Knowledge Structures

D. Albert (Ed.)

March 24, 1994

The Authors

Albert, Dietrich (Ed.)
Institut für Psychologie
Karl-Franzens-Universität Graz
Universitätsplatz 2/III
A-8010 Graz

Doignon, Jean-Paul
Université Libre de Bruxelles
Département de Mathématiques
c.p. 216, Bd du Triomphe
B-1050 Bruxelles

Dowling, Cornelia E.
Technische Universität Braunschweig
Institut für Psychologie
Postfach 3329
D-39106 Braunschweig

Held, Theo
Universität Heidelberg
Psychologisches Institut
Hauptstraße 47–51
D-69117 Heidelberg

Heller, Jürgen
Institut für Psychologie
Universität Regensburg
Universitätsstraße 31
Gebäude PT
D-93040 Regensburg

Rodenhausen, Hermann
Universität zu Köln
Seminar für Mathematik und
ihre Didaktik
Gronewaldstraße 2
D-50931 Köln

Sobik, Fred
Universität Potsdam
Institut für Computerintegrierte
Systeme GmbH
Karl-Liebknecht-Straße
D-14476 Golm

Sommerfeld, Erdmute
Friedrich-Schiller-Universität
Institut für Psychologie
Leutragraben 1
D–07743 Jena

Witteveen, Cees
Delft University of Technology
Department of Mathematics and
Computer Science
Julianalaan 132
NL–2628 BL Delft

Preface

This book is a sign of its times. Each one of the chapters – papers written by European authors of various backgrounds – illustrates a departure from the style of theorizing that has been prominent in the social sciences for most of the century. Until very recently, models for behavioral phenomena were chiefly based on numerical representations of the objects of concern, e.g. the subjects and the stimuli under study. This was due in large part to the influence of nineteenth century physics, which played the role of the successful older sister, the one that had to be imitated if one wished to be taken seriously in scientific circles. The mystical belief that there could be science only when the objects of concern were susceptible of measurement in the sense of physics was a credo that could not be violated without risks. Another, more honorable justification was that the numerical models were the only ones capable of feasible calculations. (In fact, these models were typically linear.) An early example of such theorizing in psychology is factor analysis, which attempted to represent the results of mental tests in a real vector space of small dimensionality, each subject being represented by a point in that space. A dimension was interpreted as a scale measuring some mental ability. The analysis was simple, and only required an electrical desk calculator (with spinning wheels), and a suitable amount of determination. Psychometric models and practices, which are currently the object of much criticisms, are in the same vein.

The advent of the computer marked the end of this era. To be sure, the effect was not immediate, and computers were first used for particularly long or difficult (e.g. nonlinear) numerical calculations. In the behavioral sciences, factor analysis was replaced by non metric multidimensional scaling as a choice representation technique. The numerical measurement credo was not so easily repudiated.

Nowadays, however, large scale computer searches in combinatoric structures, for example in order to find the best exemplar for some piece of data in a vast, but finite class of models, are becoming more frequent, and the style of modeling is changing accordingly. A representative model maybe a hypergraph or an order relation of some kind, sometimes equipped with a probabilistic structure. Measurement in the traditional sense of physics plays a decreasing role. The seven chapters of the book offer good examples of this trend. Note that all chapters contain exercises, which is a token of the authors' confidence that their work is not a manifestation of a passing fashion.

The first three chapters deal with various issues in knowledge space theory. The central concept, the 'knowledge space', is a hypergraph, the edges of which represent possible states of knowledge of subjects in a population, with respect to a specific field of information. An important problem is that of finding the knowledge state that best represents a subject. In his chapter, Jean-Paul Doignon reviews a number of Markovian-type assessment procedures for that purpose. The standard implementation is on a computer. The subject is asked questions, successively chosen so as to narrow down as quickly as possible the set of knowledge states consistent with the responses given. Probabilities are introduced to render the search procedure less vulnerable to errors of various kinds.

The most critical aspect of knowledge space research certainly concerns the practical construction of the family of feasible knowledge states in some empirical situation. One obvious method consists in interviewing a number of experts in the field (for example, some experienced teachers in the field). Unfortunately, even though these experts may have some implicit awareness of the knowledge states which may be present in a population of subjects under consideration, the list of of such states cannot be obtained by straightforward questioning. The list is too long, for one thing. For another, it is far from clear that an expert's representation of the knowledge states is in the form required by the theory. For instance, it is doubtful that Johnny's mathematics teacher has a representation of Johnny's knowledge state as the set of all 126 mathematics questions (say, within the standard mathematics curriculum) that Johnny is capable of solving. The relevant information can be elicited from the experts, but indirect methods must be used. Cornelia Dowling's chapter is devoted to this issue. She reviews some basic results concerning an important mathematical relation equivalent to a knowledge space. She then shows how this relation can be used to set up a practical method for interviewing experts. In Chapter 3, Dietrich Albert and Theo Held take a different tack, and propose to construct the knowledge space by a systematical analysis of the content of the problems in terms of basic components. They discuss in details two applications of their ideas: chess problems and the continuation of a sequence of numbers. They also consider the relationship between their knowledge representation scheme and decision theory.

Next in order comes a chapter by Jürgen Heller on lattice representations of semantic features. After reviewing some standard material on lattices and the 'concept lattices' of Wille and Ganter, he formulates a representation theorem for the data structure generated by a particular experimental paradigm, which he also describes. An exemplary empirical application is given.

In their chapter, Erdmute Sommerfeld and Fred Sobik use graph theory to represent cognitive structures and their transformations. Each vertex and each edge of a representing graph is assigned a label signaling some elementary property, and carrying a cognitive interpretation of the graph. The chapter contains many examples of applications of these concepts to empirical situations.

Chapter 6, due to Cees Witteveen, explores an intriguing idea. Representing the behavior of an organism as the output of a production system, he asks: "How can we infer its set of rules and control structure?" In general, this problem does not have a unique solution. Moreover, as argued by the author, it is not clear that a solution can be found in finite time. In any event, the chapter focuses on a somewhat simpler problem, namely: given the set of rules (or *procedural knowledge*), what control structure can generate the behavior? The lack of uniqueness is dealt with by considering a special class of "most efficient" control structures, where *efficiency* is defined in terms of the ratio of the number of rules executed over the number of rules applied. A practical method (an algorithm, actually) is described to construct such a most efficient control structure, based on the behavior of the system.

The last chapter, by Hermann Rodenhausen, contains a theoretical discussion of the concept of self organization. A motivation for this concept is that regularities observed in the functional structure of the brain, e.g. neighboring cells on the retina are mapped onto neighboring cells in the cortex, cannot be explained solely by genetic factors, or so it is argued by some researchers. Following the line of Kohonen, Buhmann and others, Rodenhausen discusses the possibility that topology preserving neural mappings could be generated by inputs with an appropriate statistical structure. Using Kohonen's formalism and the framework of Markov processes, Rodenhausen presents some exemplary results showing how order could emerge as the limit of a converging random process.

As suggested by this bird's-eye survey, we are dealing here with a collection of essays of a kind that would have appeared rather odd a couple of decades ago. But times and mores have changed. Hypergraphs, lattices, production systems, neural nets, have become household concepts in the cognitive sciences. To this reviewer, this trend is indicative of a major, welcome shift. As witnessed by the book edited by Dietrich Albert, the junior sister is claiming her identity.

Jean-Claude Falmagne
Irvine, November 30, 1993

Contents

List of Tables

List of Figures

1 Probabilistic assessment of knowledge

Jean-Paul Doignon[1]
Université Libre de Bruxelles, Département de Mathématiques, c.p. 216, Bd du Triomphe, B–1050 Bruxelles, Belgium
E-mail: doignon@ulb.ac.be

1.1 Introduction

An easy and common way of assessing a student's knowledge consists of a written examination. A list of questions is presented, the student's answers are collected, and finally the examiner returns an appreciation, which usually boils down to a single number or percentage. Table 1.1 presents an excerpt of such a test in elementary arithmetics and will be used for exemplary purpose. We first argue that the information provided by the testing procedure is poorly reflected by a single number. Knowing that a student provided correct answers only to questions, say, a, c, and e, entails more than a numerical appreciation (60% correct) of his or her work. It shows mastery in performing multiplications, and deficiency in division operations. Weaknesses and strengthes of the student's preparation have thus been revealed. Hence advices for further study can be inferred. Obtaining and exploiting the most precise information from

Table 1.1. An excerpt of a test in arithmetics.

a	$2 \times 378 =$	
b	$322 \div 7 =$	
c	$14.7 \times 100 =$	
d	$6442 \div 16 =$	
e	$58.7 \times 0.94 =$	

an assessment procedure is particularly needed in programmed courses. Any computer-assisted instruction system should entail a module for uncovering the user's knowledge. A straightforward, set-theoretic model for encoding this knowledge will be introduced below. This model constitutes the framework in which we will design automatic procedures for knowledge assessment. Such procedures should be efficient in many different senses that will be explicited

[1]Our work in this area is supported by NSF grant IRI 8919068 to Jean-Claude Falmagne at the University of California, Irvine. We thank the editor and three anonymous referees for their careful reading of a preliminary version of the manuscript, and for their useful remarks.

later on. In particular, they must avoid lengthy, multiple interrogations (by taking advantage of the underlying structure of the material to be learned). Only a few of the encoded questions will be asked – as few as possible is the ultimate goal. Clearly, not all of the questions must be asked when the answers collected at some time allow to draw inferences for other questions. In our example, from a positive answer to question e we could surmise a positive answer to questions a and c.

Deterministic procedures are easy to conceive, relying on a definite structuring of the material, and on answers reflecting a clearly cut, steady state of student's knowledge. However, we may not take the answers as data perfectly reflecting the examinee's knowledge. The responses we collect suffer from various random perturbations, e. g. due to careless errors (in computations, in transcribing the answer, etc.) or lucky guesses (which are not limited to multiple choice tests). Thus, our procedures must take into account these perturbations inherent to the student's behavior. As a result, the same question will be asked perhaps more than one time – in some variant of another. A well-defined questioning rule chooses the next question on the basis of the (probabilistic) available information.

The aim of this chapter is to describe a theoretical model for the assessment of knowledge that integrates such probabilistic aspects. We first need to spell out a combinatorial formalization of the knowledge state of an individual with respect to a given body of information. We thus define knowledge structures, knowledge spaces and well-graded knowledge spaces. After sketching deterministic assessment-procedures, we introduce a general setting for probabilistic procedures. Precise definitions of the questioning and so-called marking rules follow. Convergence of a class of procedures is then studied in terms of Markov chains (basic terminology is recalled). A brief report will be given on computer simulations of these procedures, and also on large-scale implementations that are under way. All are based on a standard test in elementary mathematics taken by 9th-grade students (about 15 years old) in New York City; answer data were obtained from the Office of Educational Assessment for 80,722 students. The five questions in Table 1.1 are typical of this study.

A second class of probabilistic procedures is also mentioned. All the material exposed in this chapter derives from recent investigations made by a team of researchers led by Jean-Claude Falmagne, formerly Professor of Psychology at New York University, and now at University of California, Irvine. It is a pleasure to acknowledge his innovative ideas, to mention his clear view of the topic developments, and to thank him for his constant enthusiasm. The bibliography contains twenty or more references to the originally published work, in particular Doignon and Falmagne (1985) and Falmagne and Doignon (1988b); a scientific survey is available (Falmagne, Koppen, Villano, Doignon, & Johannesen, 1990). The actual exposition is of a more pedagogical nature and does not cover all aspects of the work done so far. For instance, we leave aside the problem of how to build a knowledge structure in a particular domain (for this topic, see e. g. Dowling, 1991a, 1991b; Kambouri, Koppen, Villano, & Falmagne, 1991; Koppen, in press; or Koppen & Doignon, 1990).

1.2 Knowledge structures

The example of Table 1.1 will be intensively used. Some plain conventions are to be made when analyzing such a test. First, we assume that each question receives an answer which can be evaluated as correct or incorrect (thus a blank answer is taken as a wrong answer). Under this sole hypothesis, we may just assert the possibility of $2^5 = 32$ patterns of responses (each question generates two types of responses). Thus, any possible dependency among different items being ignored, each of the five questions has to be asked. Testing in this traditional view imposes a predetermined, large number of steps. It should be clear that in oral examination teachers strongly reduce the number of questions by making inferences from the collected answers, and also by specifically selecting the next question. These two features of the examination process are important; they are parts of the superior efficiency of oral testing over written testing. Any good automated procedure should encompass these features and exploit them to minimize the test duration.

Now what does lie at the heart of assessment procedures better than the all-questions approach? There must be some dependency among the notions tested by the responses, that is: from the mastering of one notion, the examiner can infer the mastering of other ones. In our example, a correct answer to question e reveals a good command of multiplication (be it by one-digit integers, or by two-digit integers). We thus assume that a student who correctly answers question e is also able to provide correct answers to questions a and c. Similarly, a correct answer to question d could lead the examiner to surmise (without asking the question) a correct answer to question b. Notice that these inferences for correct responses have logical counterparts for incorrect responses. Hence in the presence of an incorrect response to question b, we surmise that the answer to question d will also be incorrect.

The discussion in this Section ignores all the possible perturbations that we mentioned in the Introduction (Section 1.1). Here the answers are (temporarily) supposed to precisely reflect the constant state of knowledge of the examinee. We come, under this last assumption, to a combinatorial model in terms of (naive) set-theory. The mathematical concepts to be used are very simple ones (taught in secondary schools today), although historically they were clearly conceived only in the second part of the nineteenth century. Before giving the definitions, let us make clear that we adopt a purely descriptive approach to student evaluation. The notions tested by the questions are not meant to be in any way 'brain-located'. Since a testable theory is the final goal, all speculative concepts are disregarded. Instead, questions asked and answers collected are the central elements of the theory, and knowledge notions will be defined from these elements (regardless of other metaphysical concepts).

To start with, we consider a set of questions. In our example from Table 1.1, we define $Q = \{a, b, c, d, e\}$. Now, any student who took the test is characterized by the subset of questions he or she correctly answered. This subset constitutes his or her knowledge state. Here are some of the knowledge states corresponding to three (fictitious) students:

$$K_1 = \{a, b, c\}, \qquad K_2 = \{d\}, \qquad K_3 = \emptyset.$$

Thus, the first student gave correct responses only to questions a, b and c; the last student to no question at all (here $\emptyset$ denotes the empty set). All the 'possible' knowledge states form a collection of subsets of the set Q. For instance, by observing a population of students, we could come up with a collection $\mathcal{K}_1$ consisting of six knowledge states:

$$\mathcal{K}_1 = \{\emptyset, \{a\}, \{d\}, \{a, b, c\}, \{a, d, e\}, \{b, c, d, e\}, \{a, b, c, d, e\}\}. \tag{1.1}$$

Let us introduce some terminology.

Definition 1.2.1 A *knowledge structure* consists of a finite set Q together with a collection $\mathcal{K}$ of subsets of Q, where $\mathcal{K}$ contains at least the empty set $\emptyset$ and the whole set Q. The elements of Q are the *questions*, the members of $\mathcal{K}$ are the *knowledge states*. We write $(Q, \mathcal{K})$, or simply $\mathcal{K}$ if the set Q is implied by the context. We also say that $\mathcal{K}$ is a knowledge structure (on Q). □

Notice that not any subset of Q needs to be a knowledge state, i. e. a member of the given collection $\mathcal{K}$. Also, the concepts introduced in Definition 1.2.1 are purely descriptive and involve no deep theoretical assumption. Stronger features, such as an idea of dependency among questions, will appear in the next Section. The finiteness of the set Q of questions will be assumed throughout for simplicity (although it is not necessary for all of the definitions and results).

Looking more closely at the knowledge structure $\mathcal{K}_1$ in Equation (1.1), we see that questions b and c belong exactly to the same states, that is $\{a, b, c\}$ and $\{b, c, d, e\}$. On the contrary, questions b and e are distinguished at least by the knowledge state $\{a, b, c\}$, and thus surely test different skills. Being indiscernible with respect to the knowledge structure $\mathcal{K}_1$, questions b and c can be seen as bearing on the same notion, or, as we shall say, as *defining* the same notion.

