1
00F:000D:5
00F:0015:3
00F:0015:3
00F:0016:2
00F:0018:4
00F:0018:4
00F:001A:0
00F:001B:1
00F:001B:1
00F:001D:1
00F:001E:2
00F:0020:4
00F:0020:4
00F:0020:4
00F:0020:4
00F:0020:4
00F:0020:4
00F:0020:4
00F:0020:4
00F:0020:4
00F:0020:4
00F:0020:4
00F:0020:4
```

```
      (*                                  *)
      (*                                  *)
      (*          PROCEDURES FOR          *)
      (*        MOVEMENT OF WEIGHTS       *)
      (*                                  *)
      (*                                  *)

PROCEDURE POTOPB(VAR NPO: INTEGER; BPOS: BALPOS);
      (* ------ *)
(* MOVEMENT OF WEIGHTS FROM THE ORIGINAL POSITION TO THE BALANCE *)
BEGIN
   WRITELN(OUTPUT, ' ':45, 'MOVE PO[', NPO:2,'] TO PB[', BPOS:5,
           '] (', PO[NPO], ')');

   PB[BPOS] := PO[NPO];
   NPO := NPO - 1;

   IF BAL = EMPTY THEN
      BAL := BPOS
   ELSE BAL := BOTH;
END (* POTOPB *);

PROCEDURE PBTOPF(BPOS: BALPOS; FPOS: INTEGER);
      (* ------ *)
(* MOVEMENT OF WEIGHTS FROM THE BALANCE PANS TO THE FINAL POSITION *)
BEGIN
   WRITELN(OUTPUT, ' ':45, 'MOVE PB[', BPOS:5, '] TO PF[', FPOS:2,
           '] (', PB[BPOS], ')');

      NPF := FPOS;
      PF[NPF] := PB[BPOS];

   IF BAL = BOTH THEN BEGIN
      IF BPOS = LEFT THEN
         BAL := RIGHT
      ELSE BAL := LEFT;
   END ELSE BAL := EMPTY;
END (* PBTOPF *);
```

MASSEY UNIVERSITY

```
00F:0020:4
00F:0020:4
00F:0020:4
00F:0020:4
00F:0020:4
00F:0020:4
00F:0020:4
00F:0020:4
00F:0020:4
00F:0020:4
00F:0020:4
00F:0020:4
00F:0020:4
00F:0020:4
003:0000:1
003:0000:1
010:0000:1
010:0009:1
010:0009:1
010:0009:1
010:0009:1
010:000A:2
010:000A:2
010:0012:1
010:0013:0
010:0013:3
010:0013:3
010:0013:3
010:0013:3
010:0014:4
010:0014:4
010:001C:3
010:0025:3
010:0026:2
010:0026:2
010:002E:4
010:002E:4
010:002E:4
003:0000:1
003:0000:1
011:0000:1
011:0009:2
011:0009:2
011:0009:2
011:0009:2
011:000A:3
011:000A:3
011:0012:3
011:0013:3
011:0014:0
011:0014:0
011:0014:0
011:0014:0
011:0014:0
011:0014:0
011:0014:0
011:0014:0
011:0014:0
011:0014:0
```

```
      (*                                  *)
      (*                                  *)
      (*          PRODUCTION SYSTEM       *)
      (*                                  *)
      (*                                  *)

PROCEDURE DOSERIATE;
      (* --------- *)
BEGIN
   WRITELN(OUTPUT, 'GOAL = ', GOALSTACK[GSX]);

   (* PRODUCTION 1 *)
   (* ---------- - *)
   IF NPO > 0 THEN BEGIN

   WRITELN(OUTPUT, ' ':25, 'PRODUCTION 01');
   GOAL(FINDHEAVYLIGHT);
   END ELSE

   (* PRODUCTION 2 *)
   (* ---------- - *)
   IF NPO = 0 THEN BEGIN

   WRITELN(OUTPUT,' ':25, 'PRODUCTION 02');
   WRITELN(OUTPUT,'SUCCEED ', GOALSTACK[GSX]);
   SUCCEED(SERIATE);
   END;
END (* DOSERIATE *);

PROCEDURE DOFINDHEAVYLIGHT;
      (* --------------- *)
BEGIN
   WRITELN(OUTPUT, ' GOAL = ', GOALSTACK[GSX]);

   (* PRODUCTION 3 *)
   (* ---------- - *)
   IF BAL = EMPTY THEN BEGIN

   WRITELN(OUTPUT,' ':25, 'PRODUCTION 03');
   GOAL(WEIGH);
   END ELSE
```

MASSEY UNIVERSITY

```
011:0014:0
011:0014:0
011:0014:0
011:0014:0
011:0014:0
011:0014:0
011:0014:0
011:0014:0
011:0014:0
011:0014:0
011:0015:4
011:0015:4
011:001D:4
011:001E:5
011:0020:3
011:0021:0
011:0022:5
011:0022:5
011:002C:0
011:002C:4
011:002D:3
011:002E:0
011:002E:0
011:002E:0
011:002E:0
011:002F:2
011:002F:2
011:0037:2
011:0039:0
011:003A:4
011:003C:3
011:003D:0
011:003E:5
011:0040:3
011:0040:3
011:0049:4
011:004A:2
011:004B:1
011:004B:1
011:0057:4
011:0057:4
011:0057:4
003:0000:1
003:0000:1
012:0000:1
012:0009:2
012:0009:2
012:0009:2
012:0009:2
012:000A:3
012:000A:3
012:0012:3
012:0013:5
012:0015:2
012:001E:3
012:001F:3
012:0020:0
012:0020:0
012:0020:0
012:0020:0
```

```
    (* PRODUCTION 4 *)
    (* ---------- - *)
    IF BAL IN [LEFT,RIGHT] THEN BEGIN

    WRITELN(OUTPUT,' ':25, 'PRODUCTION 04');
       IF BAL = LEFT THEN BEGIN
       PBTOPF(LEFT,NPF+1);
       END ELSE BEGIN
       PBTOPF(RIGHT,NPF+1);
       END;
    WRITELN(OUTPUT, ' SUCCEED  ', GOALSTACK[GSX]);
    CURRENTFINALPOS;
    SUCCEED(FINDHEAVYLIGHT);
    END ELSE

    (* PRODUCTION 5 *)
    (* ---------- - *)
    IF BAL = BOTH THEN BEGIN

    WRITELN(OUTPUT,' ':25, 'PRODUCTION 05');
    IF PB[LEFT] > PB[RIGHT] THEN BEGIN
    PBTOPF(LEFT,NPF+1);
    PBTOPF(RIGHT,NPF+1);
    END ELSE BEGIN
    PBTOPF(RIGHT,NPF+1);
    PBTOPF(LEFT,NPF+1);
    END;
    WRITELN(OUTPUT, ' SUCCEED  ', GOALSTACK[GSX]);
    CURRENTFINALPOS;
    SUCCEED(FINDHEAVYLIGHT);
    END;
END (* DOFINDHEAVYLIGHT *);

PROCEDURE DOWEIGH;
      (* ------- *)
BEGIN
    WRITELN(OUTPUT, '  GOAL = ', GOALSTACK[GSX]);

    (* PRODUCTION 6 *)
    (* ---------- - *)
    IF NPO > 1 THEN BEGIN

    WRITELN(OUTPUT, ' ':25, 'PRODUCTION 06');
    POTOPB(NPO,LEFT);
    POTOPB(NPO,RIGHT);
    WRITELN(OUTPUT, '  SUCCEED  ', GOALSTACK[GSX]);
    SUCCEED(WEIGH);
    END ELSE
```

MASSEY UNIVERSITY

```
    (* PRODUCTION 7 *)                                                    012:0020:0
    (* ---------- - *)                                                    012:0020:0
    IF NPO = 1 THEN BEGIN                                                 012:0020:0
                                                                          012:0020:0
    WRITELN(OUTPUT, ' ':25, 'PRODUCTION 07');                             012:0020:0
    POTOPB(NPO,LEFT);                                                     012:0020:0
    WRITELN(OUTPUT, '  SUCCEED  ', GOALSTACK[GSX]);                      012:0020:0
    SUCCEED(WEIGH);                                                       012:0021:1
    END;                                                                  012:0021:1
END (* DOWEIGH *);                                                       012:0029:1
                                                                          012:002A:3
                                                                          012:0033:4
                                                                          012:0034:4
                                                                          012:0034:4
                                                                          012:003E:4
                                                                          012:003E:4
                                                                          012:003E
```

MASSEY UNIVERSITY

```
                                                                                012:003E:4
                                                                                012:003E:4
                                                                                012:003E:4
                                                                                012:003E:4
                                                                                012:003E:4
                                                                                012:003E:4
                                                                                012:003E:4
                                                                                012:003E:4
     (*                                *)                                       012:003E:4
     (*                                *)                                       012:003E:4
     (*           MAIN PROGRAM         *)                                       012:003E:4
     (*                                *)                                       012:003E:4
     (*                                *)                                       012:003E:4
                                                                                012:003E:4
                                                                                012:003E:4
BEGIN                                                                           012:003E:4
   INITIALISE;                                                                  003:0000:1
   INPUTOFDATA;                                                                 003:0000:5
                                                                                003:0001:3
   (* PROCESS *)                                                               003:0001:3
   REPEAT                                                                       003:0001:3
      CASE GOALSTACK[GSX] OF                                                    003:0001:3
                                                                                003:0002:4
      SERIATE:  DOSERIATE;                                                      003:0002:4
                                                                                003:0003:5
      FINDHEAVYLIGHT:  DOFINDHEAVYLIGHT;                                        003:0003:5
                                                                                003:0005:0
      WEIGH:  DOWEIGH;                                                          003:0005:0
                                                                                003:0006:1
      NONE:                                                                     003:0006:1
                                                                                003:0006:1
      END;                                                                      003:0006:1
   UNTIL GOALSTACK[GSX] = NONE;                                                 003:000D:0
                                                                                003:000E:4
   FINALPOS;  (* PRINTING OF FINAL POSITIONS *)                                003:000E:4
                                                                                003:000F:2
END.                                                                            003:000F:2
================================COMPILATION SUMMARY=============================
 FILE TITLE: CARD.
ILE TITLE: WITH.
) BY COMPILER

DE SEGMENTS   VALUE ARRAYS     EST. STACK    EST. MEMORY   DISK SECTORS
        12              1
       539             64             51           1366             49
================================================================================
```


MASSEY UNIVERSITY
LIBRARY

```
STAGE 1 WEIGHT SERIATION
------------------------

ORIGINAL POSITION:
     180.000000     179.000000     182.700000     181.800000     177.200000     180.800000     178.100000

GOAL =  SERIATE
                          PRODUCTION 01
 GOAL =  FINDHEAVYLIGHT
                          PRODUCTION 03
  GOAL =  WEIGH
                          PRODUCTION 06
                                                 MOVE PO[ 7] TO PB[ LEFT]  (      178.100000)
                                                 MOVE PO[ 6] TO PB[RIGHT]  (      180.800000)
  SUCCEED   WEIGH
 GOAL =  FINDHEAVYLIGHT
                          PRODUCTION 05
                                                 MOVE PB[RIGHT] TO PF[ 1]  (      180.800000)
                                                 MOVE PB[ LEFT] TO PF[ 2]  (      178.100000)
 SUCCEED   FINDHEAVYLIGHT
CURRENT FINAL POSITION:
     180.800000     178.100000  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00


GOAL =  SERIATE
                          PRODUCTION 01
 GOAL =  FINDHEAVYLIGHT
                          PRODUCTION 03
  GOAL =  WEIGH
                          PRODUCTION 06
                                                 MOVE PO[ 5] TO PB[ LEFT]  (      177.200000)
                                                 MOVE PO[ 4] TO PB[RIGHT]  (      181.800000)
  SUCCEED   WEIGH
 GOAL =  FINDHEAVYLIGHT
                          PRODUCTION 05
                                                 MOVE PB[RIGHT] TO PF[ 3]  (      181.800000)
                                                 MOVE PB[ LEFT] TO PF[ 4]  (      177.200000)
 SUCCEED   FINDHEAVYLIGHT
CURRENT FINAL POSITION:
     180.800000     178.100000     181.800000     177.200000  0.0000000E+00  0.0000000E+00  0.0000000E+00


GOAL =  SERIATE
                          PRODUCTION 01
 GOAL =  FINDHEAVYLIGHT
                          PRODUCTION 03
  GOAL =  WEIGH
                          PRODUCTION 06
                                                 MOVE PO[ 3] TO PB[ LEFT]  (      182.700000)
                                                 MOVE PO[ 2] TO PB[RIGHT]  (      179.000000)
  SUCCEED   WEIGH
 GOAL =  FINDHEAVYLIGHT
                          PRODUCTION 05
                                                 MOVE PB[ LEFT] TO PF[ 5]  (      182.700000)
                                                 MOVE PB[RIGHT] TO PF[ 6]  (      179.000000)
 SUCCEED   FINDHEAVYLIGHT
CURRENT FINAL POSITION:
```

```
     180.800000     178.100000     181.800000     177.200000     182.700000     179.000000  0.0000000E+00

GOAL =  SERIATE
 GOAL =  FINDHEAVYLIGHT                PRODUCTION 01
  GOAL =  WEIGH                        PRODUCTION 03
                                       PRODUCTION 07
                                                      MOVE PO[ 1] TO PB[ LEFT]  (     180.000000)
  SUCCEED   WEIGH
 GOAL =  FINDHEAVYLIGHT
                                       PRODUCTION 04
                                                      MOVE PB[ LEFT] TO PF[ 7]  (     180.000000)
 SUCCEED   FINDHEAVYLIGHT
CURRENT FINAL POSITION:
     180.800000     178.100000     181.800000     177.200000     182.700000     179.000000     180.000000

GOAL =  SERIATE
                                       PRODUCTION 02
SUCCEED   SERIATE

FINAL POSITION:
     180.800000     178.100000     181.800000     177.200000     182.700000     179.000000     180.000000
```

APPENDIX II

STAGE 1B WEIGHT SERIATION PROGRAM

MASSEY UNIVERSITY

CARD FILE := CARD.

```
PROGRAM STAGE1;

(*                                                    *)
(*                                                    *)
(*          STAGE 1B WEIGHT SERIATION                 *)
(*          ----- -- ------ ---------                 *)
(*                                                    *)
(*                                                    *)


(* SIMULATION OF STAGE 1 WEIGHT SERIATION BEHAVIOR *)
(* INPUT IS A LIST OF WEIGHTS FOR ITEMS TO BE SERIATED *)
(* LIST MAY BE ANY LENGTH, UP TO 20, SEPERATED BY COMMAS *)

CONST
   MAX = 20;       (* MAX. NO. OF ITEMS WHICH CAN BE HANDLED *)
   GSXMAX = 10;       (* MAX. DEPTH OF GOAL STACK *)

TYPE
   GOALSET = (SERIATE, FINDHEAVYLIGHT,WEIGH, NONE);
   BALSTATE = (EMPTY, LEFT, RIGHT, BOTH);
   BALPOS = LEFT .. RIGHT;

VAR
   PO,PF : ARRAY [1 .. MAX] OF REAL;  (* ORIGINAL & FINAL POSITIONS *)
   NPO,          (* NO. OF ITEMS IN ORIGINAL POSITION *)
   NPF,          (* NO. OF ITEMS IN FINAL POSITION *)
   NORIG,        (* NO. OF ITEMS ORIGINALLY PRESENTED *)
   K : 0 .. MAX;
   BAL : BALSTATE;
   PB : ARRAY [BALPOS] OF REAL;  (* BALANCE PANS *)
   GOALSTACK : ARRAY [0 .. GSXMAX ] OF GOALSET;
   GSX : 0 .. GSXMAX;
```

003:0000:1 (repeated)

MASSEY UNIVERSITY

```
003:0000:1
003:0000:1
003:0000:1
```

MASSEY UNIVERSITY

```
008:0018:2
008:0018:2
008:0018:2
008:0018:2
008:0018:2
008:0018:2
008:0018:2
008:001E:2
008:0024:2
008:0028:0
008:002B:4
008:0031:4
008:003D:3
008:0041:1
008:0044:5
008:0048:3
008:005E:4
008:005E:4
008:005E:4
003:0000:1
003:0000:1
00C:0000:1
00C:0006:1
00C:0012:0
00C:0015:4
00C:0019:2
00C:001D:0
00C:0020:4
00C:002C:4
00C:002C:4
00C:002C:4
003:0000:1
003:0000:1
00D:0000:1
00D:0003:5
00D:0007:3
00D:000D:3
00D:0019:2
00D:001D:0
00D:0027:4
00D:0027:4
00D:0027:4
```

MASSEY UNIVERSITY