Definition 1.2.2 Given the knowledge structure $(Q, \mathcal{K})$, define an equivalence relation on the set Q by setting for questions q and q' from Q:

q is *equivalent* to q' iff q and q' belong to exactly the same states.

Any equivalence class (with respect to that equivalence relation) is called a *notion* for $\mathcal{K}$. Moreover, $(Q, \mathcal{K})$ is *discriminative* when each notion consists of a single question. □

Example 1.2.1 For the structure $\mathcal{K}_1$ from Equation (1.1), the notions are

$$\{a\}, \quad \{b, c\}, \quad \{d\}, \quad \{e\}.$$

This structure is not discriminative. □

EXAMPLE 1.2.2 The knowledge structure $(Q, \mathcal{L})$ with $Q = \{a, b, c\}$ and $\mathcal{L} = \{\emptyset, \{a\}, \{a, b\}, \{a, c\}, \{a, b, c\}\}$ is discriminative because each of its notions consists of only one question. □

EXERCISE 1.2.1 List a few knowledge structures on the same five-element set. Give a formula for their total number. □

EXERCISE 1.2.2 What are the intended meanings of the following conditions in Definition 1.2.1?

(i) $\emptyset \in \mathcal{K}$; (ii) $Q \in \mathcal{K}$. □

EXERCISE 1.2.3 What is the least number of knowledge states in a discriminative knowledge structure on 5 questions? More difficult: when there are m questions, give the same least number as an expression in m. □

EXERCISE 1.2.4 Prove that the relation introduced in Definition 1.2.2 is indeed an equivalence relation on the set Q (that is, satisfies reflexivity, transitivity, and symmetry). □

FOR FURTHER REFLECTION 1.2.1 Sketch ways to discover theoretically or in practice 'the most adequate' knowledge structure associated to the test in Table 1.1. If you rely on answers from a population of students, grossly evaluate the number of students needed to collect 'reliable' data. □

1.3 Surmise relations

As announced in the preceding Section, we now refine the concept of a knowledge structure in order to tackle the idea of a dependency among the questions. A comparable structuring is sometimes made explicit by authors of university manuals, when they show the dependencies among the chapters in an illustrative diagram. An example of such a diagram is given in Figure 1.1; the readers see at a glance that chapter 8, say, has chapters 1, 2, 3, and 7 as prerequisites. A similar idea of a prerequisite has important consequences for our set of questions. For instance, returning to Table 1.1, elaborate multi-digit multiplication as in question e supposedly relies on elementary multiplication tested in question a. Consequently, from a correct answer to question e we should infer a correct answer to question a. We say that we surmise mastery of question a from mastery of question e. Formalizing the surmise idea leads to a (mathematical, binary) relation on the set of questions. Thus, the statements

mastery of questions a, b, c is surmised from mastery of question e,
mastery of question b is surmised from mastery of question d,
mastery of question c is surmised from mastery of question a.

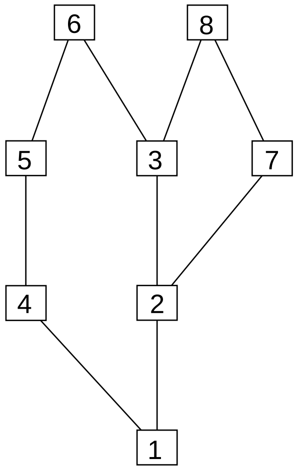


Figure 1.1. An example of dependencies among chapters in a book.

are altogether encoded in the following set S of pairs, where we shortly write ae for the (ordered) pair (a, e):

$$S = \{ae, be, ce, bd, ca\}.$$

Thus S is a relation on the set $Q = \{a, b, c, d, e\}$; we will also write qSq' when the pair qq' belongs to S. The relation S can be displayed in a diagram; see the right part of Figure 1.2 (the left part will be explained in a few lines).

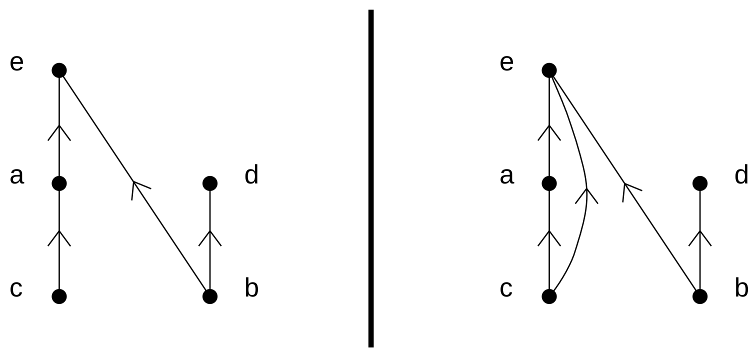


Figure 1.2. Two ways of listing prerequisite dependencies.

Some comments are in order here. Because of our presentation, the reader probably thinks of this relation as encoding logical dependencies among the tested notions. This is only one of the possible interpretations; another one relates to the fact that we could work with a definite population of students. The relation would then reflect the observed dependencies (due for instance to the past curriculum of the students).

From a more technical point of view, we see that there is some redundancy in the statements captured as above in the relation S. For instance, as pairs ca and ae encode prerequisite dependencies, then logically pair ce must also; this is

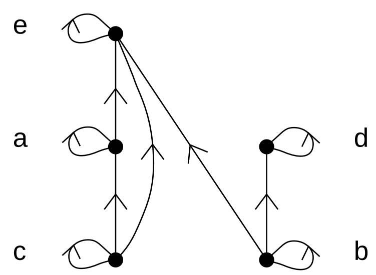


Figure 1.3. The surmise relation of Example 1.3.1.

an application of a transitivity property. (A relation S on the set Q is *transitive* when for q, q', q'' in Q, the following holds: qSq' and $q'Sq''$ imply qSq''.) Assuming transitivity, we may adopt two different positions when listing the surmise information: either we give a minimal information (in the example, we would not list the pair *ce* since it is derivable from other pairs), or we give the whole information (i. e. allowing for logical redundancy). The two cases are illustrated in the left and right parts of Figure 1.2, respectively. The general theory (as in Doignon & Falmagne, 1985) makes neither assumption. Here, we will take the second position in view of simplicity. The relation is assumed to be transitive, and also reflexive. (A relation S on the set Q is *reflexive* when qSq holds for each q in Q.) In mathematics, such a relation is often called a quasi order on Q.

Definition 1.3.1 A *surmise relation* on the (finite) set Q of questions is any transitive and reflexive relation on Q. If S denotes this relation, we also say that the pair (Q, S) is a surmise relation. □

Reflexivity of the surmise relation is assumed for convenience, without any practical motivation met insofar. It is required for later results (e. g. the one-to-one correspondence mentioned just after the statement of Theorem 1.4.1).

Example 1.3.1 Let $Q = \{a, b, c, d, e\}$ and $S = \{aa, bb, cc, dd, ee, ae, be, ce, bd, ca\}$. This surmise relation is depicted in Figure 1.3, which is very similar to the part to the right in Figure 1.2: we just added a loop at any node. □

Example 1.3.2 Let $Q = \{a, b, c, d\}$ and $S = \{aa, ab, ac, ad, bb, bc, bd, cc, cd, dd\}$. Here the surmise relation is a simple order (also called a total order, a complete order, or a linear order). □

Example 1.3.3 Let $Q = \{a, b, c, d\}$ and $S = \{aa, bb, cc, dd\}$. This is typically a situation in which no prerequisite dependency exists among the questions. □

Exercise 1.3.1 Draw figures representing the surmise relation in Example 1.3.2. Simplify the intricated drawing into a simpler one by deleting loops and all redundant pairs (as in the left part of Figure 1.2). □

1.4 Relationship between surmise relations and a class of knowledge structures

The surmise relation (Q, S) provides us with information on the possible knowledge states inside the set Q of questions (students are idealized in the sense that their answers are coherent with S). Consider Example 1.3.1: no knowledge state could contain question e while excluding question a. Otherwise, some student could master question e without mastering all of its prerequisites. As we thus see, the surmise relation S puts limitations on the possible knowledge states. It is natural then to associate to S the knowledge structure formed by all subsets of Q that represent the admissible knowledge states. In this way, our structure makes room for each of the possible students whose knowledge state does not conflict with the assumed prerequisite relation. To the relation S in Example 1.3.1, the following knowledge structure is associated:

$$\text{(1.2)} \qquad \begin{aligned} \mathcal{K}_2 \;=\; & \{\emptyset, \{b\}, \{c\}, \{b,c\}, \{a,c\}, \{b,d\}, \{a,b,c\}, \{b,c,d\}, \\ & \{a,b,c,d\}, \{a,b,c,e\}, Q\}. \end{aligned}$$

Notice that the subset $\{a, b\}$ is not a knowledge state, because the pair ca is in the relation S; thus a student who masters question a will also master question c. On the contrary, the subset $\{a, b, c\}$ is a knowledge state, because the prerequisites for questions a, b and c all belong to $\{a, b, c\}$.

Knowledge structures associated in this way to surmise relations share remarkable properties. For instance, if we take any two members of the collection $\mathcal{K}_2$ from Equation (1.3), say $\{a, b, c\}$ and $\{b, c, d\}$, we check that their intersection $\{b, c\}$ and their union $\{a, b, c, d\}$ also belong to the collection $\mathcal{K}_2$. These two properties of closure under intersection and union are always true for the knowledge states derived from a surmise relation. Moreover, in view of the following Theorem 1.4.1 on quasi orders due to Birkhoff (1937), these properties truly characterize the knowledge structures associated to surmise relations.

Definition 1.4.1 If S is a surmise relation on the (finite) set Q of questions, the *associated* knowledge structure has as knowledge states all subsets K of Q satisfying for questions q and q' from Q:

$$\text{if } qSq' \text{ and } q' \in K, \text{ then } q \in K. \qquad \square$$

Thus, to the surmise relation (Q, S), we associate through Definition 1.4.1 a knowledge structure $(Q, \mathcal{K})$. In fact, there are only two conditions to be checked in order to establish this assertion, namely $\emptyset \in \mathcal{K}$, and $Q \in \mathcal{K}$. The first condition holds because the empty set $\emptyset$ trivially satisfies Definition 1.4.1 of a knowledge state (remember that an implication is true if its antecedent is false). The second condition holds because the consequence of the implication in Definition 1.4.1 is always true when $K = Q$.

EXAMPLE 1.4.1 The knowledge structure associated to the surmise relation from Example 1.3.1 was obtained in Equation (1.3). □

EXAMPLE 1.4.2 For the surmise relation in Example 1.3.2, the associated knowledge structure is

$$\mathcal{K} = \{\emptyset, \{a\}, \{a, b\}, \{a, b, c\}, \{a, b, c, d\}\}.$$ □

EXAMPLE 1.4.3 For Example 1.3.3, each subset of Q is a knowledge state; thus, $\mathcal{K} = 2^Q$ (this is the notation used here for the collection of all subsets of Q). □

THEOREM 1.4.1 *Let $\mathcal{K}$ be the knowledge structure associated to a surmise relation on the set Q. Then any intersection and any union of knowledge states in $\mathcal{K}$ is also a knowledge state in $\mathcal{K}$. Conversely, any knowledge structure on Q which is closed under intersection and union is derived from some surmise relation on Q.* □

In fact, a one-to-one correspondence between surmise relations on Q and knowledge structures on Q, closed under intersection and union, will be established in the proof. We recall that Theorem 1.4.1, together with the existence of this correspondence, is due to Birkhoff (1937).

PROOF. Let (Q, S) be a surmise relation, and let $(Q, \mathcal{K})$ be the associated knowledge structure. Given two knowledge states K_1 and K_2 from $\mathcal{K}$, we show first that, according to Definition 1.4.1, $K_1 \cap K_2$ also belongs to $\mathcal{K}$. Assuming qSq' and $q' \in K_1 \cap K_2$ for some questions q and q' from Q, we have $q' \in K_1$ and $q' \in K_2$. Since $K_1, K_2 \in \mathcal{K}$, there follows $q \in K_1$ and $q \in K_2$ by repeated application of Definition 1.4.1, hence $q \in K_1 \cap K_2$. Thus, again by Definition 1.4.1, $K_1 \cap K_2 \in \mathcal{K}$. Now, for the union: if $K_1, K_2 \in \mathcal{K}$, and qSq', for questions q and q' with $q' \in K_1 \cup K_2$, we have $q' \in K_1$ or $q' \in K_2$. Hence, $q \in K_1$ or $q \in K_2$, thus $q \in K_1 \cup K_2$. This shows $K_1 \cup K_2 \in \mathcal{K}$.

We just proved that any knowledge structure associated to a surmise relation is closed under both intersection and union. To establish the converse, we consider a knowledge structure that is closed under intersection and union. Denoting it by $(Q, \mathcal{K}^*)$, we want to show that it is associated to some surmise relation on Q. Define a relation S on Q by

$$qSq' \iff (\text{for each } K \in \mathcal{K}^* :\ q' \in K \text{ implies } q \in K). \tag{1.3}$$

Then relation S is easily seen to be reflexive and transitive, in other words S is a surmise relation. Call $(Q, \mathcal{K})$ the knowledge structure associated to (Q, S). Clearly, any knowledge state K from the original structure $\mathcal{K}^*$ satisfies the condition in Definition 1.4.1, and thus belongs to structure $\mathcal{K}$. Conversely, we will prove that $K \in \mathcal{K}^*$ when $K \in \mathcal{K}$. First, consider for $q' \in K$ the following subset of Q:

$$A(q') = \{r \in Q \mid rSq'\}.$$

Equation (1.3) defining relation S says exactly that rSq' holds iff question r belongs to each element of $\mathcal{K}^*$ that contains question q'. Consequently, subset $A(q')$ is the intersection of the elements from $\mathcal{K}^*$ that contain question q. By the assumption of closure under intersection, subset $A(q')$ belongs itself to $\mathcal{K}^*$. Moreover, because of the reflexivity of S, subset $A(q')$ contains q', and by Definition 1.4.1, it lies entirely in K (because $q' \in K \in \mathcal{K}$). All this implies that K is the union of the various $A(q')$ for $q' \in K$, together with $A(q') \in \mathcal{K}^*$. Hence $K \in \mathcal{K}^*$. We have thus proved $\mathcal{K} = \mathcal{K}^*$, or: $(Q, \mathcal{K}^*)$ is the knowledge structure associated to relation S. This ends the proof of Theorem 1.4.1.

Notice that the first part of the proof builds, for a given surmise relation S on Q, a knowledge structure $(Q, k(S))$ closed under intersection and union. The second part of the proof builds a surmise relation $r(\mathcal{K}^*)$ on Q for a given knowledge structure $(Q, \mathcal{K}^*)$ closed under intersection and union, in such a way that $(Q, \mathcal{K}^*)$ is exactly $k(r(\mathcal{K}^*))$. Thus, the composite mapping $k \circ r$ is the identity mapping on the set of all knowledge structures on Q closed under intersection and union. One can also check that $r \circ k$ is the identity on the set of all surmise relations on Q. Thus the mappings k and r entail a one-to-one correspondence between surmise relations and knowledge structures closed under intersection and union. ■

Exercise 1.4.1 Consider several surmise relations on a four- or five-element set, and construct the associated knowledge structures. Decide whether the latter are discriminative. □

Exercise 1.4.2 Suppose $(Q, \mathcal{K})$ is the knowledge structure associated to the surmise relation S on Q. Give a necessary and sufficient condition on S in order that $(Q, \mathcal{K})$ be discriminative. □

Exercise 1.4.3 Consider the test with five questions described in Table 1.1. Propose a surmise relation on the set $\{a, b, c, d, e\}$. Construct the associated knowledge space. Is it discriminative? □

Exercise 1.4.4 Let $Q = \{a, b, c, d\}$ and $\mathcal{K} = \{\emptyset, \{a\}, \{c\}, \{a, c\}, \{c, d\}, \{a, b, c\}, \{a, c, d\}, Q\}$. Is the knowledge structure $(Q, \mathcal{K})$ associated to some surmise relation? Same exercise for $(Q, \mathcal{K}')$, with $\mathcal{K}' = \mathcal{K} \cup \{\{a, b\}, \{b, c\}, \{b, c, d\}\}$. When the answer is affirmative, find the surmise relation; is it unique? □

Exercise 1.4.5 Let S and S' be two surmise relations on the same set Q. If $S \subseteq S'$, does some inclusion hold between the two associated knowledge structures? □

Exercise 1.4.6 Let S be a surmise relation on the set Q of questions. Suppose some questions are deleted from Q, and denote by Q' the set of remaining questions. Define the *restriction* of S to Q'. Show how the knowledge structure associated to (Q', S') is derived from the one associated to (Q, S). □

For further reflection 1.4.1 Devise a test with a small number of questions in another area of knowledge, say history, car repair, art study, ... Build a corresponding surmise relation. Do some domains of knowledge make your work easier? Can you explain why? □

1.5 Surmise systems

What is for us the practical relevance of the two closure properties obtained in the previous Section? If two students, characterized by their knowledge states K and K', meet and share what they know, they will both end with the union $K \cup K'$ as their common knowledge state. This is a kind of motivation for assuming the closure of $\mathcal{K}$ under union. Even if the union $K \cup K'$ will not necessarily be observed in practice, we want to have room for it in our collection of knowledge states. On the other hand, there does not seem to be a similar motivation for the intersection. Some readers could argue that the two students would decide to retain only their common knowledge, that is $K \cap K'$. This very pessimistic point of view is usually not taken in knowledge theories: the cognitive development is rather assumed to be cumulative over time. Another argument for keeping only one of the two closure conditions comes from a mathematical result: closure under union leads to an interesting variant of Theorem 1.4.1. (By replacing knowledge states with their complements in the domain, a corresponding result would be obtained for families of subsets closed under intersection; see Exercise 1.6.7.) To state this variant (as Theorem 1.6.1), we need a generalization of surmise relations, that we motivate by a paramount example.