```
00D:0027:4
00D:0027:4
00D:0027:4
00D:0027:4
00D:0027:4
00D:0027:4
00D:0027:4
      (*                                *)   00D:0027:4
      (*                                *)   00D:0027:4
      (*          PROCEDURES FOR        *)   00D:0027:4
      (*       MOVEMENT OF WEIGHTS      *)   00D:0027:4
      (*                                *)   00D:0027:4
      (*                                *)   00D:0027:4
                                             00D:0027:4
PROCEDURE POTOPB(VAR NPO: INTEGER; BPOS: BALPOS);   00D:0027:4
    (* ------ *)                               003:0000:1
(* MOVEMENT OF WEIGHTS FROM THE ORIGINAL POSITION TO THE BALANCE *)   003:0000:1
BEGIN                                          003:0000:1
   WRITELN(OUTPUT, ' ':45, 'MOVE PO[', NPO:2,'] TO PB[', BPOS:5,   00E:0000:1
           '] (', PO[NPO], ')');               00E:000D:5
                                               00E:0015:3
   PB[BPOS] := PO[NPO];                        00E:0015:3
   NPO := NPO - 1;                             00E:0017:5
                                               00E:0019:1
   IF BAL = EMPTY THEN                         00E:0019:1
      BAL := BPOS                              00E:001A:2
   ELSE BAL := BOTH;                           00E:001A:2
END (* POTOPB *);                              00E:001C:3
                                               00E:001E:4
                                               00E:001E:4
PROCEDURE PBTOPF(BPOS: BALPOS; FPOS: INTEGER);   00E:001E:4
    (* ------ *)                               003:0000:1
(* MOVEMENT OF WEIGHTS FROM THE BALANCE PANS TO THE FINAL POSITION *)   003:0000:1
BEGIN                                          003:0000:1
   WRITELN(OUTPUT, ' ':45, 'MOVE PB[', BPOS:5, '] TO PF[', FPOS:2,   00F:0000:1
           '] (', PB[BPOS], ')');              00F:000D:5
                                               00F:0015:3
      NPF := FPOS;                             00F:0015:3
      PF[NPF] := PB[BPOS];                     00F:0016:2
                                               00F:0018:4
   IF BAL = BOTH THEN BEGIN                    00F:0018:4
      IF BPOS = LEFT THEN                      00F:001A:0
         BAL := RIGHT                          00F:001B:1
   ELSE BAL := LEFT;                           00F:001B:1
   END ELSE BAL := EMPTY;                      00F:001D:1
END (* PBTOPF *);                              00F:001E:2
                                               00F:0020:4
                                               00F:0020:4
PROCEDURE POTOPF(VAR NPO: INTEGER; FPOS: INTEGER);   00F:0020:4
    (* ------ *)                               003:0000:1
(* MOVEMENT OF WEIGHTS FROM THE ORIGINAL POSITION      003:0000:1
                          TO THE FINAL POSITION *)     003:0000:1
                                               003:0000:1
BEGIN                                          003:0000:1
   WRITELN(OUTPUT, ' ':45, 'MOVE PO[', NPO:2,'] TO PF[', FPOS:2,   010:0000:1
           '] (', PO[NPO], ')');               010:000D:0
                                               010:0014:4
      NPF := FPOS;                             010:0014:4
      PF[NPF] := PO[NPO];                      010:0015:3
      NPO := NPO - 1;
```

MASSEY UNIVERSITY

```
010:001B:4
010:001B:4
010:001B:4
010:001B:4
010:001B:4
010:001B:4
010:001B:4
010:001B:4
010:001B:4
010:001B:4
010:001B:4
010:001B:4
010:001B:4
010:001B:4
010:001B:4
010:001B:4
003:0000:1
003:0000:1
011:0000:1
011:0009:1
011:0009:1
011:0009:1
011:0009:1
011:000A:2
011:000A:2
011:0012:1
011:0013:0
011:0013:3
011:0013:3
011:0013:3
011:0013:3
011:0014:4
011:0014:4
011:001C:4
011:0025:5
011:0026:4
011:0027:1
011:0027:1
011:0027:1
011:0027:1
011:0028:2
011:0028:2
011:0030:2
011:0032:2
011:003B:3
011:003C:2
011:003C:2
011:0048:4
011:0048:4
011:0048:4
003:0000:1
003:0000:1
012:0000:1
012:0009:2
012:0009:2
012:0009:2
012:0009:2
012:000A:3
012:000A:3
012:0012:3
```

```
      (*                                  *)
      (*                                  *)
      (*          PRODUCTION SYSTEM       *)
      (*                                  *)
      (*                                  *)

PROCEDURE DOSERIATE;
      (* --------- *)
BEGIN
   WRITELN(OUTPUT, 'GOAL = ', GOALSTACK[GSX]);

   (* PRODUCTION 1 *)
   (* ---------- - *)
   IF NPO > 1 THEN BEGIN

   WRITELN(OUTPUT, ' ':25, 'PRODUCTION 01');
   GOAL(FINDHEAVYLIGHT);
   END ELSE

   (* PRODUCTION 2 *)
   (* ---------- - *)
   IF NPO = 0 THEN BEGIN

   WRITELN(OUTPUT,' ':25, 'PRODUCTION 02');
   WRITELN(OUTPUT,'SUCCEED  ', GOALSTACK[GSX]);
   SUCCEED(SERIATE);
   END ELSE

   (* PRODUCTION 8 *)
   (* ---------- - *)
   IF NPO = 1 THEN BEGIN

   WRITELN(OUTPUT,' ':25, 'PRODUCTION 08');
   POTOPF(NPO,NPF+1);
   WRITELN(OUTPUT,'SUCCEED  ', GOALSTACK[GSX]);
   SUCCEED(SERIATE);
   END;
END (* DOSERIATE *);

PROCEDURE DOFINDHEAVYLIGHT;
      (* --------------- *)
BEGIN
   WRITELN(OUTPUT, ' GOAL = ', GOALSTACK[GSX]);

   (* PRODUCTION 3 *)
   (* ---------- - *)
   IF BAL = EMPTY THEN BEGIN

   WRITELN(OUTPUT,' ':25, 'PRODUCTION 03');
   GOAL(WEIGH);
   END ELSE
```

MASSEY UNIVERSITY

```
012:0014:0
012:0014:0
012:0014:0
012:0014:0
012:0014:0
012:0014:0
012:0014:0
012:0014:0
012:0014:0
012:0014:0
012:0015:4
012:0015:4
012:001D:4
012:001E:5
012:0020:3
012:0021:0
012:0022:5
012:0022:5
012:002C:0
012:002C:4
012:002D:3
012:002E:0
012:002E:0
012:002E:0
012:002E:0
012:002F:2
012:002F:2
012:0037:2
012:0039:0
012:003A:4
012:003C:3
012:003D:0
012:003E:5
012:0040:3
012:0040:3
012:0049:4
012:004A:2
012:004B:1
012:004B:1
012:0057:4
012:0057:4
012:0057:4
003:0000:1
003:0000:1
013:0000:1
013:0009:2
013:0009:2
013:0009:2
013:0009:2
013:000A:3
013:000A:3
013:0012:3
013:0013:5
013:0015:2
013:001E:3
013:001F:3
013:0020:0
013:0020:0
013:0020:0
013:0020:0
```

```
    (* PRODUCTION 4 *)
    (* ---------- - *)
    IF BAL IN [LEFT,RIGHT] THEN BEGIN

    WRITELN(OUTPUT,' ':25, 'PRODUCTION 04');
       IF BAL = LEFT THEN BEGIN
       PBTOPF(LEFT,NPF+1);
       END ELSE BEGIN
       PBTOPF(RIGHT,NPF+1);
       END;
    WRITELN(OUTPUT,' SUCCEED  ', GOALSTACK[GSX]);
    CURRENTFINALPOS;
    SUCCEED(FINDHEAVYLIGHT);
    END ELSE

    (* PRODUCTION 5 *)
    (* ---------- - *)
    IF BAL = BOTH THEN BEGIN

    WRITELN(OUTPUT,' ':25, 'PRODUCTION 05');
    IF PB[LEFT] > PB[RIGHT] THEN BEGIN
    PBTOPF(LEFT,NPF+1);
    PBTOPF(RIGHT,NPF+1);
    END ELSE BEGIN
    PBTOPF(RIGHT,NPF+1);
    PBTOPF(LEFT,NPF+1);
    END;
    WRITELN(OUTPUT,' SUCCEED  ', GOALSTACK[GSX]);
    CURRENTFINALPOS;
    SUCCEED(FINDHEAVYLIGHT);
    END;
END (* DOFINDHEAVYLIGHT *);

PROCEDURE DOWEIGH;
      (* ------- *)
BEGIN
    WRITELN(OUTPUT,'  GOAL = ', GOALSTACK[GSX]);

    (* PRODUCTION 6 *)
    (* ---------- - *)
    IF NPO > 1 THEN BEGIN

    WRITELN(OUTPUT,' ':25, 'PRODUCTION 06');
    POTOPB(NPO,LEFT);
    POTOPB(NPO,RIGHT);
    WRITELN(OUTPUT,'  SUCCEED  ', GOALSTACK[GSX]);
    SUCCEED(WEIGH);
    END ELSE
```

MASSEY UNIVERSITY

```
    (* PRODUCTION 7 *)
    (* ------------ - *)
    IF NPO = 1 THEN BEGIN

    WRITELN(OUTPUT, ' ':25, 'PRODUCTION 07');
    POTOPB(NPO,LEET);
    WRITELN(OUTPUT, '  SUCCEED  ', GOALSTACK[GSX]);
    SUCCEED(WEIGH);
    END;
END (* DOWEIGH *);
```

```
013:0020:0
013:0020:0
013:0020:0
013:0020:0
013:0020:0
013:0020:0
013:0020:0
013:0020:0
013:0020:0
013:0021:1
013:0021:1
013:0029:1
013:002A:3
013:0033:4
013:0034:4
013:0034:4
013:003E:4
013:003E:4
013:003E
```

MASSEY UNIVERSITY

```
                                                                                013:003E:4
                                                                                013:003E:4
                                                                                013:003E:4
                                                                                013:003E:4
                                                                                013:003E:4
                                                                                013:003E:4
                                                                                013:003E:4
                                                                                013:003E:4
     (*                                 *)                                      013:003E:4
     (*                                 *)                                      013:003E:4
     (*           MAIN PROGRAM          *)                                      013:003E:4
     (*                                 *)                                      013:003E:4
     (*                                 *)                                      013:003E:4
                                                                                013:003E:4
BEGIN                                                                           013:003E:4
   INITIALISE;                                                                  003:0000:1
   INPUTOFDATA;                                                                 003:0000:5
                                                                                003:0001:3
   (* PROCESS *)                                                               003:0001:3
   REPEAT                                                                       003:0001:3
      CASE GOALSTACK[GSX] OF                                                    003:0001:3
                                                                                003:0002:4
      SERIATE:  DOSERIATE;                                                      003:0002:4
                                                                                003:0003:5
      FINDHEAVYLIGHT:  DOFINDHEAVYLIGHT;                                        003:0003:5
                                                                                003:0005:0
      WEIGH:  DOWEIGH;                                                          003:0005:0
                                                                                003:0006:1
      NONE:                                                                     003:0006:1
                                                                                003:0006:1
      END;                                                                      003:0006:1
   UNTIL GOALSTACK[GSX] = NONE;                                                 003:000D:0
                                                                                003:000E:4
   FINALPOS;  (* PRINTING OF FINAL POSITIONS *)                                003:000E:4
                                                                                003:000F:2
END.                                                                            003:000F:2
================================COMPILATION SUMMARY=============================
 FILE TITLE: CARD.
ILE TITLE: WITH.
) BY COMPILER

DE SEGMENTS   VALUE ARRAYS     EST. STACK    EST. MEMORY   DISK SECTORS
        13              1
       595             71             52          1399             53
================================================================================
```

```
STAGE 1 WEIGHT SERIATION
------------------------

ORIGINAL POSITION:
     180.000000     179.000000     182.700000     181.800000     177.200000     180.800000     178.100000

GOAL =  SERIATE
                          PRODUCTION 01
 GOAL =  FINDHEAVYLIGHT
                          PRODUCTION 03
  GOAL =  WEIGH
                          PRODUCTION 06
                                                 MOVE PO[ 7] TO PB[ LEFT]  (      178.100000)
                                                 MOVE PO[ 6] TO PB[RIGHT]  (      180.800000)
  SUCCEED   WEIGH
 GOAL =  FINDHEAVYLIGHT
                          PRODUCTION 05
                                                 MOVE PB[RIGHT] TO PF[ 1]  (      180.800000)
                                                 MOVE PB[ LEFT] TO PF[ 2]  (      178.100000)
 SUCCEED   FINDHEAVYLIGHT
CURRENT FINAL POSITION:
     180.800000     178.100000  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00

GOAL =  SERIATE
                          PRODUCTION 01
 GOAL =  FINDHEAVYLIGHT
                          PRODUCTION 03
  GOAL =  WEIGH
                          PRODUCTION 06
                                                 MOVE PO[ 5] TO PB[ LEFT]  (      177.200000)
                                                 MOVE PO[ 4] TO PB[RIGHT]  (      181.800000)
  SUCCEED   WEIGH
 GOAL =  FINDHEAVYLIGHT
                          PRODUCTION 05
                                                 MOVE PB[RIGHT] TO PF[ 3]  (      181.800000)
                                                 MOVE PB[ LEFT] TO PF[ 4]  (      177.200000)
 SUCCEED   FINDHEAVYLIGHT
CURRENT FINAL POSITION:
     180.800000     178.100000     181.800000     177.200000  0.0000000E+00  0.0000000E+00  0.0000000E+00

GOAL =  SERIATE
                          PRODUCTION 01
 GOAL =  FINDHEAVYLIGHT
                          PRODUCTION 03
  GOAL =  WEIGH
                          PRODUCTION 06
                                                 MOVE PO[ 3] TO PB[ LEFT]  (      182.700000)
                                                 MOVE PO[ 2] TO PB[RIGHT]  (      179.000000)
  SUCCEED   WEIGH
 GOAL =  FINDHEAVYLIGHT
                          PRODUCTION 05
                                                 MOVE PB[ LEFT] TO PF[ 5]  (      182.700000)
                                                 MOVE PB[RIGHT] TO PF[ 6]  (      179.000000)
 SUCCEED   FINDHEAVYLIGHT
CURRENT FINAL POSITION:
```

MASSEY UNIVERSITY

```
 180.800000    178.100000    181.800000    177.200000    182.700000    179.000000  0.0000000E+00

GOAL = SERIATE
                        PRODUCTION 08
SUCCEED   SERIATE                    MOVE PO[ 1] TO PF[ 7]  (     180.000000)

FINAL POSITION:
 180.800000    178.100000    181.800000    177.200000    182.700000    179.000000    180.000000
```

APPENDIX III

STAGE 1 (ALTERNATIVE) WEIGHT SERIATION PROGRAM

MASSEY UNIVERSITY

CARD FILE :- CARD.