Suppose that a student shows great ability in the resolution of systems of linear equations. Can we deduce from this that he or she is confident in matrix inversion, in Gaussian transformations of systems (by pivoting), or in determinant computations? Of course, there is no reason to infer mastery of a particular one of these methods for solving systems: we can just assert that at least one of the techniques is mastered. In general terms, we are led to surmise from the mastery of one question the complete mastery of at least one set of questions among some list of sets. Calling these sets the clauses for the original question q, we say: for at least one clause for q, each question in this clause is mastered. Shortly writing "q" for "q is mastered", the assertion encodes in a formula of the form

$$\text{(1.4)} \qquad \text{if } q, \text{ then } (q_1 \text{ and } q_2) \text{ or } (q_3 \text{ and } q_4 \text{ and } q_5) \text{ or } (q_6 \text{ and } q_7).$$

In artificial intelligence, all such formulas under consideration are gathered in a so-called AND/OR graph (see e. g. Barr & Feigenbaum, 1981; or Rich, 1983). This approach requires the introduction of supplementary vertices for logical connectives. In order to avoid adding extra elements, we prefer to work with the related concept of a surmise system. We first look at examples, and then state the precise axioms and definitions.

Let us consider a set Q of questions and a mapping σ that associates to

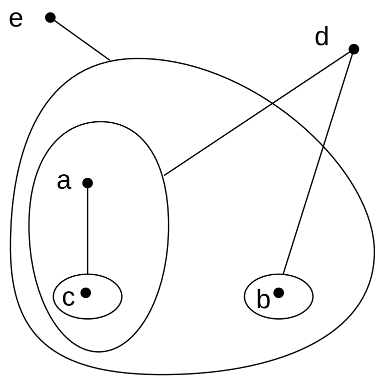


Figure 1.4. A pictorial display of the clauses in Examples 1.5.1 and 1.5.3.

any element q in Q a nonempty collection $\sigma(q)$ of subsets of Q. These subsets are the clauses for question q. In our approach, Equation (1.4) translates as

$$\sigma(q) = \{\{q_1, q_2\}, \{q_3, q_4, q_5\}, \{q_6, q_7\}\}.$$

EXAMPLE 1.5.1 Take $Q = \{a, b, c, d, e, f\}$ and set

$$\begin{aligned} \sigma(a) &= \{\{c\}\}, \\ \sigma(b) &= \{\emptyset\}, \\ \sigma(c) &= \{\emptyset\}, \\ \sigma(d) &= \{\{a, c\}, \{b\}\}, \\ \sigma(e) &= \{\{a, b, c\}\}. \end{aligned} \tag{1.5}$$

Here question a has only one clause, consisting of the single question c, while question b has the empty set as its only clause. In other words, there is only one way to master question a, through the acquisition of the single prerequisite c, while there is no prerequisite for question c. Question d, having two clauses, shows a more interesting case: it can be mastered along two different approaches, one implying the mastery of the sole question b, the other requiring the preliminary mastering of questions a and c. All the information contained in σ is captured in Figure 1.4, with obvious conventions. □

The surmise relations that we introduced in Definition 1.3.1 can also be cast in this way. More generally, to any relation S on Q, we associate the mapping σ on the set Q with $\sigma(q)$ consisting of the single clause $\{q' \in Q \mid q' S q\}$. That is, we collect in a single clause all the 'prerequisites' for q.

EXAMPLE 1.5.2 For the relations from the left part of Figure 1.2, and from Figure 1.3, we obtain:

$$\begin{aligned} \sigma_1(a) &= \{\{c\}\}, & \sigma_2(a) &= \{\{a, c\}\}, \\ \sigma_1(b) &= \{\emptyset\}, & \sigma_2(b) &= \{\{b\}\}, \\ \sigma_1(c) &= \{\emptyset\}, & \sigma_2(c) &= \{\{c\}\}, \\ \sigma_1(d) &= \{\{b\}\}, & \sigma_2(d) &= \{\{d, b\}\}, \\ \sigma_1(e) &= \{\{a, b\}\}, & \sigma_2(e) &= \{\{e, a, b, c\}\}. \end{aligned}$$

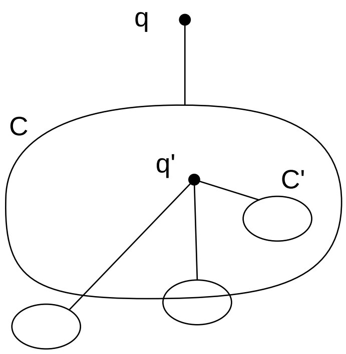


Figure 1.5. Illustration of Axiom 2 for surmise systems.

□

The two formal descriptions in Example 1.5.2 originate from the same practical example, but in the left case we looked for information saving, in the right case we imposed transitivity and reflexivity. Recall that only the column to the right corresponds to a surmise relation in the sense of Definition 1.3.1. As we set forth mathematical postulates for surmise relations, we will now formulate three axioms for a mapping σ. In the three statements, q stands for any question in Q. Axiom 2 is illustrated in Figure 1.5.

Axiom 1 Any clause for question q contains q.

Axiom 2 If $q' \in C$, with C a clause for question q, there exists some clause C' for q' satisfying $C' \subseteq C$.

Axiom 3 Any two clauses for question q are incomparable (i. e., neither is included in the other).

The three axioms are satisfied by the mapping σ_2 in Example 1.5.2. We leave the verification to the reader. On the contrary, Axioms 1 and 2 are *not* satisfied by σ_1. For instance, question a belongs to the clause $\{a, b\}$ for question e, but the only clause $\{c\}$ for a is not included in $\{a, b\}$; hence Axiom 2 is not satisfied for $q = e$.

Example 1.5.1 does *not* satisfy Axiom 1. In fact, we would rather encode as follows the information it conveys.

Example 1.5.3 Take $Q = \{a, b, c, d, e, f\}$ and set

$$\begin{aligned} \sigma_3(a) &= \{\{a, c\}\}, \\ \sigma_3(b) &= \{\{b\}\}, \\ \sigma_3(c) &= \{\{c\}\}, \\ \sigma_3(d) &= \{\{d, a, c\}, \{d, b\}\}, \\ \sigma_3(e) &= \{\{e, a, b, c\}\}. \end{aligned}$$

Axioms 1, 2, 3 are satisfied by this mapping σ_3. □

Axiom 1 plainly generalizes the reflexivity condition for a relation, while the second axiom extends the notion of transitivity. On the other hand, the third axiom is always fulfilled by relations since each question has then a unique prerequisite.

DEFINITION 1.5.1 A *surmise system* on the (finite) set Q is a mapping σ that associates to any element q in Q a nonempty collection $\sigma(q)$ of subsets of Q, and that satisfies Axioms 1, 2, and 3. The elements of the set Q will again be called questions. The subsets in $\sigma(q)$ are the *clauses* for question q. □

EXERCISE 1.5.1 The notation $\{\emptyset\}$ could be disturbing to some readers. Do you clearly see the differences among $\{\emptyset\}$, $\{0\}$ and $\emptyset$? □

1.6 Surmise systems and knowledge spaces

In the previous Section, we have defined surmise systems as a generalization of surmise relations. Here we define the knowledge states of a surmise system. Our aim is to state a variant of Birkhoff Theorem 1.4.1 (as Theorem 1.6.1).

DEFINITION 1.6.1 Let (Q, σ) be a surmise system. The *knowledge states* of (Q, σ) are all the subsets K of Q that satisfy:

$$\text{if } q \in K, \text{ there exists a clause } C \in \sigma(q) \text{ such that } C \subseteq K.$$

They constitute the knowledge structure *associated* to (Q, σ). Any knowledge structure which is closed under union is called a *knowledge space.* □

The reader should by himself or herself uncover the motivation for this definition of the knowledge states of the surmise system (Q, σ), and also check that the family of all states is indeed a knowledge structure. The knowledge states of Definition 1.6.1 can also be characterized as being a union of some family of clauses (see Exercise 1.6.3); proving this will help to understand the rôle of Axioms 1, 2, and 3 considered in Definition 1.5.1. Notice that the present definition of knowledge states, when applied to a surmise relation (Q, S) (cast into a surmise system by letting $\sigma(q) = \{\{q' \mid q'Sq\}\}$), coincides with the previous Definition 1.4.1. Here is another example.

EXAMPLE 1.6.1 Let $Q = \{a, b, c, d, e\}$ and σ_3 be given as in Example 1.5.3 or Figure 1.4:

$$\begin{aligned}
\sigma_3(a) &= \{\{a, c\}\}, \\
\sigma_3(b) &= \{\{b\}\}, \\
\sigma_3(c) &= \{\{c\}\}, \\
\sigma_3(d) &= \{\{a, c, d\}, \{b, d\}\}, \\
\sigma_3(e) &= \{\{a, b, c, e\}\}.
\end{aligned}$$

This is a surmise system. Its associated knowledge structure is

$$\begin{aligned}\mathcal{K}_3 \;=\; & \{\emptyset, \{b\}, \{c\}, \{a,c\}, \{b,c\}, \{b,d\}, \{a,b,c\}, \{a,c,d\}, \\ & \{b,c,d\}, \{a,b,c,d\}, \{a,b,c,e\}, Q\}.\end{aligned}$$

The collection $\mathcal{K}_3$ is displayed in Figure 1.6, using plain conventions for showing set-inclusion. The structure $\mathcal{K}_3$ is closed under union, in other words $\mathcal{K}_3$ is a knowledge space. Nevertheless, this space is not closed under intersection, since for instance the knowledge states $\{b,d\}$ and $\{a,c,d\}$ have $\{d\}$ as their intersection, a subset which is not a state. □

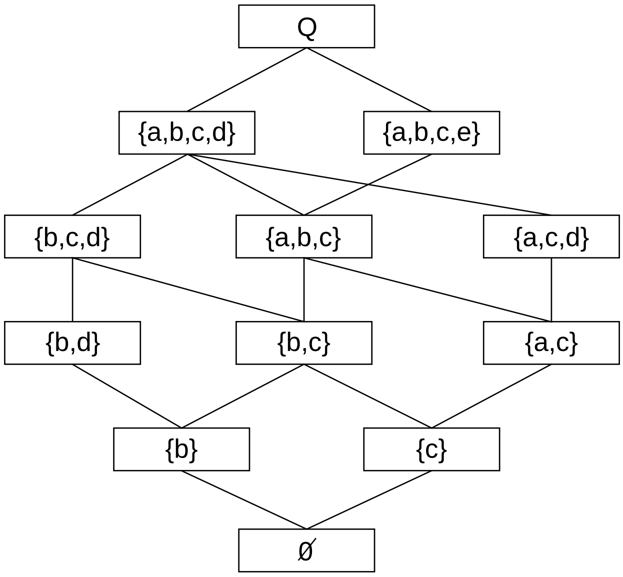


Figure 1.6. The knowledge structure $\mathcal{K}_3$ from Example 1.6.1.

We are now in a position to state and prove the promised variant of Birkhoff Theorem 1.4.1.

THEOREM 1.6.1 *The knowledge structure associated to a surmise system is closed under union (but not necessarily under intersection), i. e. it is a knowledge space. Moreover, any knowledge space is associated to some surmise system.* □

In fact, there is a one-to-one correspondence between surmise systems on Q and knowledge spaces on Q, as will be shown in the following proof (originally, a stronger result was established in Doignon & Falmagne, 1985).

PROOF. If (Q, σ) is a surmise system and $(Q, \mathcal{K})$ its associated knowledge structure, pick $K_1, K_2 \in \mathcal{K}$. We show that $K_1 \cup K_2 \in \mathcal{K}$ by referring to Definition 1.6.1. If $q \in K_1 \cup K_2$, one has $q \in K_1$ or $q \in K_2$. Thus some clause C for q is included in K_1 or K_2, hence is included in $K_1 \cup K_2$. This establishes the first assertion (with Example 1.6.1 showing that closure under intersection can fail to hold). Now, consider a knowledge space $(Q, \mathcal{K}^*)$. Since Q is finite and $Q \in \mathcal{K}^*$, there is for each question q at least one minimal knowledge state containing q (a knowledge state K is a minimal knowledge state containing q if $q \in K$, and there is no knowledge state K' satisfying $q \in K' \subset K$). We define σ by collecting in $\sigma(q)$ all the minimal knowledge states containing q. Then Axioms 1 and 3 clearly hold for σ. Axiom 2 is also obvious: if $q' \in C \in \sigma(q)$, one can repeatedly delete questions from the knowledge state C until there remains a knowledge state containing q' that is minimal: there results an element C' of $\sigma(q')$ with $C' \subseteq C$. Thus (Q, σ) is a surmise system. It remains only to establish the equality between $\mathcal{K}^*$ and the knowledge structure $\mathcal{K}$ associated to σ. First, any K from $\mathcal{K}^*$ will satisfy Definition 1.6.1 (because if $q \in K$, there is a minimal C for which $q \in C \in \mathcal{K}^*$ and $C \subseteq K$, and this C belongs to $\sigma(q)$). Thus $\mathcal{K}^* \subseteq \mathcal{K}$. Conversely, if $K \in \mathcal{K}$, notice that K is a union of clauses, one in each $\sigma(q)$ for $q \in K$. Since these clauses are by construction elements from $\mathcal{K}^*$, and $\mathcal{K}^*$ is assumed to be closed under union, it follows that K belongs also to $\mathcal{K}^*$. Similarly as in the proof given of Theorem 1.4.1, we can derive a one-to-one correspondence (here, between surmise systems on Q and knowledge spaces on Q). It can be checked that the actual one-to-one correspondence extends the previous one (meaning: surmise relations cast as surmise systems correspond to knowledge spaces that are closed under intersection). ∎

Let us repeat a few of our findings. To a surmise relation, we associated a knowledge structure which is closed under both intersection and union (see Definition 1.4.1 and Theorem 1.4.1). On the contrary, the knowledge structure associated to a surmise system is always closed under union, but not not necessarily under intersection; it is a knowledge space (see Definition 1.6.1 and Theorem 1.6.1). We started the previous Section with a discussion on the relevance of the closure conditions. At this point, we clearly see that closure under union is a 'good' axiom to impose on knowledge states. It exactly characterizes families of states derived from surmise systems. In other words, a knowledge space is one of the possible formalizations of a system of multiple prerequisites. Being conceptually very simple, it is the model on which we will base assessment procedures.

Exercise 1.6.1 Consider the following five questions (cf. Doignon & Falmagne, 1985).

i. Let p be the probability of drawing a red ball in some urn. What is the probability of observing at least one red ball in a random sample of n balls, if the sampling is done with replacement?

ii. What is the probability of the joint realization of n independent events, each of which has a probability equal to p?

iii. Give the formula for the binomial coefficient $\binom{n}{k}$. Perform the computation for $n = 7$ and $k = 5$.

iv. In the experiment of Question i, what is the probability of observing exactly k red balls? Give the formula, and perform the computation for $n = 5$ and $k = 3$.

v. Let $\Pr(A)$ be the probability of an event A in a probability space. What is the probability that A is not realized in one trial?

Propose a surmise relation on the set $\{a, b, c, d, e\}$. Propose a surmise system on the set $\{a, b, c, d, e\}$, with at least one question having more than one clause (an example is documented in Doignon & Falmagne, 1985). Construct the associated knowledge spaces. Are they discriminative? □

Exercise 1.6.2 For any (finite) set Q, describe the two surmise systems to which are associated the 'extreme' knowledge spaces $(Q, \{\emptyset, Q\})$ and $(Q, 2^Q)$. Do you find surmise relations; in other words, is there exactly one clause for each question q? □

Exercise 1.6.3 Show that the knowledge states of Definition 1.6.1 can be characterized as being all the unions of clauses. □

Exercise 1.6.4 Suppose $(Q, \mathcal{K})$ is the knowledge space associated to the surmise system (Q, σ). Give a necessary and sufficient condition in terms of clauses of σ in order that $(Q, \mathcal{K})$ be discriminative. □

Exercise 1.6.5 Define the restrictions of a surmise system (Q, σ) and of a knowledge space $(Q, \mathcal{K})$, respectively, to a subset Q' of Q. Is the restriction of (Q, σ) again a surmise system, this time on Q'? Is the restriction of $(Q, \mathcal{K})$ again a knowledge space, this time on Q'? Check whether restriction behaves well with respect to the one-to-one correspondence from the proof of Theorem 1.6.1. □

Exercise 1.6.6 (For new terminology in this Exercise, see Chapter 4 in this book.) Let $(Q, \mathcal{K})$ be a knowledge structure. Is the partially ordered set $(\mathcal{K}, \subseteq)$ a lattice? Same question for a knowledge space $(Q, \mathcal{K})$. Describe meet and join in case you obtain a lattice, and interpret these operations in the knowledge-representation context. □

Exercise 1.6.7 Let $(Q, \mathcal{K})$ be a knowledge structure. Define the *complement structure* $(Q, \mathcal{K}^c)$ by $K \in \mathcal{K}^c \iff Q \setminus K \in \mathcal{K}$. Show that this induces a one-to-one correspondence between the set of knowledge structures closed under intersection, and the set of knowledge structures closed under union. □

For further reflection 1.6.1 How would you economically store a huge knowledge space into a computer? (this question relates to the data that need to be stored rather than to the computer internal organization). □

For further reflection 1.6.2 In Exercise 1.6.1, it is most probable that many distinct surmise relations/systems will be proposed, and thus also various knowledge spaces. How could one compare two of the proposed systems? and two of the built spaces? (for instance: how to measure their dissimilarity?) These problems appeared in the analysis of a (real-life) mathematical test having 50 questions. A related question: is there a sound method for aggregating a family of surmise systems on the set Q (or respectively, knowledge spaces on Q) into one surmise system (resp. knowledge space)? (see e. g. Kambouri, Koppen, Villano, & Falmagne, 1991; or Villano, 1991). □

1.7 Well-graded knowledge spaces

We still have to introduce one more combinatorial condition on knowledge spaces that will prove useful for the design of assessment procedures. Let us compare the following two knowledge spaces on $Q = \{a, b, c, d, e\}$ (the first one was described in Example 1.6.1 and pictured in Figure 1.6):

$$\begin{aligned}\mathcal{K}_3 &= \{\emptyset, \{b\}, \{c\}, \{a,c\}, \{b,c\}, \{b,d\}, \{a,b,c\}, \{a,c,d\}, \\ &\quad \{b,c,d\}, \{a,b,c,d\}, \{a,b,c,e\}, Q\}, \\ \mathcal{L} &= \{\emptyset, \{a,b\}, \{b,c,d\}, \{a,c,d\}, \{a,d,e\}, \{a,b,c,d\}, \\ &\quad \{a,b,d,e\}, \{a,c,d,e\}, \{b,c,d,e\}, Q\}.\end{aligned}$$

Suppose we want to trace the learning history of some student who starts from mastering nothing in the set $Q = \{a, b, c, d, e\}$ and ends with mastering each of the five questions. The successive steps will correspond to knowledge states, each of which is obtained from the previous one by adding one single question (the question that was learned at this step). For instance, two such possible sequences in the context of the knowledge space $\mathcal{K}_3$ are

$$\emptyset,\ \{b\},\ \{b,c\},\ \{b,c,d\},\ \{a,b,c,d\},\ Q,$$
$$\emptyset,\ \{c\},\ \{a,c\},\ \{a,b,c\},\ \{a,b,c,d\},\ Q.$$

On the contrary, we cannot find in the knowledge space $\mathcal{L}$ a sequence of knowledge states from $\emptyset$ to Q that grows by one question at a time (simply because there is no knowledge state formed by a single question). We will be interested in knowledge spaces in which such sequences exist from $\emptyset$ to the whole set of questions, and (more demanding) from any state K_1 to any state K_2 satisfying $K_1 \subset K_2$. As will be explained later in this Section (see after Example 1.7.5), these knowledge spaces have mathematical properties of interest for the design of assessment procedures.

Definition 1.7.1 A knowledge space is said to be *well graded* when for any two of its states, say K and K', with $K \subset K'$, there is a sequence of knowledge states $K_0 = K \subset K_1 \subset K_2 \subset \ldots \subset K_n = K'$, where K_{i+1} for $i = 0, 1, \ldots, n-1$ is obtained from K_i by adding one single question. □

Well-gradedness can also be characterized in terms of the surmise system (see Koppen, 1991); only a slight variant of Axiom 2 from Section 1.5 is required. We leave the proof of Proposition 1.7.1 as an exercise, as well as that of the subsequent Corollary 1.7.1 (see Exercises 1.7.2 and 1.7.3).

PROPOSITION 1.7.1 *Let (Q, σ) be a surmise system and let $(Q, \mathcal{K})$ be its associated knowledge space. Then $(Q, \mathcal{K})$ is well graded iff the following Condition holds for each question q in Q and each clause C in $\sigma(q)$:*

$$\text{if } q' \in C \setminus \{q\}, \text{ there exists } C' \in \sigma(q') \text{ with } C' \subseteq C \setminus \{q\}. \qquad \square$$

EXAMPLE 1.7.1 The knowledge space on $Q = \{a, b, c, d, e\}$ defined by

$$\begin{aligned}\mathcal{K}_3 \;=\; \{&\emptyset, \{b\}, \{c\}, \{a, c\}, \{b, c\}, \{b, d\}, \{a, b, c\}, \{a, c, d\},\\ &\{b, c, d\}, \{a, b, c, d\}, \{a, b, c, e\}, Q\}\end{aligned}$$

is well graded. By Example 1.6.1, we know that it is associated to the surmise system (Q, σ_3) with

$$\begin{aligned}\sigma_3(a) &= \{\{a, c\}\},\\ \sigma_3(b) &= \{\{b\}\},\\ \sigma_3(c) &= \{\{c\}\},\\ \sigma_3(d) &= \{\{a, c, d\}, \{b, d\}\},\\ \sigma_3(e) &= \{\{a, b, c, e\}\}.\end{aligned}$$

One easily checks that for any clause C for any question q, the set $C \setminus \{q\}$ belongs to $\mathcal{K}_3$. This is easily seen to be equivalent to the condition in Proposition 1.7.1. $\square$

COROLLARY 1.7.1 *The knowledge space associated to a surmise relation is well graded iff it is discriminative.* $\square$

In the following example, we have a surmise relation whose associated knowledge space is not well graded (and of course not discriminative).

EXAMPLE 1.7.2 Let $Q = \{a, b, c\}$ and $S = \{aa, bb, cc, ab, ba\}$. Then the associated knowledge space $(Q, \mathcal{L})$ has $\mathcal{L} = \{\emptyset, \{c\}, \{a, b\}, Q\}$. It is not discriminative, and a fortiori not well graded. $\square$

The cardinality (number of elements) of the set A will be denoted by $|A|$. Thus, in Definition 1.7.1 above, we have $|K_{i+1} \setminus K_i| = 1$. Moreover, $A \triangle B$ denotes the *symmetric difference* of the sets A and B, that is $A \triangle B = (A \setminus B) \cup (B \setminus A)$. The mapping that associates to any two subsets A and B of Q the number $|A \triangle B|$ is a distance on the set 2^Q of all subsets of Q. (There is a precise mathematical definition of 'distance', that we will not need in the sequel). In particular the distance $d(K, K') = |K \triangle K'|$ between two states K and K' in a knowledge structure $(Q, \mathcal{K})$ has now a precise meaning. We define the *(closed) ball* around a knowledge state, given some real number ϵ:

$$B(K, \epsilon) = \{K' \in \mathcal{K} \mid d(K, K') \leq \epsilon\}$$

(thus this ball contains all knowledge states which differ by at most ϵ questions from the given knowledge state K; notice the similarity with the classical ball in Euclidean space).

Example 1.7.3 Take the same knowledge space as in Example 1.7.1. Table 1.2 gives all distances between any two of the knowledge states. From it,

Table 1.2. The distances among the knowledge states of Example 1.7.1.

	$\emptyset$	b	c	ac	bc	bd	abc	acd	bcd	$abcd$	$abce$	Q
$\emptyset$	0	1	1	2	2	2	3	3	3	4	4	5
b	1	0	2	3	1	1	2	4	2	3	3	4
c	1	2	0	1	1	3	2	2	2	3	3	4
ac	2	3	1	0	2	4	1	1	3	2	2	3
bc	2	1	1	2	0	2	1	3	1	2	2	3
bd	2	1	3	4	2	0	3	3	1	2	4	3
abc	3	2	2	1	1	3	0	2	2	1	1	3
acd	3	4	2	1	3	3	2	0	2	1	3	2
bcd	3	2	2	3	1	1	2	2	0	1	3	2
$abcd$	4	3	3	2	2	2	1	1	1	0	2	1
$abce$	4	3	3	2	2	4	1	3	3	2	0	1
Q	5	4	4	3	3	3	2	2	2	1	1	0

we can quickly find balls, for instance

$$\begin{aligned}
B(\{a,c\},1) &= \{\{a,c\},\{c\},\{a,b,c\},\{a,c,d\}\};\\
B(\{b,c\},1) &= \{\{b,c\},\{b\},\{c\},\{a,b,c\},\{b,c,d\}\};\\
B(\{a,c\},2) &= \{\{a,c\},\{c\},\{a,b,c\},\{a,c,d\},\emptyset,\{b,c\},\\
&\qquad \{a,b,c,d\},\{a,b,c,e\}\};\\
B(\{b,c\},2) &= \{\{b,c\},\{b\},\{c\},\{a,b,c\},\{b,c,d\},\\
&\qquad \emptyset,\{a,c\},\{b,d\},\{a,b,c,d\},\{a,b,c,e\}\}.
\end{aligned}$$ □

Let us now look more closely at the case $\epsilon = 1$. The knowledge states in any ball $B(K,1)$ are obtained either by deleting or by adding exactly one question to K. Of course, not any question in Q will work, that is produce from K a new knowledge state.

Definition 1.7.2 The *fringe* $F(K)$ of a knowledge state K in a knowledge space $(Q,\mathcal{K})$ consists of all questions q such that

$$q \in K \text{ and } K \setminus \{q\} \in \mathcal{K}, \text{ or } q \notin K \text{ and } K \cup \{q\} \in \mathcal{K}$$ □

Example 1.7.4 For the same knowledge space $\mathcal{K}_3$ as in Example 1.7.3 or Table 1.2, we obtain $F(\{b\}) = \{b,c,d\}$ and $F(\{b,c\}) = \{a,b,c,d\}$. □

Proposition 1.7.2 *Each of the following two conditions on a knowledge space* $(Q, \mathcal{K})$ *is equivalent to the well-gradedness of* $(Q, \mathcal{K})$*:*

(i) for any two states K *and* K'*, with* $K \neq K'$*,*
$$(K \triangle K') \cap F(K) \neq \emptyset;$$

(ii) for any two distinct states K *and* K' *there exists a sequence* $K_0 = K, K_1, K_2, \ldots, K_n = K'$ *of knowledge states such that for* $j = 0, 1, 2, \ldots, n-1$*:*
$$d(K_j, K_{j+1}) = 1 \quad \text{and} \quad d(K, K') > d(K_{j+1}, K'). \qquad \square$$

Proof. a) We show that (i) follows from well-gradedness. Assume $K' \not\subseteq K$ (if this is not the case, then $K \not\subseteq K'$, and a similar argument can be given). As $K \cup K'$ is a knowledge state from $\mathcal{K}$, there must exist $K_1 \in \mathcal{K}$ with $K \subset K_1 \subseteq K \cup K'$ and $K_1 \setminus K$ consisting of one single question q. We have $q \in (K \triangle K') \cap F(K)$.
b) Assume the knowledge space $(Q, \mathcal{K})$ satisfies (i) and let us prove it satisfies (ii). Given K, K' as in Condition (ii), we use (i) for obtaining a question q in $(K \triangle K') \cap F(K)$. If $q \in K \setminus K'$, we set $K_1 = K \setminus \{q\}$; if $q \in K' \setminus K$, we set $K_1 = K \cup \{q\}$. In both cases, $K_1 \in \mathcal{K}$, $d(K, K_1) = 1$, and also $d(K, K') > d(K_1, K')$. Hence we may use K_1 as the first knowledge state to be constructed in order to establish (ii). The second one is constructed by applying the same argument to K_1 and K'. Repeating this process will lead to a sequence that eventually ends in K'.
c) We show that a knowledge space $(Q, \mathcal{K})$ satisfying (ii) necessarily is well graded. Given $K, K' \in \mathcal{K}$ with $K \subset K'$, we consider a sequence as in (ii). Then $K_1 \triangle K$ consists of one single question q. Since $q \in K$ or $q \notin K'$ contradicts $d(K, K') > d(K_1, K')$, we have $K \subset K_1 \subseteq K'$. Repeating the same argument with K_1 and K', etc., we are able to construct the required sequence of knowledge states from K to K'. ■

Example 1.7.5 Consider again the same knowledge space $\mathcal{K}_3$ as in Examples 1.7.1, 1.7.3, 1.7.4, or Table 1.2. From Example 1.7.1, we know that $\mathcal{K}_3$ is well graded. Thus Conditions (i) and (ii) do hold. For instance, setting $K = \{b\}$, each other state K' meets $F(\{b\}) = \{b, c, d\}$. Taking $K' = Q$, we find a sequence as in condition (ii):
$$\{b\} \subset \{b, c\} \subset \{a, b, c\} \subset \{a, b, c, d\} \subset Q. \qquad \square$$

Well-gradedness will be a useful assumption on knowledge spaces when we will design assessment procedures. By Proposition 1.7.2(ii), we see that we can transform any knowledge state into another one by a succession of elementary changes (consisting each in the addition or deletion of one question), while always keeping a knowledge state. Because of this nice feature, we may design assessment procedures that explore the family $\mathcal{K}$ of states by making at each step such an elementary change in the actual 'approximate' or 'candidate' state. These procedures will never be caught in a subfamily of $\mathcal{K}$ that does not contain the knowledge state to be uncovered. Another interesting aspect of well-gradedness is relevant for the design of instruction paths: each knowledge state can be reached from the empty set by successive acquisitions of one question at a time.

Exercise 1.7.1 Are conditions (i) and (ii) in Proposition 1.7.2 equivalent to the following variant of (ii)?