```
PROGRAM STAGE1;

(*                                                    *)
(*                                                    *)
(*     STAGE 1 ALTERNATIVE WEIGHT SERIATION           *)
(*     ----- - ----------- ------ --------           *)
(*                                                    *)
(*                                                    *)


(* SIMULATION OF STAGE 1 WEIGHT SERIATION BEHAVIOR *)
(* INPUT IS A LIST OF WEIGHTS FOR ITEMS TO BE SERIATED *)
(* LIST MAY BE ANY LENGTH, UP TO 20, SEPERATED BY COMMAS *)

CONST
   MAX = 20;       (* MAX. NO. OF ITEMS WHICH CAN BE HANDLED *)
   GSXMAX = 10;       (* MAX. DEPTH OF GOAL STACK *)

TYPE
   GOALSET = (SERIATE, FINDHEAVYLIGHT,WEIGH, NONE);
   BALSTATE = (EMPTY, LEFT, RIGHT, BOTH);
   BALPOS = LEFT .. RIGHT;

VAR
   PO,PF : ARRAY [1 .. MAX] OF REAL;  (* ORIGINAL & FINAL POSITIONS *)
   NPO,          (* NO. OF ITEMS IN ORIGINAL POSITION *)
   NPF,          (* NO. OF ITEMS IN FINAL POSITION *)
   NORIG,        (* NO. OF ITEMS ORIGINALLY PRESENTED *)
   LIT,
   HEV,
   K : 0 .. MAX;
   BAL : BALSTATE;
   PB : ARRAY [BALPOS] OF REAL;  (* BALANCE PANS *)
   GOALSTACK : ARRAY [0 .. GSXMAX ] OF GOALSET;
   GSX : 0 .. GSXMAX;
```

MASSEY UNIVERSITY

```
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
004:0000:1
004:0001:3
004:0002:4
004:0004:0
004:0008:4
004:0009:2
004:0009:2
004:0009:2
003:0000:1
003:0000:1
003:0000:1
006:0000:1
006:0001:2
006:0003:0
006:0003:2
006:0008:5
006:000D:3
006:000E:1
006:000E:1
006:000E:1
003:0000:1
003:0000:1
007:0000:1
007:0002:4
007:0003:2
007:0004:0
007:0004:4
007:0004:4
007:0004:4
003:0000:1
003:0000:1
008:0000:1
008:0000:5
008:0000:5
008:0002:0
008:0008:3
008:000B:0
008:000C:1
008:000C:5
008:000D:4
008:000E:3
008:000F:5
008:0019:0
```

```
      (*                                    *)
      (*                                    *)
      (*        BASIC PROCEDURES            *)
      (*                                    *)
      (*                                    *)

PROCEDURE GOAL(NEWGOAL: GOALSET);
        (* ---- *)
   (* PUSH A NEW GOAL ONTO THE STACK *)
BEGIN
   IF GSX < GSXMAX THEN BEGIN
      GSX := SUCC(GSX);
      GOALSTACK[GSX] := NEWGOAL;
   END ELSE HALT;
END (* GOAL *);

PROCEDURE SUCCEED(OLDGOAL: GOALSET);
        (* ------- *)
   (* POP A GOAL OFF THE GOAL STACK *)
BEGIN
   IF GSX > 0 THEN BEGIN
      IF GOALSTACK[GSX] = OLDGOAL THEN
      GSX := GSX - 1
      ELSE HALT;
   END ELSE HALT;
END (* SUCCEED *);

PROCEDURE INITIALISE;
        (* ---------- *)
BEGIN
   GSX := 0;  GOAL(NONE);  GOAL(SERIATE);
   BAL := EMPTY;
   NPF := 0;
END (* INITIALISE *);

PROCEDURE INPUTOFDATA;
        (* ---------- *)
BEGIN
   NPO := 0;
   REPEAT
      NPO := NPO + 1;
      READ(INPUT, PO[NPO]);
   UNTIL EOF(INPUT);
   NPO := NPO - 1;
   HEV := 1;
   LIT := NPO;
   NORIG := NPO;
   IF NPO > MAX THEN BEGIN
      WRITELN(OUTPUT, 'TOO MANY ITEMS', NPO, MAX);
      NPO := MAX;
```

MASSEY UNIVERSITY

```
008:0019:5
008:0019:5
008:0019:5
008:0019:5
008:0019:5
008:0019:5
008:0019:5
008:001F:5
008:0025:5
008:0029:3
008:002D:1
008:0033:1
008:003F:0
008:0042:4
008:0046:2
008:004A:0
008:0060:4
008:0060:4
008:0060:4
003:0000:1
003:0000:1
00C:0000:1
00C:0006:1
00C:0012:0
00C:0015:4
00C:0019:2
00C:001D:0
00C:0020:4
00C:002C:4
00C:002C:4
00C:002C:4
003:0000:1
003:0000:1
00D:0000:1
00D:0003:5
00D:0007:3
00D:000D:3
00D:0019:2
00D:001D:0
00D:0027:4
00D:0027:4
00D:0027:4
```

MASSEY UNIVERSITY

```
00D:0027:4
00D:0027:4
00D:0027:4
```

MASSEY UNIVERSITY

```
00F:0020:4
00F:0020:4
00F:0020:4
00F:0020:4
00F:0020:4
00F:0020:4
00F:0020:4
00F:0020:4
00F:0020:4
00F:0020:4
00F:0020:4
00F:0020:4
00F:0020:4
00F:0020:4
00F:0020:4
003:0000:1
003:0000:1
010:0000:1
010:0009:1
010:0009:1
010:0009:1
010:0009:1
010:000A:2
010:000A:2
010:0012:1
010:0013:0
010:0013:3
010:0013:3
010:0013:3
010:0013:3
010:0014:4
010:0014:4
010:001C:3
010:0025:3
010:0026:2
010:0026:2
010:002E:4
010:002E:4
010:002E:4
003:0000:1
003:0000:1
011:0000:1
011:0009:2
011:0009:2
011:0009:2
011:0009:2
011:000A:3
011:000A:3
011:0012:3
011:0013:3
011:0014:0
011:0014:0
011:0014:0
011:0014:0
011:0014:0
011:0014:0
011:0014:0
011:0014:0
011:0014:0
011:0014:0
```

```
      (*                                  *)
      (*                                  *)
      (*          PRODUCTION SYSTEM       *)
      (*                                  *)
      (*                                  *)

PROCEDURE DOSERIATE;
      (* --------- *)
BEGIN
   WRITELN(OUTPUT, 'GOAL = ', GOALSTACK[GSX]);

   (* PRODUCTION 1 *)
   (* ---------- - *)
   IF NPO > 0 THEN BEGIN

   WRITELN(OUTPUT, ' ':25, 'PRODUCTION 01');
   GOAL(FINDHEAVYLIGHT);
   END ELSE

   (* PRODUCTION 2 *)
   (* ---------- - *)
   IF NPO = 0 THEN BEGIN

   WRITELN(OUTPUT,' ':25, 'PRODUCTION 02');
   WRITELN(OUTPUT,'SUCCEED ', GOALSTACK[GSX]);
   SUCCEED(SERIATE);
   END;
END (* DOSERIATE *);

PROCEDURE DOFINDHEAVYLIGHT;
      (* --------------- *)
BEGIN
   WRITELN(OUTPUT, ' GOAL = ', GOALSTACK[GSX]);

   (* PRODUCTION 3 *)
   (* ---------- - *)
   IF BAL = EMPTY THEN BEGIN

   WRITELN(OUTPUT,' ':25, 'PRODUCTION 03');
   GOAL(WEIGH);
   END ELSE
```

MASSEY UNIVERSITY

```
011:0014:0
011:0014:0
011:0014:0
011:0014:0
011:0014:0
011:0014:0
011:0014:0
011:0014:0
011:0014:0
011:0015:4
011:0015:4
011:001D:4
011:001E:5
011:0021:0
011:0021:3
011:0023:5
011:0023:5
011:002D:0
011:002D:4
011:002E:3
011:002F:0
011:002F:0
011:002F:0
011:002F:0
011:0030:2
011:0030:2
011:0038:2
011:003A:0
011:003B:2
011:003C:3
011:003E:0
011:003F:1
011:003F:4
011:0041:1
011:0042:2
011:0043:4
011:0044:5
011:0044:5
011:004E:0
011:004E:4
011:004F:3
011:004F:3
011:005B:4
011:005B:4
011:005B:4
003:0000:1
003:0000:1
012:0000:1
012:0009:2
012:0009:2
012:0009:2
012:0009:2
012:000A:3
012:000A:3
012:0012:3
012:0013:5
012:0015:2
012:001E:3
012:001F:3
```

```
    (* PRODUCTION 4 *)
    (* ---------- - *)
    IF BAL IN [LEFT,RIGHT] THEN BEGIN

    WRITELN(OUTPUT,' ':25, 'PRODUCTION 04');
       IF BAL = LEFT THEN BEGIN
       PBTOPF(LEFT,(NORIG+1) DIV 2);
       END ELSE BEGIN
       PBTOPF(RIGHT,(NORIG+1) DIV 2);
       END;
    WRITELN(OUTPUT, ' SUCCEED  ', GOALSTACK[GSX]);
    CURRENTFINALPOS;
    SUCCEED(FINDHEAVYLIGHT);
    END ELSE

    (* PRODUCTION 5 *)
    (* ---------- - *)
    IF BAL = BOTH THEN BEGIN

    WRITELN(OUTPUT,' ':25, 'PRODUCTION 05');
    IF PB[LEFT] > PB[RIGHT] THEN BEGIN
       PBTOPF(LEFT,HEV);
       HEV := HEV + 1;
       PBTOPF(RIGHT,LIT);
       LIT := LIT - 1;
    END ELSE BEGIN
       PBTOPF(RIGHT,HEV);
       HEV := HEV + 1;
       PBTOPF(LEFT,LIT);
       LIT := LIT - 1;
    END;
    WRITELN(OUTPUT, ' SUCCEED  ', GOALSTACK[GSX]);
    CURRENTFINALPOS;
    SUCCEED(FINDHEAVYLIGHT);
    END;
END (* DOFINDHEAVYLIGHT *);

PROCEDURE DOWEIGH;
      (* ------- *)
BEGIN
    WRITELN(OUTPUT, '  GOAL = ', GOALSTACK[GSX]);

    (* PRODUCTION 6 *)
    (* ---------- - *)
    IF NPO > 1 THEN BEGIN

    WRITELN(OUTPUT, ' ':25, 'PRODUCTION 06');
    POTOPB(NPO,LEFT);
    POTOPB(NPO,RIGHT);
    WRITELN(OUTPUT, '  SUCCEED  ', GOALSTACK[GSX]);
    SUCCEED(WEIGH);
    END ELSE
```

MASSEY UNIVERSITY

```
012:0020:0
012:0020:0
012:0020:0
012:0020:0
012:0020:0
012:0020:0
012:0020:0
012:0020:0
012:0020:0
012:0021:1
012:0021:1
012:0029:1
012:002A:3
012:0033:4
012:0034:4
012:0034:4
012:003E:4
012:003E:4
012:003E
```

MASSEY UNIVERSITY

```
012:003E:4
012:003E:4
012:003E:4
012:003E:4
012:003E:4
012:003E:4
012:003E:4
012:003E:4
012:003E:4
012:003E:4
012:003E:4
012:003E:4
012:003E:4
012:003E:4
003:0000:1
003:0000:5
003:0001:3
003:0001:3
003:0001:3
003:0001:3
003:0002:4
003:0002:4
003:0003:5
003:0003:5
003:0005:0
003:0005:0
003:0006:1
003:0006:1
003:0006:1
003:0006:1
003:000D:0
003:000E:4
003:000E:4
003:000F:2
003:000F:2
```

```
        (*                                  *)
        (*                                  *)
        (*            MAIN PROGRAM          *)
        (*                                  *)
        (*                                  *)

    BEGIN
       INITIALISE;
       INPUTOFDATA;

       (* PROCESS *)
       REPEAT
          CASE GOALSTACK[GSX] OF

          SERIATE:  DOSERIATE;

          FINDHEAVYLIGHT:  DOFINDHEAVYLIGHT;

          WEIGH:  DOWEIGH;

          NONE:

          END;
       UNTIL GOALSTACK[GSX] = NONE;

       FINALPOS;  (* PRINTING OF FINAL POSITIONS *)

    END.
```

```
==========================================COMPILATION SUMMARY==========================================
RAM SOURCE FILE TITLE: CARD.
RAM CODE FILE TITLE: WITH.
RRORS FOUND BY COMPILER

        CODE SEGMENTS   VALUE ARRAYS     EST. STACK    EST. MEMORY   DISK SECTORS
F:              12              1
:              545             64             53          1369             51
=====================================================================================================
```

STAGE 1 WEIGHT SERIATION

ORIGINAL POSITION:
180.000000 179.000000 182.700000 181.800000 177.200000 180.800000 178.100000

GOAL = SERIATE
PRODUCTION 01
GOAL = FINDHEAVYLIGHT
PRODUCTION 03
GOAL = WEIGH
PRODUCTION 06
MOVE PO[7] TO PB[LEFT] (178.100000)
MOVE PO[6] TO PB[RIGHT] (180.800000)
SUCCEED WEIGH
GOAL = FINDHEAVYLIGHT
PRODUCTION 05
MOVE PB[RIGHT] TO PF[1] (180.800000)
MOVE PB[LEFT] TO PF[7] (178.100000)
SUCCEED FINDHEAVYLIGHT
CURRENT FINAL POSITION:
180.800000 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 178.100000

GOAL = SERIATE
PRODUCTION 01
GOAL = FINDHEAVYLIGHT
PRODUCTION 03
GOAL = WEIGH
PRODUCTION 06
MOVE PO[5] TO PB[LEFT] (177.200000)
MOVE PO[4] TO PB[RIGHT] (181.800000)
SUCCEED WEIGH
GOAL = FINDHEAVYLIGHT
PRODUCTION 05
MOVE PB[RIGHT] TO PF[2] (181.800000)
MOVE PB[LEFT] TO PF[6] (177.200000)
SUCCEED FINDHEAVYLIGHT
CURRENT FINAL POSITION:
180.800000 181.800000 0.0000000E+00 0.0000000E+00 0.0000000E+00 177.200000 178.100000

GOAL = SERIATE
PRODUCTION 01
GOAL = FINDHEAVYLIGHT
PRODUCTION 03
GOAL = WEIGH
PRODUCTION 06
MOVE PO[3] TO PB[LEFT] (182.700000)
MOVE PO[2] TO PB[RIGHT] (179.000000)
SUCCEED WEIGH
GOAL = FINDHEAVYLIGHT
PRODUCTION 05
MOVE PB[LEFT] TO PF[3] (182.700000)
MOVE PB[RIGHT] TO PF[5] (179.000000)
SUCCEED FINDHEAVYLIGHT
CURRENT FINAL POSITION:

MASSEY UNIVERSITY

```
     180.800000     181.800000     182.700000  0.0000000E+00     179.000000     177.200000     178.100000

GOAL =  SERIATE
 GOAL =  FINDHEAVYLIGHT                PRODUCTION 01
  GOAL =  WEIGH                        PRODUCTION 03
                                       PRODUCTION 07
                                                      MOVE PO[ 1] TO PB[ LEFT]  (     180.000000)
  SUCCEED   WEIGH
 GOAL =  FINDHEAVYLIGHT
                                       PRODUCTION 04
                                                      MOVE PB[ LEFT] TO PF[ 4]  (     180.000000)
 SUCCEED   FINDHEAVYLIGHT
CURRENT FINAL POSITION:
     180.800000     181.800000     182.700000     180.000000     179.000000     177.200000     178.100000

GOAL =  SERIATE
                                       PRODUCTION 02
SUCCEED   SERIATE

FINAL POSITION:
     180.800000     181.800000     182.700000     180.000000     179.000000     177.200000     178.100000
```

APPENDIX IV

STAGE 3 WEIGHT SERIATION PROGRAM

CARD FILE := CARD.

```
PROGRAM STAGE3;

(*                                                    *)
(*                                                    *)
(*          STAGE 3 WEIGHT-SERIATION                  *)
(*          ------- - ------ --------                 *)
(*                                                    *)
(*                                                    *)

(* SIMULATION OF STAGE 3 WEIGHT SERIATION BEHAVIOR *)
(* INPUT IS A LIST OF WEIGHTS FOR ITEMS TO BE SERIATED *)
(* LIST MAY BE ANY LENGTH, UP TO 20, SEPERATED BY COMMAS *)

CONST
   MAX = 20;       (* MAX. NO. OF ITEMS WHICH CAN BE HANDLED *)
   GSXMAX = 10;    (* MAX. DEPTH OF GOAL STACK *)

TYPE
   GOALSET = (SERIATE, LCATEHEAVIEST, WEIGH, GETRIDOF, NONE);
   BALSTATE = (EMPTY, LEFT, RIGHT, BOTH);
   BALPOS = LEFT .. RIGHT;

VAR
   PO,PF,PI : ARRAY [1 .. MAX] OF REAL;
         (* ORIGINAL, INTERMEDIATE & FINAL POSITIONS *)
   NPO,          (* NO. OF ITEMS IN ORIGINAL POSITION *)
   NPF,          (* NO. OF ITEMS IN FINAL POSITION *)
   NPI,          (* NO. OF ITEMS IN INTERMEDIATE POSITION *)
   NORIG,        (* NO. OF ITEMS ORIGINALLY PRESENTED *)
   K : 0 .. MAX;
   BAL : BALSTATE;
   PB : ARRAY [BALPOS] OF REAL;  (* BALANCE PANS *)
   GOALSTACK : ARRAY [0 .. GSXMAX ] OF GOALSET;
   GSX : 0 .. GSXMAX;
```

003:0000:1

```
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
004:0000:1
004:0001:3
004:0002:4
004:0004:0
004:0008:4
004:0009:2
004:0009:2
004:0009:2
003:0000:1
003:0000:1
003:0000:1
006:0000:1
006:0001:2
006:0003:0
006:0003:2
006:0008:5
006:000D:3
006:000E:1
006:000E:1
006:000E:1
003:0000:1
003:0000:1
007:0000:1
007:0002:4
007:0003:2
007:0004:0
007:0004:4
007:0005:2
007:0005:2
007:0005:2
003:0000:1
003:0000:1
008:0000:1
008:0000:5
008:0000:5
008:0002:0
008:0008:3
008:000B:0
008:000C:1
008:000D:0
008:000D:0
008:000D:0
```

```
        (*                                      *)
        (*                                      *)
        (*          BASIC PROCEDURES            *)
        (*                                      *)
        (*                                      *)

PROCEDURE GOAL(NEWGOAL: GOALSET);
        (* ---- *)
    (* PUSH A NEW GOAL ONTO THE STACK *)
BEGIN
    IF GSX < GSXMAX THEN BEGIN
        GSX := SUCC(GSX);
        GOALSTACK[GSX] := NEWGOAL;
    END ELSE HALT;
END (* GOAL *);