(iii) for any two distinct states K and K' there exists a sequence $K_0 = K, K_1, K_2, \ldots, K_n = K'$ of knowledge states such that for $j = 0, 1, \ldots, n-1$:

$$d(K_j, K_{j+1}) = 1.$$

□

Exercise 1.7.2 Prove Proposition 1.7.1 and its Corollary 1.7.1. □

Exercise 1.7.3 Give another, direct proof of Corollary 1.7.1. □

Exercise 1.7.4 Is a well-graded knowledge space necessarily discriminative? and conversely? □

Exercise 1.7.5 Let $(Q, \mathcal{K})$ be a well-graded knowledge space. If one deletes (respectively, adds) knowledge states in order to get another knowledge space $(Q, \mathcal{K}')$, is $(Q, \mathcal{K}')$ necessarily well graded? □

Exercise 1.7.6 Let $(Q, \mathcal{K})$ be a well-graded knowledge space. Assume some of the questions are deleted, and Q' is the remaining set of questions. Is the restriction (see Exercise 1.6.5) of $\mathcal{K}$ to Q' a well-graded knowledge space? □

Exercise 1.7.7 Let (Q, σ) be a surmise system, and $(Q, \mathcal{K})$ be its associated knowledge space. Is the following condition equivalent to well-gradedness of $(Q, \mathcal{K})$? For all distinct questions q, $r \in Q$, the intersection $\sigma(q) \cap \sigma(r)$ is empty (i. e., no two distinct questions have a common clause). □

Exercise 1.7.8 Same as Exercise 1.7.6, with the following condition (cf. Example 1.7.1). For each question q, and each clause C for q, the set $C \setminus \{q\}$ is a knowledge state. □

For further reflection 1.7.1 Suppose that practical investigation has lead to some knowledge space $(Q, \mathcal{K})$ that is not well graded. Is there a 'natural' way to add or delete knowledge states in order to obtain well-gradedness? This is a presently unsolved question (a bit imprecise one also). Notice that if $(Q, \mathcal{K}_1)$ and $(Q, \mathcal{K}_2)$ are well-graded knowledge spaces, then $(Q, \mathcal{K}_1 \cap \mathcal{K}_2)$ is a knowledge space, but not necessarily a well-graded one (can you give a counter-example?). □

1.8 Deterministic assessment procedures

Let us reformulate our central problem. We have at hand a precise description of the knowledge space $(Q, \mathcal{K})$. Each student in a certain population masters some of the questions in Q; it is assumed that the questions mastered always constitute a knowledge state belonging to $\mathcal{K}$. How can we efficiently uncover, given a student in the population, which member of $\mathcal{K}$ represents his or her knowledge state?

In a preliminary approach to the assessment problem, we make the following simplifying assumptions (ignored in the later Sections). First, the knowledge state of the student is taken as constant along the testing procedure. Second, any of the responses will plainly reflect this knowledge state (that is, we temporarily rule out any careless error or lucky guess). Thus avoiding any randomness in the problem, we are to describe deterministic procedures for the assessment of knowledge. As discussed in the Introduction (Section 1.1), we want to avoid the all-question testing. Having at hand the collection $\mathcal{K}$, we devise a better strategy for the assessment procedure. A list of knowledge states is maintained; at each step, it retains the knowledge states compatible with the information yet collected.

Here is an illustration based on the knowledge space $(Q, \mathcal{K}_3)$ from Example 1.7.1 (see also Figure 1.6), that is $Q = \{a, b, c, d, e\}$ and

$$\begin{aligned}\mathcal{K}_3 \;=\; &\{\emptyset, \{b\}, \{c\}, \{a, c\}, \{b, c\}, \{b, d\}, \{a, b, c\}, \{a, c, d\}\\ &\{b, c, d\}, \{a, b, c, d\}, \{a, b, c, e\}, Q\}.\end{aligned}$$

Suppose that we first ask question a and obtain a correct response. Hence the states $\emptyset, \{b\}, \{c\}, \{b, c\}, \{b, d\}, \{b, c, d\}$ must be discarded because they do not contain question a. We are left with the knowledge states in $\mathcal{K}_3$ containing a. Their collection will be denoted as $(\mathcal{K}_3)_a$:

$$(\mathcal{K}_3)_a = \{\{a, c\}, \{a, b, c\}, \{a, c, d\}, \{a, b, c, d\}, \{a, b, c, e\}, Q\}.$$

NOTATION 1.8.1 For a knowledge structure $(Q, \mathcal{K})$, and a question q from Q, we set

$$(\mathcal{K})_q = \{K \in \mathcal{K} \mid q \in K\},$$

and

$$(\mathcal{K})_{\bar{q}} = \{K \in \mathcal{K} \mid q \notin K\}. \qquad \square$$

Let us turn back to our illustration. If we ask question e after having asked question a and then collect a wrong answer, there will remain as possible states

$$(\mathcal{K}_3)_{a\bar{e}} = \{\{a, c\}, \{a, b, c\}, \{a, c, d\}, \{a, b, c, d\}\}.$$

A correct answer to question b would then rule out the first and third of these states. Finally, asking question d will discriminate between the remaining two states. A correct answer, say, will leave

$$(\mathcal{K}_3)_{a\bar{e}bd} = \{\{a, b, c, d\}\}.$$

Notice that we asked only four out of the five questions in Q.

The global procedure can be displayed in a diagram that explains itself (see Figure 1.7) if read from left to right. A path from the node most on the left to the external node showing a knowledge state captures one realization of the deterministic assessment procedure. We see that the lengths of all these paths are on the average shorter than 5. This is our gain, resulting from the assumed organization of the material (in essence: not every subset of Q is a member of $\mathcal{K}$).

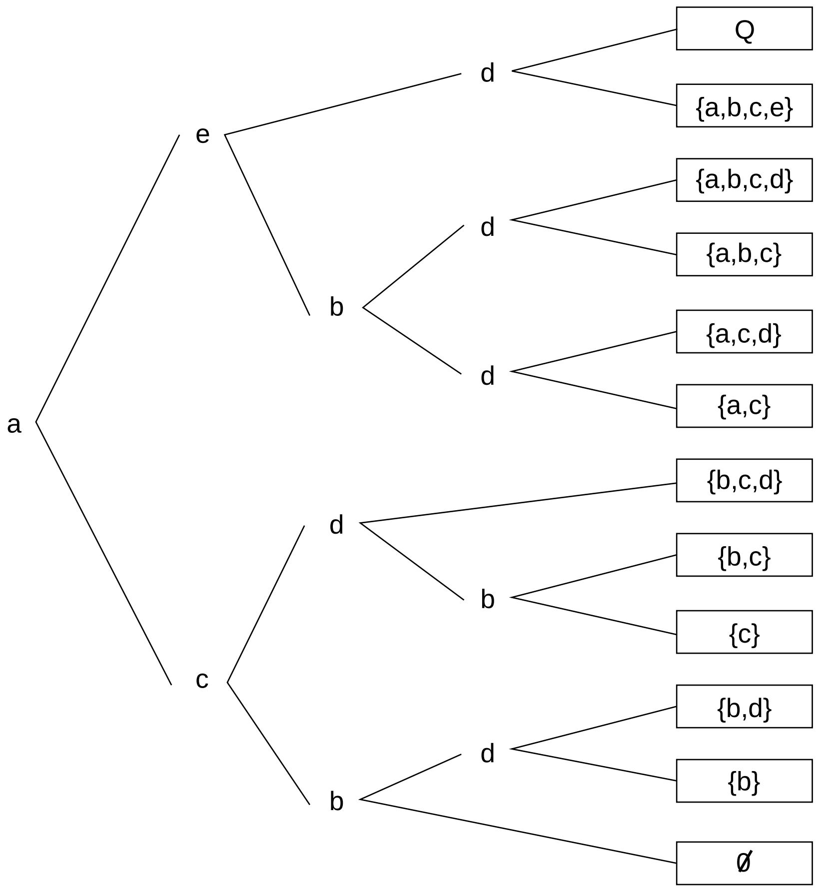


Figure 1.7. A diagram for a deterministic assessment procedure for the knowledge space $\mathcal{K}_3$.

An intuitive idea for saving some of the further questions generates a rule for question selection. After having received a correct answer to question a, we are left with the following possible knowledge states:

$$\{a, c\}, \{a, b, c\}, \{a, c, d\}, \{a, b, c, d\}, \{a, b, c, e\}, \{a, b, c, d, e\}.$$

At this time, we will surely not ask question c. Moreover, question d apparently prevails over questions b and e, because it will for any response reduce the number of possible knowledge states to 3 (while a correct answer to b or a wrong answer to e will leave 4 knowledge states). In precise terms, we select the question q that balances as closely as possible the number of remaining knowledge states that contain q versus the number of remaining knowledge states that do not contain q. Although this 'splitting rule' does not necessarily generate a procedure with an overall minimum number of questions (see next paragraph), it is a reasonable heuristic.

The design of a deterministic assessment procedure raises many questions that we will not consider here. For instance, how to devise the 'best' assessment procedure (with respect to some optimality criteria to be defined)? This problem is intensively studied in computer science: given a set of external nodes, or actions to be taken, a 'good' binary tree or decision table is sought. The interested reader may consult Hyafil and Rivest (1976) and its references.

Returning to our context, we remark that there are situations in which we discover assessment procedures before the true organization of the tested information. From observing a teacher, we may record a test protocol for a body of knowledge that is still imprecise to us. In other words, we have at our disposal a diagram as in Figure 1.7, but without the labeling of the nodes. Can we discover the set Q and the collection $\mathcal{K}$? A partial, affirmative answer can be found in Degreef, Doignon, Ducamp, and Falmagne (1986).

Exercise 1.8.1 Draw diagrams representing other deterministic assessment procedures for the same knowledge space as in Figure 1.7. Count the total number of nodes (internal nodes corresponding to questions asked, and external nodes corresponding to knowledge states). What do you see? □

Exercise 1.8.2 For a knowledge space formed by k knowledge states, how many nodes are shown in a diagram of a deterministic assessment procedure? □

Exercise 1.8.3 Try to find, for the knowledge space $(Q, \mathcal{K}_3)$, a deterministic assessment procedure that minimizes

1. the average number of questions asked to assess the knowledge states; or
2. the maximum number of questions asked to assess a knowledge state.

□

Exercise 1.8.4 Let $(Q, \mathcal{K})$ be the knowledge space associated to a simple order. Call the height of a deterministic assessment procedure the maximum number of questions to ask before assessing any state. What is the lowest possible height? (This is similar to dichotomic search in a dictionary.) If the (general) problem is too difficult, work it out for small numbers of questions ($|Q| = 3, 4, \ldots$). □

1.9 A setting for probabilistic assessment

From now on, we abandon the strong assumptions made in the previous Section: lucky guesses and careless errors enter the theory. Also, the knowledge state of the student we are examining needs not be constant. Allowing for its fluctuation is justified in our model by two arguments. First, some apparent inconsistencies in the student's responses may be explained this way. Second, we cannot hope to make a precise, univoque estimation of the subject's knowledge state. The best we can achieve is to produce various likelihoods for a number of knowledge states; hence there is no harm in admitting variations of the student state.

This general approach requires some more mathematical tools, mostly from elementary probability theory. Although we will not provide here definitions of the basic concepts, we will review in Section 1.13 a few facts about Markov chains. The reader is referred to Chung (1974) for an introduction to probability theory. The remainder of this Section informally sketches a class of probabilistic processes for the assessment of knowledge. Technical definitions are collected in Sections 1.10, 1.11, and 1.12, mathematical results in Section 1.14, and more elaborate examples in Section 1.15.

The way we model a student will be much more general than before. In the special case considered until now, a student is represented by one single, constant element in the family $\mathcal{K}$ of possible knowledge states. At present, we allow fluctuations in the student's knowledge. Thus his or her knowledge state varies among (a limited number of) elements K in $\mathcal{K}$. Moreover, we assume that the relative frequencies of his various possible K are governed by accurate probability values. To summarize, a student is modeled as a probability distribution π on $\mathcal{K}$. Hence, for any knowledge state K in $\mathcal{K}$, the number $\pi(K)$ is the probability for the student to be, at any given time, in the knowledge state K. By definition, π is any mapping from $\mathcal{K}$ to $[0,1]$ such that

$$\pi(K) \geq 0 \qquad \text{and} \qquad \sum_{K \in \mathcal{K}} \pi(K) = 1.$$

This probability distribution will govern in first approximation the responses of the student. In the simplest case, the answer provided to question q is correct iff the knowledge state of the student contains q. However, in order to tackle careless errors and lucky guesses, we attach to any question q two real parameters β_q and γ_q with $0 \leq \beta_q \leq 1$ and $0 \leq \gamma_q \leq 1$, and we assert:

> given that the student is in the knowledge state K, the probability at any time that his or her answer to question q is correct equals
>
> $$\begin{array}{ll} 1 - \beta_q & \text{if } q \text{ belongs to the student's knowledge state } K, \\ \gamma_q & \text{if } q \text{ does not belong to } K. \end{array}$$

We now turn to the assessment process. It retains at any step a list of knowledge states plausible for the student under examination. The family formed by these knowledge states is called the *marker* (in Falmagne & Doignon, 1988b, these knowledge states are said to be marked). Thus, the marker is a subfamily of $\mathcal{K}$, or equivalently an element of $2^{\mathcal{K}}$. When the correctness of an answer has been evaluated, the marker is accordingly updated. More specifically, the new marker is chosen according to a probability distribution on $2^{\mathcal{K}}$ that depends only on the last question and response, and the actual marker. Finally, we indicate in a questioning rule how to choose the next question. It will be selected in Q according to a probability distribution determined by the actual marker.

Random variables will be denoted by bold letters. A *random variable* $\mathbf{X}$ with values in the finite set V is completely described by the probabilities $\Pr(\mathbf{X} = v)$ for all $v \in V$, or, equivalently, by the probabilities $\Pr(\mathbf{X} \in T)$ for all the subsets T of V. (Except for Section 1.16, we need only consider *finite* sets V; this makes the theory simpler.) The probability values $\Pr(\mathbf{X} \in T)$ are what we need in practice. However, the mathematical definition requires first the consideration of some probability space $(\Omega, \mathcal{E}, \Pr)$, where Ω is a set of *elementary events*, and Pr a *probability measure* defined on the collection $\mathcal{E}$ of *events*, with $\mathcal{E}$ a specific collection of subsets of Ω. Probability theory imposes axioms on the triple $(\Omega, \mathcal{E}, \Pr)$ (see e.g. Chung, 1974; or Feller, 1970). To summarize grossly, the collection $\mathcal{E}$ is assumed to be a σ–algebra, and the mapping Pr to be σ–additive with values in the interval $[0, 1]$. From these technicalities, the reader needs to retain that a number $\Pr(E)$ is attached to each event E from $\mathcal{E}$, and called the *probability* of event E. Mathematically, a random variable defined on the probability space $(\Omega, \mathcal{E}, \Pr)$ with values in the finite set V is a mapping $\mathbf{X}$ from Ω to V such that for any subset T of V, the subset $\{\omega \in \Omega \mid \mathbf{X}(\omega) \in T\}$ of Ω belongs to $\mathcal{E}$. One may then set

$$\Pr(\mathbf{X} \in T) = \Pr(\omega \in \Omega \mid \mathbf{X}(\omega) \in T),$$

for any subset T of V. In the following developments, we will meet many random variables, for instance $\mathbf{R}_1$, $\mathbf{R}_2$, $\mathbf{R}_{10}$, and $\mathbf{Q}_6$; it is tacitly assumed that they are all defined on the same probability space (which we do not need to specify). In this way, for any subsets T_1, T_2, T_3, T_4 of V, the following

$$\Pr(\mathbf{R}_1 \in T_1, \mathbf{R}_2 \in T_2, \mathbf{R}_{10} \in T_3, \mathbf{Q}_6 \in T_4)$$

is meaningful. The same holds for conditional probabilities such as

$$\Pr(\mathbf{R}_1 \in T_1, \mathbf{R}_2 \in T_2 \,|\, \mathbf{R}_{10} \in T_3, \mathbf{Q}_6 \in T_4) = p$$

whenever $\Pr(\mathbf{R}_{10} \in T_3, \mathbf{Q}_6 \in T_4) \neq 0$; this number p is the probability of the event ($\mathbf{R}_1 \in T_1$, $\mathbf{R}_2 \in T_2$) knowing that ($\mathbf{R}_{10} \in T_3$, $\mathbf{Q}_6 \in T_4$) holds. It is thus equal to

$$\frac{\Pr(\{\omega \in \Omega \mid \mathbf{R}_1(\omega) \in T_1, \mathbf{R}_2(\omega) \in T_2, \mathbf{R}_{10}(\omega) \in T_3, \mathbf{Q}_6(\omega) \in T_4)}{\Pr(\omega \in \Omega \mid \mathbf{R}_{10}(\omega) \in T_3, \mathbf{Q}_6(\omega) \in T_4)}.$$

Notice also that an expression like

$$\Pr(\mathbf{R}_1 \in T_1 \,|\, \mathbf{R}_{10} \in T_2, \mathbf{Q}_6) = \ell$$

is to be interpreted as follows: for any subset T_3 of V, one has

$$\Pr(\mathbf{R}_1 \in T_1 \,|\, \mathbf{R}_{10} \in T_2, \mathbf{Q}_6 \in T_3) = \ell.$$

A *stochastic process* $(\mathbf{X}_n)$ is a sequence $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n, \ldots$ of random variables, all defined on the same probability space, and taking their values in the same set. Most often, and it is always the case here, the index values $1, 2, \ldots, n, \ldots$ designate successive time points. Turning back to knowledge assessment, we consider again a knowledge space $(Q, \mathcal{K})$, and introduce four random variables at any step n:

$\mathbf{R}_n$	for the correctness of the answer,	with values in $\{0,1\}$,
$\mathbf{Q}_n$	for the question asked,	with values in Q,
$\mathbf{K}_n$	for the student's knowledge state,	with values in $\mathcal{K}$,
$\mathbf{M}_n$	for the marker,	with values in $2^{\mathcal{K}}$.

Altogether, there results a stochastic process $(\mathbf{R}_n, \mathbf{Q}_n, \mathbf{K}_n, \mathbf{M}_n)$. That is, for each natural number n, we have a random variable $(\mathbf{X}_n) = (\mathbf{R}_n, \mathbf{Q}_n, \mathbf{K}_n, \mathbf{M}_n)$. This variable takes its values in $\{0,1\} \times Q \times \mathcal{K} \times 2^{\mathcal{K}}$. The reader may have guessed that 0 stands for a correct answer, 1 for an uncorrected answer. Values of $\mathbf{R}_n$, $\mathbf{Q}_n$, $\mathbf{K}_n$, $\mathbf{M}_n$ will often be denoted as r, q, K, and Ψ (or Υ) respectively. The complete history of the stochastic process $(\mathbf{X}_n)$ from trial 1 to trial n will be abbreviated as

$$\mathbf{W}_n = ((\mathbf{R}_n, \mathbf{Q}_n, \mathbf{K}_n, \mathbf{M}_n), \ldots, (\mathbf{R}_1, \mathbf{Q}_1, \mathbf{K}_1, \mathbf{M}_1)),$$

the empty story being designated as $\mathbf{W}_0$. Using these notations, our general axioms concern the probabilities for the values taken by the random variables $\mathbf{R}_n$, $\mathbf{Q}_n$, and $\mathbf{K}_n$, for all $n \geq 1$, and the random variable $\mathbf{M}_n$, for $n > 1$. Later on, we will be more specific about the initial value $\mathbf{M}_1$. A few comments on the axioms are given after Definition 1.9.1.

Marking rule. For some function $\mu : 2^{\mathcal{K}} \times \{0,1\} \times Q \times 2^{\mathcal{K}} \to [0,1]$,

$$\Pr(\mathbf{M}_{n+1} = \Psi \,|\, \mathbf{R}_n = r, \mathbf{Q}_n = q, \mathbf{K}_n, \mathbf{M}_n = \Upsilon, \mathbf{W}_{n-1}) = \mu(\Psi, r, q, \Upsilon).$$

Knowledge rule. For some fixed probability distribution π on $\mathcal{K}$,

$$\Pr(\mathbf{K}_n = K \,|\, \mathbf{M}_n, \mathbf{W}_{n-1}) = \pi(K).$$

Questioning rule. For some function $\tau : Q \times 2^Q \to [0,1]$,

$$\Pr(\mathbf{Q}_n = q \,|\, \mathbf{K}_n, \mathbf{M}_n = \Psi, \mathbf{W}_{n-1}) = \tau(q, \Psi).$$

Response rule.

$$\Pr(\mathbf{R}_n = 1 \,|\, \mathbf{Q}_n = q, \mathbf{K}_n = K, \mathbf{M}_n, \mathbf{W}_{n-1}) = \begin{cases} 1 - \beta_q & \text{if } q \in K; \\ \gamma_q & \text{if } q \in K. \end{cases}$$

Definition 1.9.1 A stochastic process $(\mathbf{R}_n, \mathbf{Q}_n, \mathbf{K}_n, \mathbf{M}_n)$ governed by these four rules will be called a *stochasticassessment process parametrized by* μ, π, τ, β_q, *and* γ_q on the knowledge space $(Q, \mathcal{K})$ (where $q \in Q$). The quantities β_q and γ_q are respectively the *error* and *guessing probabilities* for question q. □

The four rules used in Definition 1.9.1 constrain the evolution over time of the process $(\mathbf{R}_n, \mathbf{Q}_n, \mathbf{K}_n, \mathbf{M}_n)$. More precisely, they specify the probabilities of the random variable values at one step from the values taken at the previous step. As a matter of fact, the value of $\mathbf{K}_n$ is governed only by the probability distribution π, constant over time, according to the Knowledge Rule. Starting from an initial marker $\mathbf{M}_1$, the values of $\mathbf{Q}_1$, in other terms the first question asked, have probabilities derived from the value of $\mathbf{M}_1$ alone. The correctness of the answer collected for that question has a probability computed in the Response Rule from the actual knowledge state of the student, together with the error and guessing parameters. Then the marker is updated: the probabilities of the possible values of $\mathbf{M}_2$ reflect the observed answer to the selected question, and also the value of $\mathbf{M}_1$. A new iteration starts then. In a similar way, the probabilities of the values of $\mathbf{K}_2$, $\mathbf{Q}_2$, $\mathbf{R}_2$, and $\mathbf{M}_3$ are obtained from the four rules. The same pattern repeatedly applies. From this general scheme, it is easily understood how to build examples of stochastic assessment processes: one only needs to specify the parameters μ, π, τ, β_q, and γ_q. (There are some plain restrictions on their values, because the four rules must produce conditional probabilities.)

The concept of a stochastic assessment process just defined is of a fairly general nature. Of course, some technical conditions will soon be imposed on these functions. Nevertheless, a few interesting properties hold in the general case.

Recall that the stochastic process $(\mathbf{X}_n)$ is *Markovian* whenever the probabilities of its values at step $n+1$ depend only on the probabilities of its values at step n, thus not on the whole previous history of the process. Formally, this means

$$\begin{aligned} &\Pr(\mathbf{X}_n = x \,|\, \mathbf{X}_{n-1} = y_{n-1}, \mathbf{X}_{n-2} = y_{n-2}, \ldots, \mathbf{X}_1 = y_1) \\ = \quad &\Pr(\mathbf{X}_n = x \,|\, \mathbf{X}_{n-1} = y_{n-1}). \end{aligned}$$

Moreover, the process is *homogeneous* (with respect to time) when each of the above *transition probabilities* remains constant over all steps n of the process.

It can be easily checked that a stochastic assessment process $(\mathbf{X}_n) = (\mathbf{R}_n, \mathbf{Q}_n, \mathbf{K}_n, \mathbf{M}_n)$ as in Definition 1.9.1 is Markovian and homogeneous, as well as each of the following two derived processes: $(\mathbf{M}_n)$, and $(\mathbf{Q}_n, \mathbf{M}_n)$. Section 1.13 contains a survey of the fundamental results on Markovian chains, a particular case of Markovian processes.

Let us now give names to some special types of stochastic assessment processes.

DEFINITION 1.9.2 A stochastic assessment process $(\mathbf{R}_n, \mathbf{Q}_n, \mathbf{K}_n, \mathbf{M}_n)$ parametrized by μ, π, τ, β_q, and γ_q is *fair* when $\gamma_q = 0$ for each q (no lucky guess). The same process is *straight* when $\gamma_q = 0$ and moreover $\beta_q = 0$, for each q (no lucky guess, no careless error). □

DEFINITION 1.9.3 Let $(\mathbf{R}_n, \mathbf{Q}_n, \mathbf{K}_n, \mathbf{M}_n)$ be a stochastic assessment process parametrized by μ, π, τ, β_q, and γ_q on the knowledge space $(Q, \mathcal{K})$. The *support* of π consists of all knowledge states K in $\mathcal{K}$ such that $\pi(K) > 0$. Furthermore, π has *unit support* when there is only one knowledge state in its support. □

Thus π has unit support when there is a (unique) knowledge state K with $\pi(K) = 1$.

Even with the restrictions embodied in Definitions 1.9.2 and 1.9.3, the stochastic assessment processes would remain quite general. We need specific questioning and marking rules; these will be described in the next two Sections. Results about the convergence of the derived processes will then be established in Sections 1.14 and 1.15. Moreover, computer simulations will also be mentioned in Section 1.17. These results and simulations provide theoretical support for the choice of definitions and model we have made.

EXERCISE 1.9.1 Determine whether the following processes derived from the one in Definition 1.9.1 are Markovian: $(\mathbf{R}_n)$, $(\mathbf{Q}_n)$, $(\mathbf{K}_n)$, $(\mathbf{R}_n,\ \mathbf{Q}_n)$, $(\mathbf{K}_n, \mathbf{M}_n)$. □

1.10 Questioning rule

In order to prepare the formulation of a particular questioning rule, let us expose the underlying idea, give a preliminary example and state some definitions. At the start of the assessment process, we have no information on the examinee knowledge. It thus makes sense to put all the knowledge states in the marker, that is $\mathbf{M}_1 = \mathcal{K}$. Inspired by the splitting rule introduced in Section 1.8, we decide to select a first question which on the average reduces the size of the marker as strongly as possible. The same idea can be applied in the choice of the next questions. Nevertheless, such a rule will surely leave us at some step with a marker consisting of a single knowledge state (recall that $\mathcal{K}$ is finite), in which case we want to explore knowledge states that resemble to our actual unique candidate. Therefore we slightly generalize the above formulation as follows. Given the actual marker Ψ, we first build the collection of all knowledge states that are not too far away from a member of Ψ. Then, we pick a question that splits this new family.

To formalize this, we will rely on the following concepts, derived from the distance $d(K, K')$ between knowledge states K and K' (see Section 1.7, paragraph before Example 1.7.3).

DEFINITION 1.10.1 The *ϵ–neighborhood* $N(\Psi, \epsilon)$ of some family Ψ of knowledge states in the knowledge space $(Q, \mathcal{K})$ consists of all knowledge states K' with distance at most ϵ to some member of Ψ:

$$N(\Psi, \epsilon) = \{K' \in \mathcal{K} \,\big|\, d(K, K') \leq \epsilon \text{ for some } K \in \Psi\}.$$

Any question q in Q determines two subcollections of $N(\Psi, \epsilon)$ with respect to membership of q:

$$\begin{aligned} N_q(\Psi, \epsilon) &= \{K' \in \mathcal{K} \mid q \in K', d(K, K') \leq \epsilon \text{ for some } K \in \Psi\} \\ &= N(\Psi, \epsilon) \cap \mathcal{K}_q, \\ N_{\bar{q}}(\Psi, \epsilon) &= \{K' \in \mathcal{K} \mid q \notin K', d(K, K') \leq \epsilon \text{ for some } K \in \Psi\} \\ &= N(\Psi, \epsilon) \cap \mathcal{K}_{\bar{q}}. \end{aligned}$$ □

For simplicity, abbreviations like $N(K, \epsilon)$ for $N(\{K\}, \epsilon)$ will be used.

Example 1.10.1 Consider again the knowledge space $(Q, \mathcal{K}_3)$ (see Example 1.7.1 or Figure 1.6):

$$\begin{aligned}\mathcal{K}_3 &= \{\emptyset, \{b\}, \{c\}, \{a,c\}, \{b,c\}, \{b,d\}, \{a,b,c\}, \{a,c,d\},\\ &\qquad \{b,c,d\}, \{a,b,c,d\}, \{a,b,c,e\}, Q\}.\end{aligned}$$

Assume $\epsilon = 1$, and take $\Psi = \{\{a,c\}, \{b,c\}\}$. Then one derives (referring to Table 1.2):

$$N(\Psi, 1) = \{\{b\}, \{c\}, \{a,c\}, \{b,c\}, \{a,b,c\}, \{a,c,d\}, \{b,c,d\}\},$$

and

$$\begin{aligned}
N_a(\Psi,1) &= \{\{a,c\}, \{a,b,c\}, \{a,c,d\}\},\\
N_{\bar{a}}(\Psi,1) &= \{\{b\}, \{c\}, \{b,c\}, \{b,c,d\}\},\\
N_b(\Psi,1) &= \{\{b\}, \{b,c\}, \{a,b,c\}, \{b,c,d\}\},\\
N_{\bar{b}}(\Psi,1) &= \{\{c\}, \{a,c\}, \{a,c,d\}\},\\
N_c(\Psi,1) &= \{\{c\}, \{a,c\}, \{b,c\}, \{a,b,c\}, \{a,c,d\}, \{b,c,d\}\},\\
N_{\bar{c}}(\Psi,1) &= \{\{b\}\},\\
N_d(\Psi,1) &= \{\{a,c,d\}, \{b,c,d\}\},\\
N_{\bar{d}}(\Psi,1) &= \{\{b\}, \{c\}, \{a,c\}, \{b,c\}, \{a,b,c\}\},\\
N_e(\Psi,1) &= \emptyset,\\
N_{\bar{e}}(\Psi,1) &= \{\{b\}, \{c\}, \{a,c\}, \{b,c\}, \{a,b,c\}, \{a,c,d\}, \{b,c,d\}\}. \qquad \Box
\end{aligned}$$

Before formulating a particular questioning rule, let us make clear that we need a value for ϵ to build an ϵ–neighborhood of the marker. A function of the size of the marker itself will be used. If Ψ denotes the marker, the value r of ϵ is thus derived from $|\Psi|$. Then $N_q(\Psi, r)$ and $N_{\bar{q}}(\Psi, r)$ make sense: they contain all the knowledge states K at distance at most ϵ from some member of Ψ, with moreover $q \in K$, resp. $q \notin K$. The next question q is chosen in order to balance the sizes of these two families of knowledge states, that is q is taken in such a way that the two numbers $|N_q(\Psi, r)|$ and $|N_{\bar{q}}(\Psi, r)|$ are as close as possible one to the other.

Definition 1.10.2 Let ϵ be some function from the set of nonnegative integers to the same set. The following questioning rule is said to be *ϵ–half-split*. If Ψ denotes the actual marker, set $r = \epsilon(|\Psi|)$. The next question is uniformly drawn among the set of questions that minimize the absolute value

$$||N_q(\Psi, r)| - |N_{\bar{q}}(\Psi, r)|| \tag{1.6}$$ $\Box$

Example 1.10.2 will illustrate this Definition. In the formal notations of Definition 1.9.1, the questioning rule is ϵ–half-split when $\tau(q, \Psi)$ has a constant value for each question q that minimizes the quantity in Equation (1.6), and value 0 for each other question.

Example 1.10.2 Following Example 1.10.1 on, assume the marker is

$$\Psi = \{\{a,c\}, \{b,c\}\}.$$

If $\epsilon(2) = 1$, the values of the absolute size-differences are 1, 1, 5, 3, 7 for respectively questions a, b, c, d, and e. Hence, according to Definition 1.10.2, the next question will be a or b, each with probability 0.5. This really appears as a good choice for discriminating between $\{a, c\}$ and $\{b, c\}$. □

EXERCISE 1.10.1 Starting with $\mathcal{K}_3$ as in Example 1.10.1, how will the first question be chosen? That is: give for any question q in Q the probability that q will be asked at step 1. □

1.11 Marking rule

The design of a marking rule follows from the discussion in the previous Section. Recall that at any step a question is selected to split a family of knowledge states Υ that is 'slightly' larger than the present marker. We will define the new marker as formed by the knowledge states in Υ that are consistent with the information provided by the last answer and question. Notice that this rule is almost deterministic, in the sense that the new family is well-defined, not chosen among several ones.