PROCEDURE SUCCEED(OLDGOAL: GOALSET);
        (* ------- *)
    (* PUSH A NEW GOAL ONTO THE STACK *)
BEGIN
    IF GSX > 0 THEN BEGIN
        IF GOALSTACK[GSX] = OLDGOAL THEN
        GSX := GSX - 1
        ELSE HALT;
    END ELSE HALT;
END (* SUCCEED *);

PROCEDURE INITIALISE;
        (* ---------- *)
BEGIN
    GSX := 0;  GOAL(NONE);  GOAL(SERIATE);
    BAL := EMPTY;
    NPF := 0;
    NPI := 0;
END (* INITIALISE *);

PROCEDURE INPUTOFDATA;
        (* ---------- *)
BEGIN
    NP0 := 0;
    REPEAT
        NP0 := NP0 + 1;
        READ(INPUT, P0[NP0]);
    UNTIL EOF(INPUT);
    NP0 := NP0 - 1;
    NORIG := NP0;
```

```
008:000D:0
008:000D:0
008:000D:0
008:000D:0
008:000D:0
008:000D:0
008:000D:0
008:000D:0
008:000E:2
008:0017:3
008:0018:2
008:0018:2
008:001E:2
008:0024:2
008:0028:0
008:002B:4
008:0031:4
008:003D:3
008:0041:1
008:0044:5
008:0048:3
008:005E:4
008:005E:4
008:005E:4
003:0000:1
003:0000:1
00C:0000:1
00C:0006:1
00C:0012:0
00C:0015:4
00C:0019:2
00C:001D:0
00C:0020:4
00C:002C:4
00C:002C:4
00C:002C:4
003:0000:1
003:0000:1
00D:0000:1
00D:0003:5
00D:0007:3
00D:000D:3
00D:0019:2
00D:001D:0
00D:0020:4
00D:0024:2
00D:0032:4
00D:0032:4
00D:0032:4
```

```
00D:0032:4
00D:0032:4
00D:0032:4
00D:0032:4
00D:0032:4
00D:0032:4
00D:0032:4
00D:0032:4
00D:0032:4
00D:0032:4
00D:0032:4
00D:0032:4
00D:0032:4
00D:0032:4
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
00E:0000:1
00E:0001:2
00E:0003:4
00E:0011:2
00E:0019:0
00E:0019:0
00E:001A:2
00E:001B:3
00E:001B:3
00E:001D:3
00E:001E:4
00E:0020:4
00E:0020:4
00E:0020:4
003:0000:1
003:0000:1
003:0000:1
00F:0000:1
00F:0001:2
00F:0003:4
00F:0011:2
00F:0019:0
00F:0019:0
00F:001A:2
00F:001B:3
00F:001B:3
00F:001D:3
00F:001E:4
00F:0020:4
00F:0020:4
00F:0020:4
00F:0020:4
00F:0020:4
00F:0020:4
00F:0020:4
00F:0020:4
00F:0020:4
00F:0020:4
00F:0020:4
```

```
      (*                                  *)
      (*                                  *)
      (*          PROCEDURES FOR          *)
      (*       MOVEMENT OF WEIGHTS        *)
      (*                                  *)
      (*                                  *)

PROCEDURE PBTOPI(BPOS: BALPOS);
      (* ------ *)
(* MOVEMENT OF WEIGHTS FROM THE BALANCE PANS
                    TO THE INTERMEDIATE POSITION *)

BEGIN
   NPI := NPI + 1;
   PI[NPI] := PB[BPOS];
   WRITELN(OUTPUT, ' ':45, 'MOVE PB[', BPOS:5,'] TO PI[', NPI:2,
           '] (',PB[BPOS],')');

   IF BAL = BOTH THEN BEGIN
      IF BPOS = LEFT THEN
         BAL := RIGHT
      ELSE BAL := LEFT;
   END ELSE BAL := EMPTY;
END (* PBTOPI *);

PROCEDURE PBTOPF(BPOS: BALPOS);
      (* ------ *)
(* MOVEMENT OF WEIGHTS FROM THE BALANCE PANS TO THE FINAL POSITION *)
BEGIN
   NPF := NPF + 1;
   PF[NPF] := PB[BPOS];
   WRITELN(OUTPUT, ' ':45, 'MOVE PB[', BPOS:5,'] TO PF[', NPF:2,
           '] (',PB[BPOS],')');

   IF BAL = BOTH THEN BEGIN
      IF BPOS = LEFT THEN
         BAL := RIGHT
      ELSE BAL := LEFT;
   END ELSE BAL := EMPTY;
END (* PBTOPF *);
```

MASSEY UNIVERSITY

```
00F:0020:4
00F:0020:4
00F:0020:4
00F:0020:4
00F:0020:4
00F:0020:4
003:0000:1
003:0000:1
003:0000:1
003:0000:1
010:0000:1
010:0008:0
010:000A:3
010:000B:2
010:000C:0
010:000F:4
010:0013:2
010:0019:2
010:0025:1
010:0028:5
010:002C:3
010:0030:1
010:0040:4
010:0040:4
010:0040:4
003:0000:1
003:0000:1
003:0000:1
011:0000:1
011:0000:5
011:0015:3
011:0017:5
011:0019:1
011:0019:1
011:001A:2
011:001A:2
011:001C:3
011:001E:4
011:001E:4
011:001E:4
```

MACSLY UNIVERSITY

```
011:001E:4
011:001E:4
011:001E:4
011:001E:4
011:001E:4
011:001E:4
011:001E:4
011:001E:4
011:001E:4
011:001E:4
011:001E:4
011:001E:4
011:001E:4
011:001E:4
003:0000:1
003:0000:1
012:0000:1
012:0009:1
012:0009:1
012:0009:1
012:0009:1
012:000A:2
012:000A:2
012:0012:2
012:0013:1
012:0013:4
012:0013:4
012:0013:4
012:0013:4
012:0016:1
012:0016:1
012:001E:1
012:001F:0
012:001F:3
012:001F:3
012:001F:3
012:001F:3
012:0021:3
012:0021:3
012:0029:3
012:002A:1
012:002A:4
012:002A:4
012:002A:4
012:002A:4
012:002E:0
012:0036:0
012:0037:1
012:0038:0
012:0038:3
012:0039:3
012:0039:3
012:0042:4
012:0043:3
012:0043:3
012:004F:4
012:004F:4
012:004F:4
012:004F:4
```

```
        (*                                  *)
        (*                                  *)
        (*        PRODUCTION SYSTEM         *)
        (*                                  *)
        (*                                  *)

PROCEDURE DOSERIATE;
        (* ---------- *)
BEGIN
   WRITELN(OUTPUT, 'GOAL = ', GOALSTACK[GSX]);

   (* PRODUCTION 1 *)
   (* ---------- - *)
   IF NPO > 1 THEN BEGIN

   WRITELN(OUTPUT, ' ':25, 'PRODUCTION 01');
   GOAL(LOCATEHEAVIEST);
   END ELSE

   (* PRODUCTION 2 *)
   (* ---------- - *)
   IF (NPO = 1) AND (BAL IN [LEFT,RIGHT]) THEN BEGIN

   WRITELN(OUTPUT, ' ':25, 'PRODUCTION 02');
   GOAL(LOCATEHEAVIEST);
   END ELSE

   (* PRODUCTION 3 *)
   (* ---------- - *)
   IF (NPO = 0) AND (NPI > 0) THEN BEGIN

   WRITELN(OUTPUT, ' ':25, 'PRODUCTION 03');
      PITOPO;
   END ELSE

   (* PRODUCTION 4 *)
   (* ---------- - *)
   IF (NPO = 0) AND (NPI = 0) AND (BAL IN [LEFT,RIGHT]) THEN BEGIN

   WRITELN(OUTPUT, ' ':25, 'PRODUCTION 04');
      IF BAL = LEFT THEN BEGIN
         PBTOPF(LEFT);
      END ELSE BEGIN
         PBTOPF(RIGHT);
      END;
   WRITELN(OUTPUT, 'SUCCEED  ', GOALSTACK[GSX]);
   SUCCEED(SERIATE);
   END;
END (* DOSERIATE *);
```

```
012:004F:4
012:004F:4
012:004F:4
012:004F:4
012:004F:4
012:004F:4
012:004F:4
003:0000:1
003:0000:1
013:0000:1
013:0009:2
013:0009:2
013:0009:2
013:0009:2
013:000B:0
013:000B:0
013:0013:0
013:0014:0
013:0014:3
013:0014:3
013:0014:3
013:0014:3
013:0016:4
013:0016:4
013:001E:4
013:001F:4
013:0020:1
013:0020:1
013:0020:1
013:0020:1
013:0022:2
013:0022:2
013:002A:2
013:002C:0
013:002C:5
013:002D:2
013:002E:2
013:002E:2
013:0037:3
013:0038:1
013:0039:0
013:0039:0
013:0043:4
013:0043:4
013:0043:4
003:0000:1
003:0000:1
014:0000:1
014:0009:2
014:0009:2
014:0009:2
014:0009:2
014:000A:3
014:000A:3
014:0012:3
014:0013:5
014:0015:2
014:0015:5
014:0015:5
014:0015:5
014:0015:5
```

```
PROCEDURE DOLCATEHEAVIEST;
      (* ---------------- *)
BEGIN
   WRITELN(OUTPUT, ' GOAL = ', GOALSTACK[GSX]);

   (* PRODUCTION 5 *)
   (* ---------- - *)
   IF BAL IN [LEFT,RIGHT,EMPTY] THEN BEGIN

   WRITELN(OUTPUT, ' ':25, 'PRODUCTION 05');
      GOAL(WEIGH);
   END ELSE

   (* PRODUCTION 6 *)
   (* ---------- - *)
   IF (BAL = BOTH) AND (NPO > 0) THEN BEGIN

   WRITELN(OUTPUT, ' ':25, 'PRODUCTION 06');
      GOAL(GETRIDOF);
   END ELSE

   (* PRODUCTION 7 *)
   (* ---------- - *)
   IF (BAL = BOTH) AND (NPO = 0) THEN BEGIN

   WRITELN(OUTPUT, ' ':25, 'PRODUCTION 07');
   IF PB[LEFT] > PB[RIGHT] THEN BEGIN
         PBTOPF(LEFT);
   END ELSE BEGIN
         PBTOPF(RIGHT);
   END;
   WRITELN(OUTPUT, ' SUCCEED  ', GOALSTACK[GSX]);
   CURRENTFINALPOS;
   SUCCEED(LCATEHEAVIEST);
   END;
END (* DOLCATEHEAVIEST *);

PROCEDURE DOWEIGH;
      (* ------- *)
BEGIN
   WRITELN(OUTPUT, '  GOAL = ', GOALSTACK[GSX]);

   (* PRODUCTION 8 *)
   (* ---------- - *)
   IF BAL = EMPTY THEN BEGIN

   WRITELN(OUTPUT, ' ':25, 'PRODUCTION 08');
      POTOPB(NPO,LEFT);
      POTOPB(NPO,RIGHT);
   END ELSE BEGIN
```

```
014:0015:5
014:0015:5
014:0015:5
014:0015:5
014:0015:5
014:0015:5
014:0015:5
014:0015:5
014:0017:3
014:0017:3
014:001F:3
014:0020:4
014:0022:1
014:0022:4
014:0024:0
014:0024:0
014:0024:0
014:0024:0
014:002D:1
014:002E:1
014:0036:4
014:0036:4
014:0036:4
003:0000:1
003:0000:1
015:0000:1
015:0009:2
015:0009:2
015:0009:2
015:0009:2
015:0011:2
015:0013:0
015:0014:0
015:0014:3
015:0015:2
015:0015:2
015:001E:3
015:001F:3
015:0025:4
015:0025:4
015:0025:4
```

MASSEY UNIVERSITY

```
015:0025:4
015:0025:4
015:0025:4
015:0025:4
015:0025:4
015:0025:4
015:0025:4
015:0025:4
015:0025:4
015:0025:4
015:0025:4
015:0025:4
015:0025:4
015:0025:4
015:0025:4
015:0025:4
003:0000:1
003:0000:1
003:0000:5
003:0000:5
003:0001:3
003:0001:3
003:0001:3
003:0001:3
003:0002:4
003:0002:4
003:0003:5
003:0003:5
003:0005:0
003:0005:0
003:0006:1
003:0006:1
003:0007:2
003:0007:2
003:0007:2
003:0007:2
003:0010:0
003:0011:4
003:0011:4
003:0012:2
003:0012:2
```

```
        (*                                  *)
        (*                                  *)
        (*          MAIN PROGRAM            *)
        (*                                  *)
        (*                                  *)

    BEGIN
      INITIALISE;
      INPUTOFDATA;
      (* PROCESS *)
      REPEAT
        CASE GOALSTACK[GSX] OF
        SERIATE:  DOSERIATE;
        LCATEHEAVIEST:  DOLCATEHEAVIEST;
        WEIGH:  DOWEIGH;
        GETRIDOF:  DOGETRIDOF;
        NONE:
        END;
      UNTIL GOALSTACK[GSX] = NONE;
      FINALPOS;  (* PRINTING OF FINAL POSITIONS *)
    END.
```

```
=================================COMPILATION SUMMARY=================================
 SOURCE FILE TITLE: CARD.
 CODE FILE TITLE: WITH.
RS FOUND BY COMPILER

   CODE SEGMENTS   VALUE ARRAYS     EST. STACK    EST. MEMORY   DISK SECTORS
          15              1
         702             82             56          1470            62
=====================================================================================
```

STAGE 3 WEIGHT SERIATION

ORIGINAL POSITION:
179.000000 181.800000 178.100000 180.000000 180.800000 177.200000 182.700000

GOAL = SERIATE
PRODUCTION 01
GOAL = LCATEHEAVIEST
PRODUCTION 05
GOAL = WEIGH
PRODUCTION 08
MOVE PO[7] TO PB[LEFT] (182.700000)
MOVE PO[6] TO PB[RIGHT] (177.200000)
SUCCEED WEIGH
GOAL = LCATEHEAVIEST
PRODUCTION 06
GOAL = GETRIDOF
PRODUCTION 10
MOVE PB[RIGHT] TO PI[1] (177.200000)
SUCCEED GETRIDOF
GOAL = LCATEHEAVIEST
PRODUCTION 05
GOAL = WEIGH
PRODUCTION 09
MOVE PO[5] TO PB[RIGHT] (180.800000)
SUCCEED WEIGH
GOAL = LCATEHEAVIEST
PRODUCTION 06
GOAL = GETRIDOF
PRODUCTION 10
MOVE PB[RIGHT] TO PI[2] (180.800000)
SUCCEED GETRIDOF
GOAL = LCATEHEAVIEST
PRODUCTION 05
GOAL = WEIGH
PRODUCTION 09
MOVE PO[4] TO PB[RIGHT] (180.000000)
SUCCEED WEIGH
GOAL = LCATEHEAVIEST
PRODUCTION 06
GOAL = GETRIDOF
PRODUCTION 10
MOVE PB[RIGHT] TO PI[3] (180.000000)
SUCCEED GETRIDOF
GOAL = LCATEHEAVIEST
PRODUCTION 05
GOAL = WEIGH
PRODUCTION 09
MOVE PO[3] TO PB[RIGHT] (178.100000)
SUCCEED WEIGH
GOAL = LCATEHEAVIEST
PRODUCTION 06
GOAL = GETRIDOF
PRODUCTION 10
MOVE PB[RIGHT] TO PI[4] (178.100000)
SUCCEED GETRIDOF
GOAL = LCATEHEAVIEST
PRODUCTION 05

```
GOAL = WEIGH
                                PRODUCTION 09
                                                    MOVE PO[ 2] TO PB[RIGHT]  (     181.800000)
  SUCCEED   WEIGH
GOAL = LCATEHEAVIEST
                                PRODUCTION 06
  GOAL = GETRIDOF
                                PRODUCTION 10
                                                    MOVE PB[RIGHT] TO PI[ 5]  (     181.800000)
  SUCCEED   GETRIDOF
GOAL = LCATEHEAVIEST
                                PRODUCTION 05
  GOAL = WEIGH
                                PRODUCTION 09
                                                    MOVE PO[ 1] TO PB[RIGHT]  (     179.000000)
  SUCCEED   WEIGH
GOAL = LCATEHEAVIEST
                                PRODUCTION 07
                                                    MOVE PB[ LEFT] TO PF[ 1]  (     182.700000)
  SUCCEED   LCATEHEAVIEST
CURRENT FINAL POSITION:
     182.700000  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00

GOAL = SERIATE
                                PRODUCTION 03
                                                    TRANSFORM PI TO PO