We need a more stringent definition of the family Υ. If Ψ is the marker, we set $\Upsilon = N(\Psi, r)$ for some value of r. In Falmagne and Doignon (1988b), r is taken as an unspecified function from the size $|\Psi|$. Here, we will restrict ourselves to a simpler case. When the marker contains more than one knowledge state, we decide $r = 0$, henceforth $\Upsilon = \Psi$. Otherwise $\Psi = \{K\}$ for some knowledge state K. We put again $r = 0$, except when the collected answer does not 'confirm' K. This exception happens in two cases: when the answer to the last question q is correct and $q \notin K$, or when it is incorrect and $q \in K$.

DEFINITION 1.11.1 The marking rule is *selective* when for $n > 0$ the value of $\mathbf{M}_{n+1}$ is derived with probability 1 from the value Ψ of $\mathbf{M}_n$, according to the following cases.

Case (i):
$|\mathbf{M}_n| > 1, \quad \mathbf{R}_n = 1, \quad \mathbf{Q}_n = q, \quad$ and $\mathbf{M}_{n+1} = \Psi_q$;
Case (ii):
$|\mathbf{M}_n| > 1, \quad \mathbf{R}_n = 0, \quad \mathbf{Q}_n = q, \quad$ and $\mathbf{M}_{n+1} = \Psi_{\bar{q}}$;
Case (iii):
$\mathbf{M}_n = \{K\}, \quad \mathbf{R}_n = 1, \quad \mathbf{Q}_n = q \in K, \quad$ and $\mathbf{M}_{n+1} = \{K\}$;
Case (iv):
$\mathbf{M}_n = \{K\}, \quad \mathbf{R}_n = 1, \quad \mathbf{Q}_n = q \notin K, \quad$ and $\mathbf{M}_{n+1} = N_q(\{K\}, 1)$;
Case (v):
$\mathbf{M}_n = \{K\}, \quad \mathbf{R}_n = 0, \quad \mathbf{Q}_n = q \in K, \quad$ and $\mathbf{M}_{n+1} = N_{\bar{q}}(\{K\}, 1)$;
Case (vi):
$\mathbf{M}_n = \{K\}, \quad \mathbf{R}_n = 0, \quad \mathbf{Q}_n = q \notin K, \quad$ and $\mathbf{M}_{n+1} = \{K\}$. □

Example 1.11.1 Take again the knowledge space in Figure 1.6 or Table 1.2:

$$\begin{aligned}\mathcal{K}_3 \;=\;& \{\emptyset, \{b\}, \{c\}, \{a,c\}, \{b,c\}, \{b,d\}, \{a,b,c\}, \{a,c,d\},\\ & \{b,c,d\}, \{a,b,c,d\}, \{a,b,c,e\}, Q\}\end{aligned}$$

Each of the six cases in Definition 1.11.1 will be illustrated (statements are to be understood with probability 1). Notice first

$$N(\{a,c\},1) = \{\{c\}, \{a,c\}, \{a,b,c\}, \{a,c,d\}\}.$$

Then:

(i) and (ii): let $\mathbf{M}_n = \{\{a,c\},\{b,c\}\}$, and $\mathbf{Q}_n = a$.
If $\mathbf{R}_n = 1$, then $\mathbf{M}_{n+1} = \{\{a,c\}\}$, else $\mathbf{M}_{n+1} = \{\{b,c\}\}$;

(iii) and (v): let $\mathbf{M}_n = \{\{a,c\}\}$, and $\mathbf{Q}_n = a$.
If $\mathbf{R}_n = 1$, then $\mathbf{M}_{n+1} = \{\{a,c\}\}$, else $\mathbf{M}_{n+1} = \{\{c\}\}$;

(iv) and (vi): let $\mathbf{M}_n = \{\{a,c\}\}$, and $\mathbf{Q}_n = b$.
If $\mathbf{R}_n = 1$, then $\mathbf{M}_{n+1} = \{\{a,b,c\}\}$, else $\mathbf{M}_{n+1} = \{\{a,c\}\}$. □

While reading Definition 1.11.1, the reader should immediately raise an objection about the cases (iv) and (v). Notice that for $q \notin K$, the family $N_q(\{K\},1)$ is either empty or formed by the single set $K \cup \{q\}$, and that for $q \in K$, the family $N_{\bar{q}}(\{K\},1)$ is either empty or formed by the single set $K \setminus \{q\}$. An *empty* marker would be nonsense (notice that we intentionally forgot about the case $\mathbf{M}_n = \emptyset$ in Definition 1.11.1). If we make no assumption on the knowledge space $(Q,\mathcal{K})$ under consideration, Cases (iv) and (v) could lead to $\mathbf{M}_{n+1} = \emptyset$. Clearly, the questions q guaranteeing a nonempty new marker in Cases (iv) and (v) constitute the fringe $F(K)$ (cf. Definition 1.7.2), and we want to have both

$$\begin{aligned}&(Q \setminus K) \cap F(K) \neq \emptyset \text{ for } K \neq Q, \qquad \text{and}\\ &K \cap F(K) \neq \emptyset \text{ for } K \neq \emptyset.\end{aligned}$$

By Proposition 1.7.2(i), both requirements are surely fulfilled when $(Q,\mathcal{K})$ is well graded. This assumption will be part of our definition of unitary processes for knowledge assessment (see Definition 1.12.1). There is no harm in deleting the empty set from the set of possible values for the marker.

Exercise 1.11.1 Let q be a question and K be a knowledge state in a knowledge structure $(Q,\mathcal{K})$, with $q \notin \mathcal{K}$. Show that, in general, the cardinality of the set $N_q(\{K\},1)$ can take only two values. Characterize the situation in which it is equal to 1, by referring to $F(K)$. □

1.12 Unitary processes

The definition we now state was prepared along the two preceding Sections.

Definition 1.12.1 Let $(\mathbf{X}_n) = (\mathbf{R}_n, \mathbf{Q}_n, \mathbf{K}_n, \mathbf{M}_n)$ be a stochastic assessment process with $\mathbf{M}_n$, for $n \geq 1$, taking as values nonempty subfamilies of the family $\mathcal{K}$ of knowledge states. Assume its underlying knowledge space $(Q, \mathcal{K})$ is well graded (Definition 1.7.1), its questioning rule is ϵ–half-split (Definition 1.10.2), its marking rule is selective (Definition 1.11.1), and moreover

$$\epsilon(1) = 1, \qquad \epsilon(n) = 0 \text{ for } n > 1.$$

Then $(\mathbf{X}_n)$ will be said *unitary*, or for short called a *unitary process.* Its unspecified parameters, denoted as before, are now only the β_q (error probabilities), γ_q (guessing probabilities) and π. □

As a matter of fact, a small additional requirement is needed in this definition; see Remark 1.12.1.

Let us first collect some easy results, among whose some have been yet suggested.

PROPOSITION 1.12.1 *Assume that a unitary process satisfies* $\mathbf{M}_n = \{K\}$ *at some step* n*, with the fringe* $F(K)$ *of the knowledge state* K *satisfying* $|F(K)| \geq 2$*. Then* $\mathbf{Q}_n$ *takes with probability* 1 *its value in* $F(K)$*, and each question in* $F(K)$ *will be asked at step* n *with probability* $1/|F(K)|$. □

PROOF. The thesis derives from Definition 1.10.2. Just remark that

$$|N_q(\{K\}, 1)| - |N_{\bar{q}}(\{K\}, 1)| \tag{1.7}$$

equals

$$\begin{array}{rll} |F(K)| - 2 & \text{when} & q \in K \cap F(K), \\ 2 - |F(K)| & \text{when} & q \in F(K) \setminus K, \\ |F(K)| & \text{when} & q \in K \setminus F(K), \\ -|F(K)| & \text{when} & q \notin K \cup F(K). \end{array}$$

Thus, assuming $|F(K)| \geq 2$, the absolute value of the quantity in Equation (1.7) is minimized for $q \in F(K)$. ■

REMARK 1.12.1 Notice that the assumption $|F(K)| \geq 2$ in Proposition 1.12.1 is satisfied in a well-graded knowledge space for any knowledge state K with $K \neq \emptyset$ and $K \neq Q$ (because of the existence of two knowledge states K_1 and K_2 such that $K_1 \subset K \subset K_2$ and $|K \setminus K_1| = |K_2 \setminus K| = 1$). There can be only two special cases in which $|F(K)| = 1$: when $K = \emptyset$ and there is only one knowledge state of cardinality 1, or $K = Q$ and there is only one knowledge state of cardinality $|Q| - 1$. As we need the conclusion of Proposition 1.12.1 in these cases also, we add the following requirement in Definition 1.12.1: in these two special cases also, $\mathbf{Q}_n$ takes with probability 1 its value in the fringe $F(K)$ (which is clearly formed by a single question). □

PROPOSITION 1.12.2 *Consider again a unitary process. Then*

$$\begin{aligned} \Pr(\mathbf{M}_{n+1} = \{K \cup \{\mathbf{Q}_n\}\} \mid \mathbf{Q}_n \in (K_0 \setminus K) \cap F(K), \mathbf{M}_n = \{K\}, \mathbf{K}_n = K_0) \\ = \quad 1 - \beta_{\mathbf{Q}_n}. \end{aligned}$$

□

We leave the proof as an exercise, as well as the derivation of similar statements (see Exercises 1.12.2 and 1.12.3).

Any unitary process $(\mathbf{X}_n) = (\mathbf{R}_n, \mathbf{Q}_n, \mathbf{K}_n, \mathbf{M}_n)$ is a Markovian process, as is the derived process $(\mathbf{M}_n)$ (see before Definition 1.9.2: transitions from the value at one step to the value at the next step obey probabilities that do not depend on the previous history of the process). The main theoretical problems concern $(\mathbf{M}_n)$ and will be investigated in terms of Markov theory. For instance, does the evolution over time of $(\mathbf{M}_n)$ tell us something about the probability distribution π? Uncovering π is our ultimate goal. Nevertheless, it would be interesting to obtain partial information, e. g. to determine the support of π (family of states K for which $\pi(K) > 0$). Or one could make stronger assumptions on the process. An important one is to work under the hypothesis $\pi(K_0) = 1$ for some knowledge state K_0. Thus, the student's knowledge remains almost stable; only accidental deviations will be observed. This case, although very simplified, is an important one from a theoretical point of view. Our unitary processes should uncover the student's distribution π at least in that case! Positive results are reported in Section 1.14. Some other, non-elementary, situations will be analyzed in Section 1.15.

Exercise 1.12.1 Compute

$$|N_q(\{K\}, 1)| - |N_{\bar{q}}(\{K\}, 1)|$$

in the four cases mentioned in the proof of Proposition 1.12.1. □

Exercise 1.12.2 Establish the formula in Proposition 1.12.2. □

Exercise 1.12.3 Derive other formulas similar to the one in Proposition 1.12.2, for instance give

$$\Pr(\mathbf{M}_{n+1} = \{K\} \,|\, \mathbf{Q}_n \in (K_0 \setminus K) \cap F(K), \mathbf{M}_n = \{K\}, \mathbf{K}_n = K_0),$$
$$\Pr(\mathbf{M}_{n+1} = \{K \setminus \{\mathbf{Q}_n\}\} \,|\, \mathbf{Q}_n \in (K \setminus K_0) \cap F(K), \mathbf{M}_n = \{K\}, \mathbf{K}_n = K_0),$$
$$\Pr(\mathbf{M}_{n+1} = \{K \cup \{\mathbf{Q}_n\}\} \,|\, \mathbf{Q}_n \in F(K) \setminus (K_0 \cup K), \mathbf{M}_n = \{K\}, \mathbf{K}_n = K_0),$$
$$\ldots$$

As a matter of fact, one can compute all the probabilities of transitions from $\mathbf{M}_n = \{K\}$ to possible values of $\mathbf{M}_{n+1}$, assuming $\mathbf{K}_n = \{K_0\}$. □

1.13 Basics of Markov chains

Some basic terminology of Markov chains is settled in view of the next Section. The reader is referred to classical treatises for precise definitions and rigorous statements of results. We recommend for its remarkable pedagogical approach the textbook by Chung (1974), and for more advanced developments Kemeny and Snell (1965) or Feller (1970).

Only homogeneous finite Markov chains are of interest to us here. For all this Section, let $(\mathbf{X}_n)$ be a Markovian stochastic process (as in Section 1.9, all random variables are defined on a fixed probability space $(\Omega, \mathcal{E}, \Pr)$). We assume again that $\mathbf{X}_n$ takes its values in a finite set V, whose elements will be called *Markov states* and denoted as i, j, ... By assumption (with the conventions made in Section 1.9), we have:

$$\begin{aligned} &\Pr(\mathbf{X}_n = i \,|\, \mathbf{X}_{n-1} = j, \mathbf{X}_{n-2} = j_{n-2}, \ldots, X_1 = j_1) \\ = \quad &\Pr(\mathbf{X}_n = i \,|\, \mathbf{X}_{n-1} = j) \\ = \quad &p_{ij}, \end{aligned}$$

where, by the granted homogeneity, the transition probability p_{ij} from j to i does not depend on n. Accordingly, we say that $(\mathbf{X}_n)$ is a (finite, homogeneous) *Markov chain.*

Let us first study the relative reachability among Markov states. When $p_{ij} > 0$, we say that i is directly reachable from j, and write jDi. The transitive closure of the relation D on V is called the *reachability relation*, and is denoted by R. One can show that jRi is equivalent to the following assertion: if the process is in Markov state j, then the probability is positive that it will at some later moment be in state i. More precisely, the following holds:

for every $n \geq 1$, there is some $m > 0$ such that :

$$\Pr(\mathbf{X}_{n+m} = i \,|\, \mathbf{X}_n = j) > 0.$$

Notice that the relation R captures only a small part of the data in the matrix (p_{ij}) of transition probabilities (only positiveness of numbers matters for deriving R).

Example 1.13.1 Set $V = \{1, 2, \ldots, 11\}$ and take the 11×11 transition probabilities in the matrix $[p_{ij}]$ shown in Table 1.3. All the information, together with the relation of direct reachability, is displayed in Figure 1.8. The relation of reachability is shown in Figure 1.9. There are clearly subsets of $\{1, 2, \ldots, 11\}$ from which no 'arrow' emanates (that is, starts from a Markov state in the subset and ends in a Markov state out of the subset); this is the

Table 1.3. The matrix of transition probabilities in Example 1.13.1.

$$\begin{pmatrix} 0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.1 & 0.3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.3 & 0.1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.6 & 0.1 & 0 & 0.6 & 0 & 0 & 0.2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.4 & 0 & 0 & 0.8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.2 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

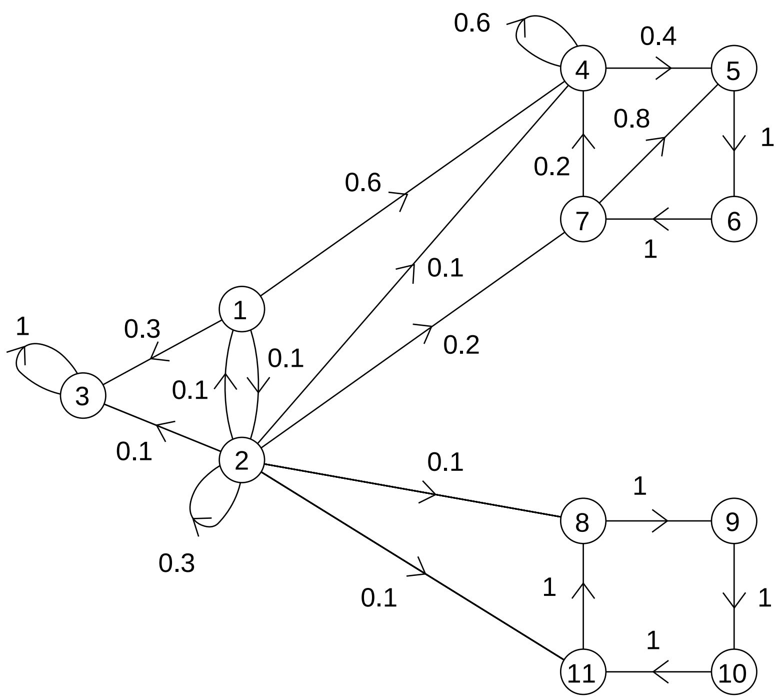


Figure 1.8. The direct reachability relation in Example 1.13.1.

case for $\{4, 5, 6, 7\}$, for $\{3\}$, for $\{3, 4, 5, 6, 7\}$, and also for $\{8, 9, 10, 11\}$, but not for $\{1, 2, 4, 5, 6, 7\}$ or $\{1\}$. □

A subset C of the set V of all Markov states is *closed* when jDi never holds for $j \in C$ and $i \in V \setminus C$. A closed set C of Markov states is an *ergodic* set when moreover any $i \in C$ can be reached from any $j \in C$ (allowing $i = j$). In Example 1.13.1, the ergodic sets are $\{3\}$, $\{4, 5, 6, 7\}$, and $\{8, 9, 10, 11\}$. Without entering into details, we mention that the last two ergodic sets differ in that the second is *periodic* (a return to 8 is possible only after a number of steps which is a multiple of 4), while the first is not. A Markov state is said to be *ergodic* if it belongs to some ergodic set; otherwise, it is *transient* (in Example 1.13.1, 1 and 2 are the transient Markov states).