NEW ORIGINAL POSITION:
     177.200000     180.800000     180.000000     178.100000     181.800000

GOAL = SERIATE
                                PRODUCTION 01
  GOAL = LCATEHEAVIEST
                                PRODUCTION 05
  GOAL = WEIGH
                                PRODUCTION 09
                                                    MOVE PO[ 5] TO PB[ LEFT]  (     181.800000)
  SUCCEED   WEIGH
GOAL = LCATEHEAVIEST
                                PRODUCTION 06
  GOAL = GETRIDOF
                                PRODUCTION 10
                                                    MOVE PB[RIGHT] TO PI[ 1]  (     179.000000)
  SUCCEED   GETRIDOF
GOAL = LCATEHEAVIEST
                                PRODUCTION 05
  GOAL = WEIGH
                                PRODUCTION 09
                                                    MOVE PO[ 4] TO PB[RIGHT]  (     178.100000)
  SUCCEED   WEIGH
GOAL = LCATEHEAVIEST
                                PRODUCTION 06
  GOAL = GETRIDOF
                                PRODUCTION 10
                                                    MOVE PB[RIGHT] TO PI[ 2]  (     178.100000)
  SUCCEED   GETRIDOF
GOAL = LCATEHEAVIEST
```

```
                    PRODUCTION 05
  GOAL =  WEIGH
                    PRODUCTION 09
                                        MOVE PO[ 3] TO PB[RIGHT]  (     180.000000)
  SUCCEED    WEIGH
GOAL =  LCATEHEAVIEST
                    PRODUCTION 06
   GOAL =  GETRIDOF
                    PRODUCTION 10
                                        MOVE PB[RIGHT] TO PI[ 3]  (     180.000000)
  SUCCEED    GETRIDOF
GOAL =  LCATEHEAVIEST
                    PRODUCTION 05
   GOAL =  WEIGH
                    PRODUCTION 09
                                        MOVE PO[ 2] TO PB[RIGHT]  (     180.800000)
  SUCCEED    WEIGH
GOAL =  LCATEHEAVIEST
                    PRODUCTION 06
   GOAL =  GETRIDOF
                    PRODUCTION 10
                                        MOVE PB[RIGHT] TO PI[ 4]  (     180.800000)
  SUCCEED    GETRIDOF
GOAL =  LCATEHEAVIEST
                    PRODUCTION 05
   GOAL =  WEIGH
                    PRODUCTION 09
                                        MOVE PO[ 1] TO PB[RIGHT]  (     177.200000)
  SUCCEED    WEIGH
GOAL =  LCATEHEAVIEST
                    PRODUCTION 07
                                        MOVE PB[ LEFT] TO PF[ 2]  (     181.800000)
 SUCCEED    LCATEHEAVIEST
CURRENT FINAL POSITION:
     182.700000     181.800000  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00

GOAL =  SERIATE
                    PRODUCTION 03
                                        TRANSFORM PI TO PO

NEW ORIGINAL POSITION:
     179.000000     178.100000     180.000000     180.800000

GOAL =  SERIATE
                    PRODUCTION 01
   GOAL =  LCATEHEAVIEST
                    PRODUCTION 05
    GOAL =  WEIGH
                    PRODUCTION 09
                                        MOVE PO[ 4] TO PB[ LEFT]  (     180.800000)
  SUCCEED    WEIGH
GOAL =  LCATEHEAVIEST
                    PRODUCTION 06
   GOAL =  GETRIDOF
                    PRODUCTION 10
                                        MOVE PB[RIGHT] TO PI[ 1]  (     177.200000)
```

```
GOAL =  LCATEHEAVIEST
                        PRODUCTION 05
 GOAL =  WEIGH
                        PRODUCTION 09
                                        MOVE PO[ 3] TO PB[RIGHT]  (      180.000000)
  SUCCEED   WEIGH
GOAL =  LCATEHEAVIEST
                        PRODUCTION 06
   GOAL =  GETRIDOF
                        PRODUCTION 10
                                        MOVE PB[RIGHT] TO PI[ 2]  (      180.000000)
   SUCCEED   GETRIDOF
GOAL =  LCATEHEAVIEST
                        PRODUCTION 05
 GOAL =  WEIGH
                        PRODUCTION 09
                                        MOVE PO[ 2] TO PB[RIGHT]  (      178.100000)
  SUCCEED   WEIGH
GOAL =  LCATEHEAVIEST
                        PRODUCTION 06
   GOAL =  GETRIDOF
                        PRODUCTION 10
                                        MOVE PB[RIGHT] TO PI[ 3]  (      178.100000)
   SUCCEED   GETRIDOF
GOAL =  LCATEHEAVIEST
                        PRODUCTION 05
 GOAL =  WEIGH
                        PRODUCTION 09
                                        MOVE PO[ 1] TO PB[RIGHT]  (      179.000000)
  SUCCEED   WEIGH
GOAL =  LCATEHEAVIEST
                        PRODUCTION 07
                                        MOVE PB[ LEFT] TO PF[ 3]  (      180.800000)
 SUCCEED   LCATEHEAVIEST
CURRENT FINAL POSITION:
     182.700000     181.800000     180.800000  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00

GOAL =  SERIATE
                        PRODUCTION 03
                                        TRANSFORM PI TO PO

NEW ORIGINAL POSITION:
     177.200000     180.000000     178.100000

GOAL =  SERIATE
                        PRODUCTION 01
 GOAL =  LCATEHEAVIEST
                        PRODUCTION 05
  GOAL =  WEIGH
                        PRODUCTION 09
                                        MOVE PO[ 3] TO PB[ LEFT]  (      178.100000)
  SUCCEED   WEIGH
 GOAL =  LCATEHEAVIEST
                        PRODUCTION 06
   GOAL =  GETRIDOF
                        PRODUCTION 10
                                        MOVE PB[ LEFT] TO PI[ 1]  (      178.100000)
```

```
  SUCCEED   GETRIDOF
GOAL =  LCATEHEAVIEST
                        PRODUCTION 05
 GOAL =  WEIGH
                        PRODUCTION 09
                                        MOVE PO[ 2] TO PB[ LEFT]  (      180.000000)
  SUCCEED   WEIGH
GOAL =  LCATEHEAVIEST
                        PRODUCTION 06
  GOAL =  GETRIDOF
                        PRODUCTION 10
                                        MOVE PB[RIGHT] TO PI[ 2]  (      179.000000)
  SUCCEED   GETRIDOF
GOAL =  LCATEHEAVIEST
                        PRODUCTION 05
 GOAL =  WEIGH
                        PRODUCTION 09
                                        MOVE PO[ 1] TO PB[RIGHT]  (      177.200000)
  SUCCEED   WEIGH
GOAL =  LCATEHEAVIEST
                        PRODUCTION 07
                                        MOVE PB[ LEFT] TO PF[ 4]  (      180.000000)
 SUCCEED   LCATEHEAVIEST
CURRENT FINAL POSITION:
     182.700000      181.800000      180.800000      180.000000  0.0000000E+00  0.0000000E+00  0.0000000E+00

GOAL =  SERIATE
                        PRODUCTION 03
                                        TRANSFORM PI TO PO

NEW ORIGINAL POSITION:
     178.100000      179.000000

GOAL =  SERIATE
                        PRODUCTION 01
 GOAL =  LCATEHEAVIEST
                        PRODUCTION 05
  GOAL =  WEIGH
                        PRODUCTION 09
                                        MOVE PO[ 2] TO PB[ LEFT]  (      179.000000)
  SUCCEED   WEIGH
GOAL =  LCATEHEAVIEST
                        PRODUCTION 06
  GOAL =  GETRIDOF
                        PRODUCTION 10
                                        MOVE PB[RIGHT] TO PI[ 1]  (      177.200000)
  SUCCEED   GETRIDOF
GOAL =  LCATEHEAVIEST
                        PRODUCTION 05
 GOAL =  WEIGH
                        PRODUCTION 09
                                        MOVE PO[ 1] TO PB[RIGHT]  (      178.100000)
  SUCCEED   WEIGH
GOAL =  LCATEHEAVIEST
                        PRODUCTION 07
                                        MOVE PB[ LEFT] TO PF[ 5]  (      179.000000)
 SUCCEED   LCATEHEAVIEST
```

```
CURRENT FINAL POSITION:
     182.700000     181.800000     180.800000     180.000000     179.000000  0.0000000E+00  0.0000000E+00

GOAL =  SERIATE
                              PRODUCTION 03
                                                  TRANSFORM PI TO PO

NEW ORIGINAL POSITION:
     177.200000

GOAL =  SERIATE
                              PRODUCTION 02
 GOAL =  LCATEHEAVIEST        PRODUCTION 05
  GOAL =  WEIGH               PRODUCTION 09
                                                  MOVE PO[ 1] TO PB[ LEFT]  (     177.200000)
  SUCCEED   WEIGH
 GOAL =  LCATEHEAVIEST
                              PRODUCTION 07
                                                  MOVE PB[RIGHT] TO PF[ 6]  (     178.100000)
 SUCCEED   LCATEHEAVIEST
CURRENT FINAL POSITION:
     182.700000     181.800000     180.800000     180.000000     179.000000     178.100000  0.0000000E+00

GOAL =  SERIATE
                              PRODUCTION 04
                                                  MOVE PB[ LEFT] TO PF[ 7]  (     177.200000)
SUCCEED   SERIATE

FINAL POSITION:
     182.700000     181.800000     180.800000     180.000000     179.000000     178.100000     177.200000
```

APPENDIX V

STAGE 3 (UPGRADED) WEIGHT SERIATION PROGRAM

CARD FILE := CARD.

```
PROGRAM STAGE3;

(*                                                    *)
(*                                                    *)
(*      STAGE 3 (UPGRADED) WEIGHT SERIATION           *)
(*      ----- - ---------- ------ --------            *)
(*                                                    *)
(*                                                    *)


(* SIMULATION OF STAGE 3 WEIGHT SERIATION BEHAVIOR *)
(* INPUT IS A LIST OF WEIGHTS FOR ITEMS TO BE SERIATED *)
(* LIST MAY BE ANY LENGTH, UP TO 20, SEPERATED BY COMMAS *)

CONST
    MAX = 20;       (* MAX. NO. OF ITEMS WHICH CAN BE HANDLED *)
    GSXMAX = 10;        (* MAX. DEPTH OF GOAL STACK *)

TYPE
    GOALSET = (SERIATE, LCATEHEAVIEST, WEIGH, GETRIDOF, NONE);
    BALSTATE = (EMPTY, LEFT, RIGHT, BOTH);
    BALPOS = LEFT .. RIGHT;

VAR
    PO,PF,PI : ARRAY [1 .. MAX] OF REAL;
        (* ORIGINAL, INTERMEDIATE & FINAL POSITIONS *)
    NPO,            (* NO. OF ITEMS IN ORIGINAL POSITION *)
    NPF,            (* NO. OF ITEMS IN FINAL POSITION *)
    NPI,            (* NO. OF ITEMS IN INTERMEDIATE POSITION *)
    NORIG,          (* NO. OF ITEMS ORIGINALLY PRESENTED *)
    K : 0 .. MAX;
    BAL : BALSTATE;
    PB : ARRAY [BALPOS] OF REAL;  (* BALANCE PANS *)
    GOALSTACK : ARRAY [0 .. GSXMAX ] OF GOALSET;
    GSX : 0 .. GSXMAX;
    S : REAL;  (* SEED FOR RANDOM NUMBER GENERATOR *)
```

003:00001:1

```
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
004:0000:1
004:0001:3
004:0002:4
004:0004:0
004:0008:4
004:0009:2
004:0009:2
004:0009:2
003:0000:1
003:0000:1
003:0000:1
006:0000:1
006:0001:2
006:0003:0
006:0003:2
006:0008:5
006:000D:3
006:000E:1
006:000E:1
006:000E:1
003:0000:1
003:0000:1
007:0000:1
007:0002:4
007:0003:2
007:0004:0
007:0004:4
007:0005:4
007:0006:2
007:0006:2
007:0006:2
003:0000:1
003:0000:1
008:0000:1
008:0000:5
008:0000:5
008:0002:0
008:0008:3
008:000B:0
008:000C:1
```

```
      (*                                  *)
      (*                                  *)
      (*        BASIC PROCEDURES          *)
      (*                                  *)
      (*                                  *)

PROCEDURE GOAL(NEWGOAL: GOALSET);
      (* ---- *)
   (* PUSH A NEW GOAL ONTO THE STACK *)
BEGIN
   IF GSX < GSXMAX THEN BEGIN
      GSX := SUCC(GSX);
      GOALSTACK[GSX] := NEWGOAL;
   END ELSE HALT;
END (* GOAL *);

PROCEDURE SUCCEED(OLDGOAL: GOALSET);
      (* ------- *)
   (* PUSH A NEW GOAL ONTO THE STACK *)
BEGIN
   IF GSX > 0 THEN BEGIN
      IF GOALSTACK[GSX] = OLDGOAL THEN
      GSX := GSX - 1
      ELSE HALT;
   END ELSE HALT;
END (* SUCCEED *);

PROCEDURE INITIALISE;
      (* ---------- *)
BEGIN
   GSX := 0; *GOAL(NONE);  GOAL(SERIATE);
   BAL := EMPTY;
   NPF := 0;
   NPI := 0;
   S := 1731;
END (* INITIALISE *);

PROCEDURE INPUTOFDATA;
      (* ---------- *)
BEGIN
   NPO := 0;
   REPEAT
      NPO := NPO + 1;
      READ(INPUT, PO[NPO]);
   UNTIL EOF(INPUT);
   NPO := NPO - 1;
   NORIG := NPO;
```

```
                                                                                                                    008:000D:0
                                                                                                                    008:000D:0
                                                                                                                    008:000D:0
                                                                                                                    008:000D:0
                                                                                                                    008:000D:0
                                                                                                                    008:000D:0
                                                                                                                    008:000D:0
    IF NPO > MAX THEN BEGIN                                                                                         008:000E:2
       WRITELN(OUTPUT, 'TOO MANY ITEMS', NPO, MAX);                                                                 008:0017:3
       NPO := MAX;                                                                                                  008:0018:2
    END;                                                                                                            008:0018:2
    WRITELN(OUTPUT, 'STAGE 3 WEIGHT SERIATION');                                                                    008:001E:2
    WRITELN(OUTPUT, '------------------------');                                                                    008:0024:2
    WRITELN(OUTPUT);                                                                                                008:0028:0
    WRITELN(OUTPUT);                                                                                                008:002B:4
    WRITELN(OUTPUT, 'ORIGINAL POSITION:');                                                                          008:0031:4
    FOR K := 1 TO NPO DO WRITE(OUTPUT, PO[K]);                                                                      008:003D:3
    WRITELN(OUTPUT);                                                                                                008:0041:1
    WRITELN(OUTPUT);                                                                                                008:0044:5
    WRITELN(OUTPUT);                                                                                                008:0048:3
END (* INPUTOFDATA *);                                                                                              008:005E:4
                                                                                                                    008:005E:4
PROCEDURE CURRENTFINALPOS;                                                                                          008:005E:4
      (* ---------------- *)                                                                                       003:0000:1
BEGIN                                                                                                               003:0000:1
    WRITELN(OUTPUT, 'CURRENT FINAL POSITION:');                                                                     00C:0000:1
    FOR K := 1 TO NORIG DO WRITE(OUTPUT, PF[K]);                                                                    00C:0006:1
    WRITELN(OUTPUT);                                                                                                00C:0012:0
    WRITELN(OUTPUT);                                                                                                00C:0015:4
    WRITELN(OUTPUT);                                                                                                00C:0019:2
    WRITELN(OUTPUT);                                                                                                00C:001D:0
END (* CURRENTFINALPOS *);                                                                                          00C:0020:4
                                                                                                                    00C:002C:4
PROCEDURE FINALPOS;                                                                                                 00C:002C:4
      (* --------- *)                                                                                              00C:002C:4
BEGIN                                                                                                               003:0000:1
    WRITELN(OUTPUT);                                                                                                003:0000:1
    WRITELN(OUTPUT);                                                                                                00D:0000:1
    WRITELN(OUTPUT, 'FINAL POSITION:');                                                                             00D:0003:5
    FOR K := 1 TO NPF DO WRITE(OUTPUT, PF[K]);                                                                      00D:0007:3
    WRITELN(OUTPUT);                                                                                                00D:000D:3
    WRITELN(OUTPUT);                                                                                                00D:0019:2
    WRITELN(OUTPUT);                                                                                                00D:001D:0
END (* FINALPOS *);                                                                                                 00D:0020:4
                                                                                                                    00D:0024:2
                                                                                                                    00D:0032:4
                                                                                                                    00D:0032:4
                                                                                                                    00D:0032:4
                                                                                                                    00D:0032:4
                                                                                                                    00D:0032:4
                                                                                                                    00D:0032:4
                                                                                                                    00D:0032:4
                                                                                                                    00D:0032:4
                                                                                                                    00D:0032:4
                                                                                                                    00D:0032:4
                                                                                                                    00D:0032:4
                                                                                                                    00D:0032:4
                                                                                                                    00D:0032:4
```

```
00D:0032:4
00D:0032:4
00D:0032:4
00D:0032:4
00D:0032:4
00D:0032:4
00D:0032:4
00D:0032:4
00D:0032:4
00D:0032:4
00D:0032:4
00D:0032:4
00D:0032:4
00D:0032:4
00D:0032:4
003:0000:1
003:0000:1
003:0000:1
003:0000:1
003:0000:1
00E:0000:1
00E:0000:1
00E:0002:5
00E:0004:0
00E:0005:2
00E:0007:4
00E:000A:0
00E:0019:3
00E:0027:1
00E:002E:5
00E:002F:2
00E:0031:4
00E:003F:2
00E:0047:0
00E:0047:0
00E:0047:0
00E:0048:2
00E:0049:3
00E:0049:3
00E:004B:3
00E:004C:4
00E:0053:4
00E:0053:4
00E:0053:4
003:0000:1
003:0000:1
003:0000:1
010:0000:1
010:0001:2
010:0003:4
010:0011:2
010:0019:0
010:0019:0
010:001A:2
010:001B:3
010:001B:3
010:001D:3
```

```
      (*                                  *)
      (*                                  *)
      (*          PROCEDURES FOR          *)
      (*        MOVEMENT OF WEIGHTS       *)
      (*                                  *)
      (*                                  *)

PROCEDURE PBTOPI(BPOS: BALPOS);
      (* ------ *)
(* MOVEMENT OF WEIGHTS FROM THE BALANCE PANS
                  TO THE INTERMEDIATE POSITION *)

   VAR POS: 1 .. MAX;
BEGIN
   POS := TRUNC(RANDOM(S)*NPI + 1);
   NPI := NPI + 1;
   IF POS < NPI THEN BEGIN
      PI[NPI] := PI[POS];
      PI[POS] := PB[BPOS];
      WRITELN(OUTPUT, ' ':45, 'MOVE PI[', POS:2,'] TO PI[', NPI:2,']');
      WRITELN(OUTPUT, ' ':45, 'MOVE PB[', BPOS:5,'] TO PI[', POS:2,
              '] (',PB[BPOS],')');
   END ELSE BEGIN
      PI[NPI] := PB[BPOS];
      WRITELN(OUTPUT, ' ':45, 'MOVE PB[', BPOS:5,'] TO PI[', NPI:2,
              '] (',PB[BPOS],')');
   END;

   IF BAL = BOTH THEN BEGIN
      IF BPOS = LEFT THEN
         BAL := RIGHT
      ELSE BAL := LEFT;
   END ELSE BAL := EMPTY;
END (* PBTOPI *);

PROCEDURE PBTOPF(BPOS: BALPOS);
      (* ------ *)
(* MOVEMENT OF WEIGHTS FROM THE BALANCE PANS TO THE FINAL POSITION *)
BEGIN
   NPF := NPF + 1;
   PF[NPF] := PB[BPOS];
   WRITELN(OUTPUT, ' ':45, 'MOVE PB[', BPOS:5,'] TO PF[', NPF:2,
           '] (',PB[BPOS],')');

   IF BAL = BOTH THEN BEGIN
      IF BPOS = LEFT THEN
         BAL := RIGHT
      ELSE BAL := LEFT;
   END ELSE BAL := EMPTY;
```

```
                                                                                                                    010:0020:4
                                                                                                                    010:0020:4
                                                                                                                    010:0020:4
                                                                                                                    010:0020:4
                                                                                                                    010:0020:4
                                                                                                                    010:0020:4
PROCEDURE PITOPO;                                                                                                   010:0020:4
        (* ------ *)                                                                                               003:0000:1
(* MOVEMENT OF WEIGHTS FROM THE INTERMEDIATE POSITION                                                             003:0000:1
                              TO THE NEW ORIGINAL POSITION *)                                                       003:0000:1
BEGIN                                                                                                               003:0000:1
    WRITELN(OUTPUT, ' ':45, 'TRANSFORM PI TO PO');                                                                  011:0000:1
    PO := PI;                                                                                                       011:0008:0
    NPO := NPI;                                                                                                     011:000A:3
    NPI := 0;                                                                                                       011:000B:2
    WRITELN(OUTPUT);                                                                                                011:000C:0
    WRITELN(OUTPUT);                                                                                                011:000F:4
    WRITELN(OUTPUT, 'NEW ORIGINAL POSITION:');                                                                      011:0013:2
    FOR K := 1 TO NPO DO WRITE(OUTPUT, PO[K]);                                                                      011:0019:2
    WRITELN(OUTPUT);                                                                                                011:0025:1
    WRITELN(OUTPUT);                                                                                                011:0028:5
    WRITELN(OUTPUT);                                                                                                011:002C:3
END (* PITPPO *);                                                                                                   011:0030:1
                                                                                                                    011:0040:4
                                                                                                                    011:0040:4
PROCEDURE POTOPB(VAR NPO: INTEGER; BPOS: BALPOS);                                                                   011:0040:4
        (* ------ *)                                                                                               003:0000:1
(* MOVEMENT OF WEIGHTS FROM THE ORIGINAL POSITION TO THE BALANCE *)                                               003:0000:1
BEGIN                                                                                                               003:0000:1
    WRITELN(OUTPUT, ' ':45, 'MOVE PO[', NPO:2,'] TO PB[', BPOS:5,                                                  012:0000:1
            '] (', PO[NPO], ')');                                                                                   012:000E:1
    PB[BPOS] := PO[NPO];                                                                                            012:0016:0
    NPO := NPO - 1;                                                                                                 012:0018:2
                                                                                                                    012:0019:4
    IF BAL= EMPTY THEN                                                                                              012:0019:4
        BAL := BPOS                                                                                                 012:001A:5
    ELSE BAL := BOTH;                                                                                               012:001A:5
END (* POTOPB *);                                                                                                   012:001D:0
                                                                                                                    012:001F:4
                                                                                                                    012:001F:4
                                                                                                                    012:001F:4
```

```
                                                                                012:001F:4
                                                                                012:001F:4
                                                                                012:001F:4
                                                                                012:001F:4
                                                                                012:001F:4
                                                                                012:001F:4
                                                                                012:001F:4
                                                                                012:001F:4
    (*                                  *)                                      012:001F:4
    (*                                  *)                                      012:001F:4
    (*        PRODUCTION SYSTEM         *)                                      012:001F:4
    (*                                  *)                                      012:001F:4
    (*                                  *)                                      012:001F:4
                                                                                012:001F:4
                                                                                012:001F:4
PROCEDURE DOSERIATE;                                                            012:001F:4
    (* ---------- *)                                                           003:0000:1
BEGIN                                                                           003:0000:1
  WRITELN(OUTPUT, 'GOAL = ', GOALSTACK[GSX]);                                   013:0000:1
                                                                                013:0009:3
  (* PRODUCTION 1 *)                                                           013:0009:3
  (* ---------- - *)                                                           013:0009:3
  IF NPO > 1 THEN BEGIN                                                         013:0009:3
                                                                                013:000A:4
  WRITELN(OUTPUT, ' ':25, 'PRODUCTION 01');                                     013:000A:4
  GOAL(LOCATEHEAVIEST);                                                         013:0012:4
  END ELSE                                                                      013:0013:3
                                                                                013:0014:0
  (* PRODUCTION 2 *)                                                           013:0014:0
  (* ---------- - *)                                                           013:0014:0
  IF (NPO = 1) AND (BAL IN [LEFT,RIGHT]) THEN BEGIN                             013:0014:0
                                                                                013:0016:3
  WRITELN(OUTPUT, ' ':25, 'PRODUCTION 02');                                     013:0016:3
  GOAL(LOCATEHEAVIEST);                                                         013:001E:3
  END ELSE                                                                      013:001F:2
                                                                                013:001F:5
  (* PRODUCTION 3 *)                                                           013:001F:5
  (* ---------- - *)                                                           013:001F:5
  IF (NPO = 0) AND (NPI > 0) THEN BEGIN                                         013:001F:5
                                                                                013:0021:5
  WRITELN(OUTPUT, ' ':25, 'PRODUCTION 03');                                     013:0021:5
    PITOPO;                                                                     013:0029:5
  END ELSE                                                                      013:002A:3
                                                                                013:002B:0
  (* PRODUCTION 4 *)                                                           013:002B:0
  (* ---------- - *)                                                           013:002B:0
  IF (NPO = 0) AND (NPI = 0) AND (BAL IN [LEFT,RIGHT]) THEN BEGIN               013:002B:0
                                                                                013:002E:2
  WRITELN(OUTPUT, ' ':25, 'PRODUCTION 04');                                     013:002E:2
    IF BAL = LEFT THEN BEGIN                                                    013:0036:2
      PBTOPF(LEFT);                                                             013:0037:3
    END ELSE BEGIN                                                              013:0038:2
      PBTOPF(RIGHT);                                                            013:0038:5
    END;                                                                        013:0039:5
  WRITELN(OUTPUT, 'SUCCEED ', GOALSTACK[GSX]);                                  013:0039:5
  SUCCEED(SERIATE);                                                             013:0043:1
  END;                                                                          013:0044:0
END (* DOSERIATE *);                                                           013:0044:0
                                                                                013:0050:4
```

```
                                                                                013:0050:4
                                                                                013:0050:4
                                                                                013:0050:4
                                                                                013:0050:4
                                                                                013:0050:4
                                                                                013:0050:4
PROCEDURE DOLCATEHEAVIEST;                                                      013:0050:4
      (* ---------------- *)                                                    003:0000:1
BEGIN                                                                           003:0000:1
    WRITELN(OUTPUT, ' GOAL = ', GOALSTACK[GSX]);                                014:0000:1
                                                                                014:0009:3
    (* PRODUCTION 5 *)                                                         014:0009:3
    (* ---------- - *)                                                         014:0009:3
    IF BAL IN [LEFT,RIGHT,EMPTY] THEN BEGIN                                     014:0009:3
                                                                                014:000B:1
    WRITELN(OUTPUT, ' ':25, 'PRODUCTION 05');                                   014:000B:1
       GOAL(WEIGH);                                                             014:0013:1
    END ELSE                                                                    014:0014:1
                                                                                014:0014:4
    (* PRODUCTION 6 *)                                                         014:0014:4
    (* ---------- - *)                                                         014:0014:4
    IF (BAL = BOTH) AND (NPO > 0) THEN BEGIN                                    014:0014:4
                                                                                014:0016:5
    WRITELN(OUTPUT, ' ':25, 'PRODUCTION 06');                                   014:0016:5
       GOAL(GETRIDOF);                                                          014:001E:5
    END ELSE                                                                    014:001F:5
                                                                                014:0020:2
    (* PRODUCTION 7 *)                                                         014:0020:2
    (* ---------- - *)                                                         014:0020:2
    IF (BAL = BOTH) AND (NPO = 0) THEN BEGIN                                    014:0020:2
                                                                                014:0022:3
    WRITELN(OUTPUT, ' ':25, 'PRODUCTION 07');                                   014:0022:3
    IF PB[LEFT] > PB[RIGHT] THEN BEGIN                                          014:002A:3
          PBTOPF(LEFT);                                                         014:002C:1
    END ELSE BEGIN                                                              014:002D:0
          PBTOPF(RIGHT);                                                        014:002D:3
    END;                                                                        014:002E:3
    WRITELN(OUTPUT, ' SUCCEED  ', GOALSTACK[GSX]);                              014:002E:3
    CURRENTFINALPOS;                                                            014:0037:5
    SUCCEED(LCATEHEAVIEST);                                                     014:0038:3
    END;                                                                        014:0039:2
END (* DOLCATEHEAVIEST *);                                                     014:0039:2
                                                                                014:0043:4
                                                                                014:0043:4
PROCEDURE DOWEIGH;                                                              014:0043:4
      (* -------- *)                                                           003:0000:1
BEGIN                                                                           003:0000:1
    WRITELN(OUTPUT, '  GOAL = ', GOALSTACK[GSX]);                               015:0000:1
                                                                                015:0009:3
    (* PRODUCTION 8 *)                                                         015:0009:3
    (* ---------- - *)                                                         015:0009:3
    IF BAL = EMPTY THEN BEGIN                                                   015:0009:3
                                                                                015:000A:4
    WRITELN(OUTPUT, ' ':25, 'PRODUCTION 08');                                   015:000A:4
       POTOPB(NPO,LEFT);                                                        015:0012:4
       POTOPB(NPO,RIGHT);                                                       015:0014:0
    END ELSE BEGIN                                                              015:0015:3
                                                                                015:0016:0
```

```
    (* PRODUCTION 9 *)
    (* ------------ *)
    IF (BAL IN [LEFT,RIGHT]) THEN BEGIN
    WRITELN(OUTPUT, ' ':25, 'PRODUCTION 09');
       IF BAL = LEFT THEN BEGIN
       POTOPB(NPO,RIGHT);
       END ELSE BEGIN
       POTOPB(NPO,LEFT);
       END;
       END;
    END;
    WRITELN(OUTPUT, '  SUCCEED  ', GOALSTACK[GSX]);
       SUCCEED(WEIGH);
END (* DOWEIGH *);

PROCEDURE DOGETRIDOF;
       (* ---------- *)
BEGIN
    WRITELN(OUTPUT, '   GOAL = ', GOALSTACK[GSX]);

    (* PRODUCTION 10 *)
    (* ------------ *)
    WRITELN(OUTPUT, ' ':25, 'PRODUCTION 10');
    IF PB[LEFT] > PB[RIGHT] THEN BEGIN
       PBTOPI(RIGHT);
    END ELSE BEGIN
       PBTOPI(LEFT);
    END;
    WRITELN(OUTPUT, '   SUCCEED  ', GOALSTACK[GSX]);
    SUCCEED(GETRIDOF);
END (* DOGETRIDOF *);
```

```
015:0016:0
015:0016:0
015:0016:0
015:0016:0
015:0016:0
015:0016:0
015:0016:0
015:0016:0
015:0016:0
015:0017:4
015:0017:4
015:001F:4
015:0020:5
015:0022:2
015:0022:5
015:0024:1
015:0024:1
015:0024:1
015:0024:1
015:002D:3
015:002E:3
015:0036:4
015:0036:4
015:0036:4
003:0000:1
003:0000:1
016:0000:1
016:0009:3
016:0009:3
016:0009:3
016:0009:3
016:0011:3
016:0013:1
016:0014:1
016:0014:4
016:0015:3
016:0015:3
016:001E:5
016:001F:5
016:0026:4
016:0026:4
016:0026:4
```

```
                                                                                016:0026:4
                                                                                016:0026:4
                                                                                016:0026:4
                                                                                016:0026:4
                                                                                016:0026:4
                                                                                016:0026:4
                                                                                016:0026:4
                                                                                016:0026:4
    (*                                  *)                                      016:0026:4
    (*                                  *)                                      016:0026:4
    (*          MAIN PROGRAM            *)                                      016:0026:4
    (*                                  *)                                      016:0026:4
    (*                                  *)                                      016:0026:4
                                                                                016:0026:4
                                                                                016:0026:4
BEGIN                                                                           003:0000:1
  INITIALISE;                                                                   003:0000:1
                                                                                003:0000:5
  INPUTOFDATA;                                                                  003:0000:5
                                                                                003:0001:3
  (* PROCESS *)                                                                 003:0001:3
  REPEAT                                                                        003:0001:3
    CASE GOALSTACK[GSX] OF                                                      003:0001:3
                                                                                003:0002:4
    SERIATE:  DOSERIATE;                                                        003:0002:4
                                                                                003:0003:5
    LCATEHEAVIEST:  DOLCATEHEAVIEST;                                            003:0003:5
                                                                                003:0005:0
    WEIGH:  DOWEIGH;                                                            003:0005:0
                                                                                003:0006:1
    GETRIDOF:  DOGETRIDOF;                                                      003:0006:1
                                                                                003:0007:2
    NONE:                                                                       003:0007:2
                                                                                003:0007:2
    END;                                                                        003:0007:2
  UNTIL GOALSTACK[GSX] = NONE;                                                  003:0010:0
                                                                                003:0011:4
  FINALPOS;  (* PRINTING OF FINAL POSITIONS *)                                 003:0011:4
                                                                                003:0012:2
END.                                                                            003:0012:2
=================================COMPILATION SUMMARY=============================
 FILE TITLE: CARD.
ILE TITLE: SERIATION.
D BY COMPILER

DE SEGMENTS    VALUE ARRAYS      EST. STACK     EST. MEMORY    DISK SECTORS
        15               1
       758              88              57            1502              64
=================================================================================
```

STAGE 3 WEIGHT SERIATION

ORIGINAL POSITION:
179.000000 181.800000 178.100000 180.000000 180.800000 177.200000 182.700000

GOAL = SERIATE
GOAL = LCATEHEAVIEST
PRODUCTION 01
GOAL = WEIGH
PRODUCTION 05
PRODUCTION 08
MOVE PO[7] TO PB[LEFT] (182.700000)
MOVE PO[6] TO PB[RIGHT] (177.200000)
SUCCEED WEIGH
GOAL = LCATEHEAVIEST
PRODUCTION 06
GOAL = GETRIDOF
PRODUCTION 10
MOVE PB[RIGHT] TO PI[1] (177.200000)
SUCCEED GETRIDOF
GOAL = LCATEHEAVIEST
PRODUCTION 05
GOAL = WEIGH
PRODUCTION 09
MOVE PO[5] TO PB[RIGHT] (180.800000)
SUCCEED WEIGH
GOAL = LCATEHEAVIEST
PRODUCTION 06
GOAL = GETRIDOF
PRODUCTION 10
MOVE PI[1] TO PI[2]
MOVE PB[RIGHT] TO PI[1] (180.800000)
SUCCEED GETRIDOF
GOAL = LCATEHEAVIEST
PRODUCTION 05
GOAL = WEIGH
PRODUCTION 09
MOVE PO[4] TO PB[RIGHT] (180.000000)
SUCCEED WEIGH
GOAL = LCATEHEAVIEST
PRODUCTION 06
GOAL = GETRIDOF
PRODUCTION 10
MOVE PI[2] TO PI[3]
MOVE PB[RIGHT] TO PI[2] (180.000000)
SUCCEED GETRIDOF
GOAL = LCATEHEAVIEST
PRODUCTION 05
GOAL = WEIGH
PRODUCTION 09
MOVE PO[3] TO PB[RIGHT] (178.100000)
SUCCEED WEIGH
GOAL = LCATEHEAVIEST
PRODUCTION 06
GOAL = GETRIDOF
PRODUCTION 10
MOVE PI[1] TO PI[4]
MOVE PB[RIGHT] TO PI[1] (178.100000)

```
   SUCCEED   GETRIDOF
GOAL =  LCATEHEAVIEST
                        PRODUCTION 05
 GOAL =  WEIGH
                        PRODUCTION 09
                                                MOVE PO[ 2] TO PB[RIGHT]  (     181.800000)
 SUCCEED   WEIGH
GOAL =  LCATEHEAVIEST
                        PRODUCTION 06
  GOAL =  GETRIDOF
                        PRODUCTION 10
                                                MOVE PI[ 2] TO PI[ 5]
                                                MOVE PB[RIGHT] TO PI[ 2]  (     181.800000)
   SUCCEED   GETRIDOF
GOAL =  LCATEHEAVIEST
                        PRODUCTION 05
 GOAL =  WEIGH
                        PRODUCTION 09
                                                MOVE PO[ 1] TO PB[RIGHT]  (     179.000000)
 SUCCEED   WEIGH
GOAL =  LCATEHEAVIEST
                        PRODUCTION 07
                                                MOVE PB[ LEFT] TO PF[ 1]  (     182.700000)
SUCCEED   LCATEHEAVIEST
CURRENT FINAL POSITION:
     182.700000  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00


GOAL =  SERIATE
                        PRODUCTION 03
                                                TRANSFORM PI TO PO

NEW ORIGINAL POSITION:
     178.100000     181.800000     177.200000     180.800000     180.000000

GOAL =  SERIATE
                        PRODUCTION 01
 GOAL =  LCATEHEAVIEST
                        PRODUCTION 05
  GOAL =  WEIGH
                        PRODUCTION 09
                                                MOVE PO[ 5] TO PB[ LEFT]  (     180.000000)
 SUCCEED   WEIGH
GOAL =  LCATEHEAVIEST
                        PRODUCTION 06
  GOAL =  GETRIDOF
                        PRODUCTION 10
                                                MOVE PB[RIGHT] TO PI[ 1]  (     179.000000)
   SUCCEED   GETRIDOF
GOAL =  LCATEHEAVIEST
                        PRODUCTION 05
 GOAL =  WEIGH
                        PRODUCTION 09
                                                MOVE PO[ 4] TO PB[RIGHT]  (     180.800000)
 SUCCEED   WEIGH
GOAL =  LCATEHEAVIEST
                        PRODUCTION 06
```

```
                                PRODUCTION 10
                                                    MOVE PI[ 1] TO PI[ 2]
                                                    MOVE PB[ LEFT] TO PI[ 1]  (      180.000000)
  SUCCEED    GETRIDOF
GOAL =  LCATEHEAVIEST
                                PRODUCTION 05
 GOAL =  WEIGH
                                PRODUCTION 09
                                                    MOVE PO[ 3] TO PB[ LEFT]  (      177.200000)
  SUCCEED    WEIGH
GOAL =  LCATEHEAVIEST
                                PRODUCTION 06
   GOAL =  GETRIDOF
                                PRODUCTION 10
                                                    MOVE PI[ 2] TO PI[ 3]
                                                    MOVE PB[ LEFT] TO PI[ 2]  (      177.200000)
  SUCCEED    GETRIDOF
GOAL =  LCATEHEAVIEST
                                PRODUCTION 05
 GOAL =  WEIGH
                                PRODUCTION 09
                                                    MOVE PO[ 2] TO PB[ LEFT]  (      181.800000)
  SUCCEED    WEIGH
GOAL =  LCATEHEAVIEST
                                PRODUCTION 06
   GOAL =  GETRIDOF
                                PRODUCTION 10
                                                    MOVE PI[ 1] TO PI[ 4]
                                                    MOVE PB[RIGHT] TO PI[ 1]  (      180.800000)
  SUCCEED    GETRIDOF
GOAL =  LCATEHEAVIEST
                                PRODUCTION 05
 GOAL =  WEIGH
                                PRODUCTION 09
                                                    MOVE PO[ 1] TO PB[RIGHT]  (      178.100000)
  SUCCEED    WEIGH
GOAL =  LCATEHEAVIEST
                                PRODUCTION 07
                                                    MOVE PB[ LEFT] TO PF[ 2]  (      181.800000)
  SUCCEED    LCATEHEAVIEST
CURRENT FINAL POSITION:
     182.700000     181.800000  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00

GOAL =  SERIATE
                                PRODUCTION 03
                                                    TRANSFORM PI TO PO

NEW ORIGINAL POSITION:
     180.800000     177.200000     179.000000     180.000000

GOAL =  SERIATE
                                PRODUCTION 01
 GOAL =  LCATEHEAVIEST
                                PRODUCTION 05
```

```
  SUCCEED   WEIGH
GOAL = LCATEHEAVIEST
                        PRODUCTION 06
  GOAL = GETRIDOF
                        PRODUCTION 10
                                          MOVE PB[RIGHT] TO PI[ 1]  (      178.100000)
  SUCCEED   GETRIDOF
GOAL = LCATEHEAVIEST
                        PRODUCTION 05
  GOAL = WEIGH
                        PRODUCTION 09
                                          MOVE PO[ 3] TO PB[RIGHT]  (      179.000000)
  SUCCEED   WEIGH
GOAL = LCATEHEAVIEST
                        PRODUCTION 06
  GOAL = GETRIDOF
                        PRODUCTION 10
                                          MOVE PI[ 1] TO PI[ 2]
                                          MOVE PB[RIGHT] TO PI[ 1]  (      179.000000)
  SUCCEED   GETRIDOF
GOAL = LCATEHEAVIEST
                        PRODUCTION 05
  GOAL = WEIGH
                        PRODUCTION 09
                                          MOVE PO[ 2] TO PB[RIGHT]  (      177.200000)
  SUCCEED   WEIGH
GOAL = LCATEHEAVIEST
                        PRODUCTION 06
  GOAL = GETRIDOF
                        PRODUCTION 10
                                          MOVE PI[ 1] TO PI[ 3]
                                          MOVE PB[RIGHT] TO PI[ 1]  (      177.200000)
  SUCCEED   GETRIDOF
GOAL = LCATEHEAVIEST
                        PRODUCTION 05
  GOAL = WEIGH
                        PRODUCTION 09
                                          MOVE PO[ 1] TO PB[RIGHT]  (      180.800000)
  SUCCEED   WEIGH
GOAL = LCATEHEAVIEST
                        PRODUCTION 07
                                          MOVE PB[RIGHT] TO PF[ 3]  (      180.800000)
SUCCEED   LCATEHEAVIEST
CURRENT FINAL POSITION:
    182.700000     181.800000     180.800000  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00

GOAL = SERIATE
                        PRODUCTION 03
                                          TRANSFORM PI TO PO

NEW ORIGINAL POSITION:
    177.200000     178.100000     179.000000

GOAL = SERIATE
  GOAL = LCATEHEAVIEST  PRODUCTION 01
                        PRODUCTION 05
```

```
GOAL = WEIGH
                                PRODUCTION 09
                                                MOVE PO[ 3] TO PB[RIGHT]  (     179.000000)
SUCCEED   WEIGH
GOAL = LCATEHEAVIEST
                                PRODUCTION 06
  GOAL = GETRIDOF
                                PRODUCTION 10
                                                MOVE PB[RIGHT] TO PI[ 1]  (     179.000000)
SUCCEED   GETRIDOF
GOAL = LCATEHEAVIEST
                                PRODUCTION 05
  GOAL = WEIGH
                                PRODUCTION 09
                                                MOVE PO[ 2] TO PB[RIGHT]  (     178.100000)
SUCCEED   WEIGH
GOAL = LCATEHEAVIEST
                                PRODUCTION 06
  GOAL = GETRIDOF
                                PRODUCTION 10
                                                MOVE PI[ 1] TO PI[ 2]
                                                MOVE PB[RIGHT] TO PI[ 1]  (     178.100000)
SUCCEED   GETRIDOF
GOAL = LCATEHEAVIEST
                                PRODUCTION 05
  GOAL = WEIGH
                                PRODUCTION 09
                                                MOVE PO[ 1] TO PB[RIGHT]  (     177.200000)
SUCCEED   WEIGH
GOAL = LCATEHEAVIEST
                                PRODUCTION 07
                                                MOVE PB[ LEFT] TO PF[ 4]  (     180.000000)
SUCCEED   LCATEHEAVIEST
CURRENT FINAL POSITION:
     182.700000      181.800000      180.800000      180.000000  0.0000000E+00  0.0000000E+00  0.0000000E+00

GOAL = SERIATE
                                PRODUCTION 03
                                                TRANSFORM PI TO PO

NEW ORIGINAL POSITION:
     178.100000      179.000000

GOAL = SERIATE
                                PRODUCTION 01
  GOAL = LCATEHEAVIEST
                                PRODUCTION 05
    GOAL = WEIGH
                                PRODUCTION 09
                                                MOVE PO[ 2] TO PB[ LEFT]  (     179.000000)
  SUCCEED   WEIGH
GOAL = LCATEHEAVIEST
                                PRODUCTION 06
    GOAL = GETRIDOF
                                PRODUCTION 10
                                                MOVE PB[RIGHT] TO PI[ 1]  (     177.200000)
  SUCCEED   GETRIDOF
```

```
GOAL =  LCATEHEAVIEST
                        PRODUCTION 05
 GOAL =  WEIGH
                        PRODUCTION 09
                                        MOVE PO[ 1] TO PB[RIGHT]  (     178.100000)
 SUCCEED   WEIGH
GOAL =  LCATEHEAVIEST
                        PRODUCTION 07
                                        MOVE PB[ LEFT] TO PF[ 5]  (     179.000000)
SUCCEED   LCATEHEAVIEST
CURRENT FINAL POSITION:
     182.700000     181.800000     180.800000     180.000000     179.000000  0.0000000E+00  0.0000000E+00

GOAL =  SERIATE
                        PRODUCTION 03
                                        TRANSFORM PI TO PO

NEW ORIGINAL POSITION:
     177.200000

GOAL =  SERIATE
                        PRODUCTION 02
 GOAL =  LCATEHEAVIEST
                        PRODUCTION 05
  GOAL =  WEIGH
                        PRODUCTION 09
                                        MOVE PO[ 1] TO PB[ LEFT]  (     177.200000)
  SUCCEED   WEIGH
 GOAL =  LCATEHEAVIEST
                        PRODUCTION 07
                                        MOVE PB[RIGHT] TO PF[ 6]  (     178.100000)
 SUCCEED   LCATEHEAVIEST
CURRENT FINAL POSITION:
     182.700000     181.800000     180.800000     180.000000     179.000000     178.100000  0.0000000E+00

GOAL =  SERIATE
                        PRODUCTION 04
                                        MOVE PB[ LEFT] TO PF[ 7]  (     177.200000)
SUCCEED   SERIATE

FINAL POSITION:
     182.700000     181.800000     180.800000     180.000000     179.000000     178.100000     177.200000
```

APPENDIX VI

OTHER TRANSITION MECHANISMS

What other mechanisms or processes can be posited which may govern the child's movement through the developmental sequence? Piaget's formulations unfortunately are highly general statements which do not deal with the particular mechanisms governing developmental changes or specify the conditions under which they take place. In this chapter I will attempt a brief review of a number of different approaches to the problem.

1. Klahr and Wallace

To understand the transition mechanism proposed by Klahr and Wallace (1976), we need to have some background information of the structure of the human information-processor as conceived by them. The main unit, of relevance to this discussion, is the long-term memory (LTM). The LTM consists of a complex list of production rules (as outlined in an earlier section). This list can be divided into three tiers. The first tier contains the productions and production systems derived as a result of specific individual experiences. Consistencies provided by on-going experience are detected and added to the system's LTM, and are geared to the specific experience of the system. The second tier contains a repertoire of problem solving strategies. These

productions are much less differentiated than those on the first tier, and represent possible modes of reaction in situations in which the previous experience of the system has not given rise to relevant consistent sequences. The third tier consists of productions and production systems which underlie the self-modification capability of the information-processor. It comprises the productions involved in the processes of detecting consistent sequences and rule generation, and is of fundamental importance, as it is the source of cognitive development as a whole.

Looking specifically at Tier 3, we can see that Piaget's concept of equilibration has been replaced in this approach by a production system of a more general and more basic nature. Klahr and Wallace (1976) assume the existence of certain "innate productions" comparable to Piaget's innate "functional kernel" of equilibration. These innate productions perform such processes as detecting consistencies and eliminating redundancies. A concrete example of this second process is given by Klahr and Wallace (1976) and is worth reporting as the same basic process has been observed by the author.

Two children aged 11.4 years and 8.5 years both succeed on a weight seriation task by using the "Locate Heaviest" strategy, in which all the blocks are weighed in turn, and the heaviest of each block pair is retained on the balance. The child of 11.4 years takes 16 comparisons to construct the series, mainly because he realizes that, when one weight which is heavier than all those previously weighed is suddenly lighter than the last one, it must be the second heaviest. The boy of 8.5 years does not realize this, and proceeds to weigh the whole remaining weights again, requiring 21 comparisons. Thus a fundamental rule preventing redundant comparisons has developed a specific production rule which enables the child to operate more efficiently.

Klahr and Wallace (1973) feel that sequential regularity detection is a fundamental aspect of cognitive development, with the knowledge of a series of changes preceding

any external knowledge of things. Klahr and Wallace point out that their concepts have yet to be verified, but the ideas have sufficient within them to be expressible, eventually, in fairly concrete terms, possibly in the form of a running computer program.

This brief presentation of Klahr and Wallace's approach also helps to show the powerfulness of the concept of a production system, which could be extended down to the basis of intellectual development itself.

2. Pascual-Leone

Pascual-Leone (Pascual-Leone & Smith, 1969; Pascual-Leone, 1970) outlines his belief that the child as a psychological system has three basic components: a repertoire (H) of behavioral units, a central computing space (M), and a number of organizational laws. He describes the repertoire (H) as containing schemes (or "plans of action" (Pascual-Leone & Smith, 1969, p.329)) outlined in S-R terminology as an ordered pair of implicit responses (e.g. s,r), or as an ordered triplet, (s,r, s') (Tolman, 1959). The schemes are recursive and hence very powerful. He states: 'A superordinate scheme or superscheme is analogous to a computer program which uses subroutines (i.e. subordinate schemes or subschemes) which are stored elsewhere in the subject's memory or repertoire H (e.g. Reitman, 1965)' (Pascual-Leone, 1970, p.306). It is easy to see that this concept is closely analogous to the idea of a production system stored in LTM, which has been outlined in earlier sections.

However the important aspect of Pascual-Leone's work is with his concept of a central processor M. The size of M increases in a lawful manner as development proceeds. 'The general structural characteristics of the piagetian stages would then be interpretable as qualitative manifestations of this internal computing system or M operator' (Pascual-Leone, 1970, p.304). He makes a distinction between the S's maximum capacity (structural M - M_s) and his functional M (M_f), or the amount of M_s space actually used. M_f can vary from zero to M_s, depending on the influence of

a large number of factors, such as the degree of fatigue, the degree of arousal, and various individual difference variables such as Witkin's field-dependence-independence (Witkin et al, 1962).

The computing space M transforms and coordinates the information initially available to the psychological system. The maximum number of schemes that M can attend to or integrate in a single act is assumed to increase as a function of age, and can be considered as a quantitative characteristic of each developmental stage. Pascual-Leone feels that such a concept can be integrated with Miller's (1956) "magical number seven" to indicate the upper limit of the adult's computing space M.

A particularly useful aspect of Pascual-Leone's theoretical work is the fact that it includes as a major feature a cognitive-developmental variable which could not be handled by any other information-processing models available. He ties his concept back to Piaget's expression "field of equilibrium" (Piaget, 1956 in Pascual-Leone, 1970). This can be seen as an important development of this Piagetian concept which creates a further link between Piaget's work and the various information-processing models developing to explain it.

REFERENCES

ACKOFF, R.L., Gupta, S.K. and Minas, J.S. (1962)
Scientific method: Optimizing applied research decisions. New York: Wiley.

APPEL, M.H. and Goldberg, L.S. (Eds.) (1977)
Topics in cognitive development: Volume 1: Equilibration - Theory, research, and application. New York: Plenum Press.

APTER, M.J. (1970)
The computer simulation of behavior. New York: Harper and Row.

ASHBY, W.R. (1956)
An introduction to cybernetics. London: Chapman and Hall.

BARTON, R.E. (1970)
A primer on simulation and gaming. Englewood Cliffs: Prentice-Hall.

BAYLOR, G.W. and Gascon, J. (1974)
An information processing theory of aspects of the development of weight seriation in children. Cognitive Psychology, 6, 1 - 40.

BAYLOR, G.W. and Lemoyne, G. (1975)
Experiments in seriation with children: Toward an information processing explanation of the horizontal

décalage. Canadian Journal of Behavioral Science, 7(1), 4 - 29.

BEISHON, J. and Peters, G. (Eds.) (1976)
Systems behavior. 2nd ed. London: Harper and Row.

BERNOULLI, D. (1738)
Exposition of a new theory on the measurement of risk. (Translated by L. Sommer from 'Specimen theoriae novae de mensura sortis', Commentarii Academiae Scientiarum Imperialis Petropolitanane, Tomus V. (Papers of the Imperial Academy of Sciences in Petersburg, Volume 5, 175 - 192.)). (From Miller, 1964.)

BOBROW, D.G. and Collins, A. (1975)
Representation and understanding: Studies in cognitive science. New York: Academic Press.

BOULDING, K.E. (1956)
The image: Knowledge in life and society. Ann Arbour, Michigan: The University of Michigan Press.

BRAINE, M.D.S. (1959)
The ontogeny of certain logical operations: Piaget's formulation examined by nonverbal methods. Psychological Monographs, 73(5), (Whole number 475).

BRAINE, M.D.S. (1962)
Piaget on reasoning: A methodological critique and alternative proposal. In Kessen, W. and Kuhlmann, C. (Eds.) Thought in the young child. Monographs of the Society for Research in Child Development, 27(2), p. 33 - 53.

BRAINE, M.D.S. (1964)
Development of a grasp of transitivity of length: A reply to Smedslund. Child Development, 35, 799 - 810.

CHECKLAND, P.B. (1976)
Toward a systems-based methodology for real-world problem solving. In Beishon, J. and Peters, G. (Eds.) Systems Behavior. 2nd ed. London: Harper and Row. p. 51 - 77.

CHOMSKY, A.N. (1957)
Syntactic structures. The Hague: Mouton.

DeGREENE, K.B. (1976)
Systems and psychology. In Beishon, J. and Peters, G. (Eds.) Systems behavior. 2nd ed. London: Harper and Row. p. 114 - 149.

DEUTSCH, J.A. (1960)
The structural basis of behavior. London: Cambridge University Press.

DUCKWORTH, E. (1964)
Piaget rediscovered. In Ripple, R.E. and Rockcastle, V.N. (Eds.) Piaget rediscovered. Journal of Research in Science Teaching, 2, 167 - 252. p. 172 - 175.

DUTTON, J.M. and Briggs, W.G. (1971)
Simulation model construction. In Dutton, J.M. and Starbuck, W.H. (Eds.) Computer simulation of human behavior. New York: Wiley. p. 103 - 127.

DUTTON, J.M. and Starbuck, W.H. (1971)
Computer simulation of human behavior. New York: Wiley.

ESTES, W.K., Koch,S., MacCorquodale, K., Meehl, P.E., Mueller, C.G., Schoenfeld, W.N. and Verplank, W.S. (1954)
Modern learning theory: A critical analysis of five examples. New York: Appleton-Century-Crofts.

FELDMAN, J. (1971)
Simulation of behavior in the binary choice experiment. In Dutton, J.M. and Starbuck, W.H. (Eds.) Computer simulation of human behavior. New York: Wiley. p. 170 - 182.

FLAVELL, J.H. (1963)
The developmental psychology of Jean Piaget. New York: Van Nostrand.

FURTH, H.G. (1969)
Piaget and knowledge: Theoretical foundations. Englewood Cliffs: Prentice-Hall.

GALLAGHER, J.M. (1977)
Piaget's concept of equilibration. In Appel, M.H. and Goldberg, L.S. (Eds.) Topics in cognitive development:

Volume 1: Equilibration - Theory, research, and application. New York: Plenum Press. p. 21 - 32.

GASCON, J. and Baylor, G.W. (1973)
B.G. Manual. Working paper M.C.P. #13, Université de Montréal.

GEORGE, F.H. (1965)
Cybernetics and biology. London: Oliver and Boyd.

GLUSKOV, V.M. (1966)
Introduction to cybernetics. New York: Academic Press.

GOLLIN, E.S., Moody, M. and Schadler, M. (1974)
Relational learning of a size concept. Developmental Psychology, 10(1), 101 - 107.

GRODINS, F.S. (1965)
Computer simulation of cybernetic systems. In Stacy, R.W. and Waxman, B.D. (Eds.) Computers in biomedical research (Volume 1). New York: Academic Press. p. 135 - 164.

GUILBAUD, G.T. (1959)
What is cybernetics? (Translated by V. Mackay.) London: Heinemann.

GULLAHORN, J.T. and Gullahorn, J.E. (1971)
Computer simulation of role conflict resolution. In Dutton, J.M. and Starbuck, W.H. (Eds.) Computer simulation of human behavior. New York: Wiley. p. 350 - 364.

HAGGETT, P. (1976)
On systems and models. In Beishon, J. and Peters, G. (Eds.) Systems Behavior. 2nd ed. London: Harper and Row. p. 314 - 318.

HOPCRAFT, J.E. and Ullman, J.D. (1969)
Formal languages and their relation to automata. Reading, Mass.: Addison-Wesley.

INHELDER, B. and Piaget, J. (1964)
The early growth of logic in the child. New York: Harper and Row.

JENKINS, G.M. (1976)
The systems approach. In Beishon, J, and Peters, G. Systems Behavior. 2nd ed. London: Harper and Row. p. 78 - 104.

KAST, F.E. and Rosenzweig, J.E. (1976)
The modern view: A systems approach. In Beishon, J. and Peters, G. (Eds.) Systems behavior. 2nd ed. London: Harper and Row. p. 15 - 29.

KLAHR, D. (1973)
An information processing approach to the study of cognitive development. In Pick, A.D. (Ed.) Minnesota symposia on child psychology: Volume 7. Minneapolis: University of Minnesota Press.

KLAHR, D. (1976)
Steps toward the simulation of intellectual development. In Resnick, L.B. (Ed.) The nature of intelligence. Hillsdale, N.J.: Lawrence Erlbaum Association.

KLAHR, D. and Wallace, J.G. (1970)
An information processing analysis of some Piagetian experimental tasks. Cognitive Psychology, 1, 358 - 387.

KLAHR, D. and Wallace, J.G. (1973)
The role of quantification operators in the development of conservation of quantity. Cognitive Psychology, 4, 301 - 327.

KLAHR, D. and Wallace, J.G. (1976)
Cognitive development: An information processing view. Hillsdale, N.J.: Lawrence Erlbaum Associates.

LAMENDELLA, J.T. (1975)
Theories of the brain; behavior; the mind; robots and cognitive processes, respectively. In Storer, T. and Winter, D. Formal aspects of cognitive processes. (Lecture notes in computer science, Volume 22 - G. Goos and J. Hartmanis (Eds.).) Berlin: Springer-Verlag. p. 4 - 13.

LAUGHERY, K.R. and Gregg, L.W. (1971)
Simulation of human problem-solving behavior. In Dutton, J.M. and Starbuck, W.H. (Eds.) Computer simulation of human behavior. New York: Wiley. p. 183 - 194.

LEWIN, K. (1936)
Principles of topological psychology. New York: McGraw-Hill.

LEWIN, K. (1965)
Formalization and progress in psychology. In Lindzey, G. and Hall, C.S. Theories of personality: Primary sources and research. New York: Wiley.

MAIER, H.W. (1965)
Three theories of child development: Contributions of Erik H. Erikson, Jean Piaget and Robert R. Sears, and their applications. New York: Harper and Row.

MASLOW, A.H. (1967)
A theory of metamotivation: The biological rooting of the value life. Journal of Humanistic Psychology, 7(2), 93 - 127.

MASLOW, A.H. (1970)
Motivation and Personality. 2nd ed. New York: Harper and Row.

MILLER, G.A. (1956)
The magical number seven plus or minus two: Some limits on our capacity for processing information. Psychological Review, 63, 81 - 97.

MILLER, G.A. (1964)
Mathematics and psychology. New York: Wiley.

MINSKY, M. (1967)
Computation: Finite and infinite machines. Englewood Cliffs: Prentice-Hall.

MODGIL, S. (1974)
Piagetian research: A handbook of recent studies. Windsor, Berks.: N.F.E.R.

MODGIL, S. and Modgil, C. (1976)
Piagetian Research: Compilation and commentary. (Volume 3: The growth of logic-concrete and formal operations). Windsor, Berks.: N.F.E.R.

NEWELL, A. (1962)
A guide to GPS-2-2 program. Pittsburgh: Carnegie Institute of Technology.

NEWELL, A. (1972)
A theoretical exploration of mechanisms for coding the stimulus. In Melton, A.W. and Martin, E. (Eds.) Coding processes in human memory. Washington, D.C.: Winston. p. 373 - 434.

NEWELL, A. (1973)
You can't play 20 questions with nature and win. In Chase, W.G. (Ed.) Visual information processing. New York: Academic Press. p. 283 - 308.

NEWELL, A. (1973a)
Artificial intelligence and the concept of mind. In Schank, R.C. and Colby, K.M. (Eds.) Computer models of thought and language. San Francisco: Freeman.

NEWELL, A. (1973b)
Production systems: Models of control structures. In Chase, W.G. (Ed.) Visual information processing. New York: Academic Press. p. 463 - 526.

NEWELL, A. and Simon, H.A. (1971)
Simulation of human thought. In Dutton, J.M. and Starbuck, W.H. (Eds.) Computer simulation of human behavior. New York: Wiley. p. 150 - 170.

NEWELL, A. and Simon, H.A. (1972)
Human problem solving. Englewood Cliffs: Prentice-Hall.

O'REILLY, E. and Steger, J.A. (1970)
Children's use of context in judgement of weight. Child Development, 41, 1095 - 1101.

PASCUAL-LEONE, J. (1970)
A mathematical model for the transition rule in Piaget's

developmental stages. Acta Psychologica, 32, 301 - 345.

PASCUAL-LEONE, J. and Smith, J. (1969)
The encoding and decoding of symbols by children: A new experimental paradigm and a Neo-Piagetian model. Journal of Experimental Child Psychology, 8, 328 - 355.

PASK, G. (1961)
An approach to cybernetics. London: Hutchinson.

PHILLIPS, J.L. (1969)
The origins of intellect: Piaget's theory. San Francisco: Freeman.

PIAGET, J. (1942)
Les trois structures fondamentales de la vie psychiques: Rythme, régulation et groupement. Revue Suisse Psychologie, 1, 9 - 21.

PIAGET, J. (1949)
Traite de logique: Essai de logistique operatoire. Paris: Colin.

PIAGET, J. (1952)
The child's conception of number. London: Routledge and Kegan Paul.

PIAGET, J. (1952a)
Autobiography of Jean Piaget. In Langfeld, H.S., Boring, E.G., Werner, H. and Yerkes, R.M. (Eds.) A history of psychology. Worcester, Mass.: Clark University Press.

PIAGET, J. (1953)
Structures opérationelles et cybérnetique. Année Psychologie, 53, 379 - 390.

PIAGET, J. (1954)
Problem of consciousness and symbolic processes. In Abramson, H.E. Problems of consciousness. Transactions of the fourth conference, New York: Josiah Macy, Jr. Foundation.

PIAGET, J. (1956)
Logique et équilibre dans les comportements du sujet. In Apostel, L., Mandelbrot, B. and Piaget, J. (Eds.)

Logique et équilibre. Paris: P.U.F. p. 27 - 117.

PIAGET, J. (1957)
Logic and psychology. New York: Basic Books.

PIAGET, J. (1964)
Development and learning. In Ripple, R.E. and Rockcastle, V.N. (Eds.) Piaget rediscovered: Journal of Research in Science Teaching, 2, 167 - 252. p. 176 - 186.

PIAGET, J. (1970)
Piaget's theory. In Mussen, P.H. (Ed.) Carmichael's manual of child psychology. 3rd ed. New York: Wiley. p. 703 - 732.

PIAGET, J. (1971)
Biology and knowledge. Chicago: University of Chicago Press.

PIAGET, J. (1977)
The grasp of consciousness: Action and concept in the young child. London: Routledge and Kegan Paul.

PIAGET, J. (1977a)
Problems of equilibration. In Appel, M.H. and Goldberg, L.S. (Eds.) Topics in cognitive development Volume 1: Equilibration - Theory, research and application. New York: Plenum Press. p. 3 - 13.

PIAGET, J. and Szeminska, A. (1941)
La genèse du nombre chez l'enfant (The development of the number concept in the child). Neuchâtel, Paris: Delachaux, Niestlé. (From Psychological Abstracts, 1943, 17, number 2938)

POST, E.L. (1943)
Formal reductions of the general combinatorial decision problem. American Journal of Mathematics, 65, 197 - 268.

REITMAN, W.R. (1965)
Cognition and thought: An information processing approach. New York: Wiley.

SAMMET, J.E. (1969)
Programming languages: History and fundamentals. Englewood Cliffs, N.J.: Prentice-Hall.

SIEGEL, L.S. (1971)
The sequence of development of certain number concepts in preschool children. Developmental Psychology, 5(2), 357 - 361.

SIEGEL, L.S. (1972)
Development of the concept of seriation. Developmental Psychology, 6(1), 135 - 137.

SINHA, C. and Walkerdine, V. (1978)
Children, logic and learning. New Society, 12th January.

SKINNER, B.F. (1971)
Beyond freedom and dignity. New York: Knopf.

SMEDSLUND, J. (1963)
Development of concrete transitivity of length in children. Child Development, 34, 389 - 405.

SMEDSLUND, J. (1963a)
The acquisition of transitivity of weight in 5 to 7 year old children. Journal of Genetic Psychology, 102, 245 - 255.

SMEDSLUND, J. (1965)
The development of transitivity of length: A comment on Braine's reply. Child Development, 36, 577 - 580.

STARBUCK, W.H. and Dutton, J.M. (1971)
The history of simulation models. In Dutton, J.M. and Starbuck, W.H. (Eds.) Computer simulation of human behavior. New York: Wiley. p. 9 - 103.

STERNBERG, S. (1970)
Mental scanning: Mental processes revealed by reaction time experiments. In Antrobus, J.S. (Ed.) Cognition and effect. Boston: Little Brown. p. 13 - 58.

TANNER, J.M. and Inhelder, B. (Eds.) (1960)
Discussions on child development (Volume 4). London: Tavistock.

TOLMAN, E.C. (1959)
Principles of purposive behavior. In Koch, S. (Ed.) Psychology: A study of a science (Volume 2). New York: McGraw-Hill. p. 92 - 157.

VON BERTALANFFY, L. (1951)
General system theory: A new approach to unity of science. I. Problems of general system theory. Human Biology, 23, 302 - 312.

VON BERTALANFFY, L. (1960)
General system theory and the behavioral sciences. In Tanner, J.M. and Inhelder, B. (Eds.) Discussions on child development (Volume 4). London: Tavistock. p. 155 - 175.

WADSWORTH, B.J. (1971)
Piaget's theory of cognitive development. New York: McKay.

WATERMAN, D.A. and Newell, A. Protocol analysis as a task for artificial intelligence. Artificial Intelligence, 2, 285 - 318.

WATERMAN, D.A. (1974)
Adaptive production systems. C.I.P. Working Paper 285. Pittsburgh: Carnegie-Mellon University, Department of Psychology.

WIENER, N. (1948)
Cybernetics: Or control and communication in the animal and the machine. Mass.: M.I.T. Press.

WITKIN, H.A., Dyk, R.B., Faterson, H.A., Goodenough, D.R. and Karp, S.A. (1962)
Psychological differentiation. New York: Wiley.

YOUNG, R.M. (1973)
Children's seriation behavior: A production-system analysis. Unpublished Doctoral Dissertation. Carnegie-Mellon University.