One of the main problems about a Markov chain is to determine its asymptotic behavior. For instance, in the 'long' run, the values of the chain will be (with a 'high' probability) ergodic Markov states; more precisely,

$$\lim_{n\to\infty} \Pr(\mathbf{X}_n \text{ takes its value in some ergodic set}) = 1.$$

Moreover, if the chain ever comes in an ergodic set, it will stay there (this assertion holds with probability 1). Assume now that the chain in Example 1.13.1 at some step comes into the ergodic set $\{4, 5, 6, 7\}$, and let us study the resulting chain at later stages; we suppose that subsequent values are in that set. Which are the asymptotic probabilities of the various Markov states 4, 5, 6, 7, if these asymptotic probabilities make sense? Using aperiodicity, it is possible

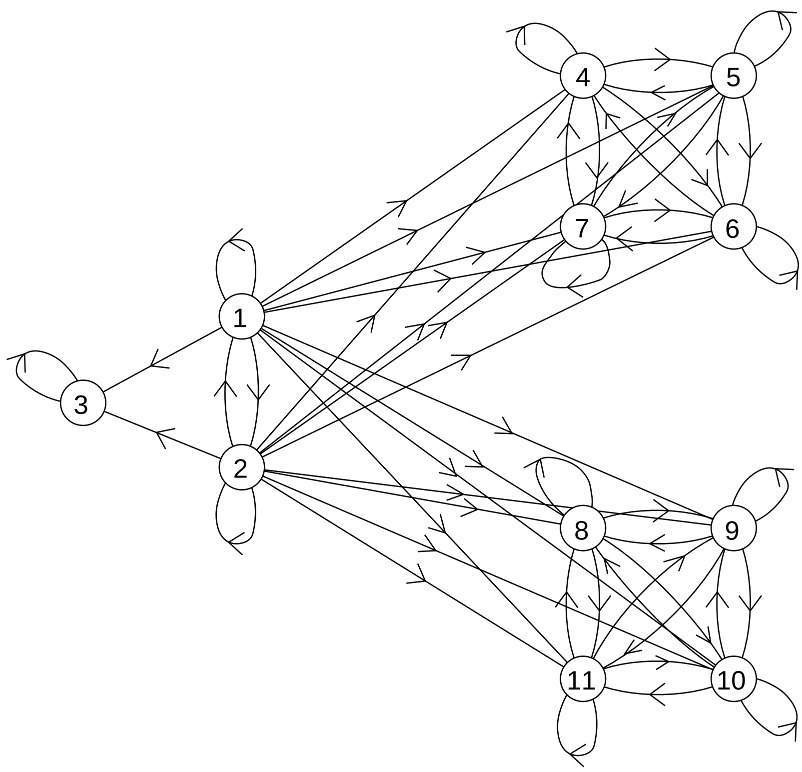


Figure 1.9. The reachability relation in Example 1.13.1.

to prove that the probabilities for the various Markov states at step n, that is

$$\Pr(\mathbf{X}_n = 4), \quad \Pr(\mathbf{X}_n = 5), \quad \Pr(\mathbf{X}_n = 6), \quad \Pr(\mathbf{X}_n = 7),$$

respectively converge to the values of a well-defined probability distribution on $\{4, 5, 6, 7\}$. Moreover this limit, or asymptotic, distribution probability has the particularly nice property of remaining stable under one-step transitions. To be more precise, let us consider the matrix of transition probabilities of the Markov chain restricted in the 'obvious way' to $\{4, 5, 6, 7\}$. With rows and columns indexed by the Markov states 4, 5, 6, 7 in that order, this matrix equals

$$\begin{pmatrix} 0.6 & 0 & 0 & 0.2 \\ 0.4 & 0 & 0 & 0.8 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

The limit-distribution probabilities p_4, p_5, p_6, p_7 for the Markov states 4, 5, 6, 7 satisfy the following system of linear equations (expressing stability under one-step transitions):

$$\begin{pmatrix} 0.6 & 0 & 0 & 0.2 \\ 0.4 & 0 & 0 & 0.8 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} p_4 \\ p_5 \\ p_6 \\ p_7 \end{pmatrix} = \begin{pmatrix} p_4 \\ p_5 \\ p_6 \\ p_7 \end{pmatrix},$$

and

$$p_4 + p_5 + p_6 + p_7 = 1.$$

Whence the unique solution

$$p_4 = \frac{1}{7}, \quad p_5 = \frac{2}{7}, \quad p_6 = \frac{2}{7}, \quad p_7 = \frac{2}{7}.$$

This probability distribution is the (unique) *stationary distribution* of a Markov chain having only one ergodic set C, with the further assumption that this set C is aperiodic. It gives a fairly complete information on the asymptotic behavior of the chain $(\mathbf{X}_n)$ from Example 1.13.1, restricted to $C = \{4, 5, 6, 7\}$ – or of the original chain $(\mathbf{X}_n)$, assuming it eventually enters $\{4, 5, 6, 7\}$.

Another interesting feature of the limit distribution obtained above relates to 'horizontal averaging'. Assume one counts, among the n first steps in which the chain $(\mathbf{X}_n)$ is in the set C, the numbers of times it is in each of the respective Markov states forming C. Thus, the relative frequencies of $\mathbf{X}_n = 4$, $\mathbf{X}_n = 5$, $\mathbf{X}_n = 6$, $\mathbf{X}_n = 7$ are obtained. A theoretical result asserts that these relative frequencies converge (for $n \to \infty$) to the asymptotic probabilities p_4, p_5, p_6, p_7, respectively.

We have avoided in this Section any of the technical proofs on which our general assertions should rely. The reader who feels the need for more rigor should consult classical treatises on Markov chains, e. g. those mentioned at the beginning of the Section.

Exercise 1.13.1 Study the Markov chains having transition probabilities given by the following matrices:

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix}, \begin{pmatrix} 0.5 & 1 \\ 0.5 & 0 \end{pmatrix}, \begin{pmatrix} 0.5 & 0 \\ 0.5 & 1 \end{pmatrix},$$

$$\begin{pmatrix} 0.4 & 0.3 & 0.2 \\ 0.6 & 0.7 & 0.8 \\ 0 & 0 & 0 \end{pmatrix}, \quad \begin{pmatrix} 0.2 & 0.3 & 0.2 \\ 0.8 & 0.7 & 0.8 \\ 0 & 0 & 0 \end{pmatrix}.$$

□

1.14 General results

Consider a well-graded knowledge space $(Q, \mathcal{K})$, and a *unitary* stochastic assessment process $(\mathbf{X}_n) = (\mathbf{R}_n, \mathbf{Q}_n, \mathbf{K}_n, \mathbf{M}_n)$ parametrized by μ, π, τ, β_q, and γ_q on $(Q, \mathcal{K})$ (as in Definition 1.12.1 and Remark 1.12.1). For all this Section except Theorem 1.14.2, we also assume that π has unit support, say $\{K_0\}$. Thus, the student's knowledge state is stochastically certain: it is equal to K_0 with probability 1. Nevertheless, answers provided by the student to the testing procedure could accidentally reflect another knowledge state. Moreover, answers to the question q are biased, due to careless errors or lucky guesses; recall that our parameters β_q and γ_q account for these perturbations.

The proofs of the first three theorems are easy, and left as exercises.

THEOREM 1.14.1 *For the unitary stochastic assessment process* $(\mathbf{X}_n)$ *having* $\beta_q = \gamma_q = 0$ *(straight case) and unit support* $\{K_0\}$*, one has for* $n > 0$

$$\Pr(K_0 \in \mathbf{M}_{n+1} \,|\, K_0 \in \mathbf{M}_n) = 1.$$

Thus, the set of all subfamilies of $\mathcal{K}$ *containing* K_0 *is a closed set for the Markov chain* $(\mathbf{M}_n)$*.* □

THEOREM 1.14.2 *For the unitary stochastic assessment process* $(\mathbf{X}_n)$*, we have for* $n > 0$*:*

$$\Pr(|\mathbf{M}_{n+1}| < |\mathbf{M}_n| \,\big|\, |\mathbf{M}_n| > 1) = 1,$$

and

$$\Pr(|\mathbf{M}_{n+1}| = 1 \,\big|\, |\mathbf{M}_n| = 1) = 1. \tag{1.8}$$

Hence there exists some positive integer m *such that, for* $n > m$*:*

$$\Pr(|\mathbf{M}_n| = 1) = 1.$$ □

In the fair case (no lucky guess, that is $\gamma_q = 0$ for each question q), we can improve Equation (1.8). This requires a careful analysis of all possible cases when $\mathbf{M}_n$ consists of a single knowledge state. Remember that $F(K)$ denotes the fringe of the knowledge state K (cf. Definition 1.7.2). (We will use $\bigcup \mathcal{F}$ to denote the union of the family $\mathcal{F}$ of knowledge states.)

THEOREM 1.14.3 *Let* $(\mathbf{X}_n)$ *be a fair, unitary stochastic assessment process having unit support* $\{K_0\}$*, and* K *be a knowledge state. Setting*

$$c = |F(K) \setminus (K \cup K_0)| + \sum_{q \in (F(K) \cap K_0) \setminus K} \beta_q + \sum_{q \in F(K) \cap K_0 \cap K} (1 - \beta_q),$$

the transition probabilities from the Markov state $\{K\}$ *are given as:*

$$\Pr(\mathbf{M}_{n+1} = \{K'\} \,|\, \mathbf{M}_n = \{K\}) =$$

$$\begin{cases} (1-\beta_q)/(|F(K)|) & \text{if } K' = K \cup \{q\}, q \in (F(K) \cap K_0) \setminus K; \\ 1/|F(K)| & \text{if } K' = K \setminus \{q\}, q \in (F(K) \cap K) \setminus K_0; \\ \beta_q/|F(K)| & \text{if } K' = K \setminus \{q\}, q \in F(K) \cap K \cap K_0; \\ c/|F(K)| & \text{if } K = K'; \\ 0 & \text{otherwise.} \end{cases}$$

In particular, one has for any knowledge state K *with* $K \not\subseteq K_0$*,*

$$\Pr(\mathbf{M}_{n+1} = \{K\} \,|\, \mathbf{M}_n = \{K_0\}) = 0.$$ □

From the last proposition, a nice convergence feature is derived for our assessment process in the straight case ($\gamma_q = \beta_q = 0$). For any step n after one on which $|\mathbf{M}_n| = 1$, the distance to K_0 from the single knowledge state remaining in $\mathbf{M}_n$ does not increase any more (of course, we mean that this holds with probability 1). Recall from Section 1.7 (paragraph before Example 1.7.3), that the distance from K to K_0 is defined as $d(K, K_0) = |K \triangle K_0|$.

THEOREM 1.14.4 *For the unitary stochastic assessment process* $(\mathbf{X}_n)$ *having* $\beta_q = \gamma_q = 0$ *(straight case) and unit support* $\{K_0\}$,

$$\lim_{n\to\infty} \Pr(\mathbf{M}_n = \{K_0\}) = 1.$$

Moreover, for any state K *satisfying* $d(K, K_0) = j > 0$, *and any positive integers* m, n,

$$\text{(1.9)} \quad \Pr(\mathbf{M}_{m+n} = \{K_0\} \,|\, \mathbf{M}_n = \{K\}) \geq \sum_{k=0}^{m-j} \binom{j+k-1}{k} \lambda^j (1-\lambda)^k$$

where λ *defined as*

$$\lambda = \min_{K,K'\in\mathcal{K}, K\neq K'} \frac{|(K \triangle K') \cap F(K)|}{|F(K)|}$$

is strictly positive. □

PROOF. First notice that $\lambda > 0$ amounts to well-gradedness, by Proposition 1.7.2(i). We next turn to Equation (1.9). Let $n \leq i < n+m$ and $\mathbf{M}_i = \{L\}$ (remember Equation (1.8)). The question $\mathbf{Q}_i = q$ is then chosen in $F(L)$. If $q \in (L \triangle K_0) \cap F(L)$, the new marker $M_{i+1} = \{L'\}$ will be 'closer' to $\{K_0\}$, more precisely $d(L', K_0) = d(L, K_0) - 1$ (this is to be understood with probability 1, and follows from the straightness assumption). If on the contrary $q \in F(L) \setminus (L \triangle K_0)$, one has $d(L', K_0) = d(L, K_0)$ because $L' = L$ (with probability 1). Thus, at each step i, we either decrease by 1 the distance to $\{K_0\}$, or leave it unchanged. The probability of a decrease is equal to

$$\frac{|(L \triangle K_0) \cap F(L)|}{|F(L)|}.$$

Thus the probability in (1.9) is equal to the probability that a number of at least j decreases occur among steps $n, n+1, \ldots, n+m-1$. At each step, the probability of a decrease is bounded from below by λ. It can then be shown that the probability in (1.9) is bounded from below by the probability of the following event, in the setting of a sequence of Bernoulli trials having outcomes either 'success' with probability λ, or 'failure' with probability $1-\lambda$:

$$A = \{\text{the number of trials required to observe } j \text{ successes is less or equal to } m\}.$$

The event A decomposes into the following mutually exclusive events, with $k = 0, 1, \ldots, m-j$:

$$A_k = \{\text{ the number of trials required to observe } j \text{ successes is equal to } j+k\}.$$

Since

$$\Pr(A_k) = \binom{j+k-1}{j-1} \lambda^{j-1}(1-\lambda)^k \lambda,$$

Equation (1.9) follows. To derive the first assertion in the statement, it is sufficient to prove that the right side of Equation (1.9) converges to 1 for $m \to \infty$ (which is obvious from its meaning as the probability of A). ■

Theorem 1.14.5 *For the unitary stochastic assessment process* $(\mathbf{X}_n)$ *having* $\gamma_q = 0$ *(fair case) and unit support* $\{K_0\}$*, the Markov chain* $(\mathbf{M}_n)$ *has a unique ergodic set* E*. This set* E *can contain only Markov states of the form* $\{K\}$*, with* $K \subseteq K_0$*, and necessarily contains* $\{K_0\}$*. If one assumes moreover* $\beta_q > 0$ *for all questions* q *in* K_0*, then* E *is exactly the family of all Markov states* $\{K\}$*, with* $K \subseteq K_0$*.* □

Proof. By Theorem 1.14.2, an ergodic Markov state contains only one knowledge state. It is easy to check that $\{K_0\}$ can be reached from any $\{K\}$, with $K \in \mathcal{K}$ (use well-gradedness together with Theorem 1.14.3). The assertions follow easily. ■

Exercise 1.14.1 Give a proof of Theorem 1.14.1. □

Exercise 1.14.2 Give a proof of Theorem 1.14.2. □

Exercise 1.14.3 Give a proof of Theorem 1.14.3. □

1.15 Some other examples

A strong assumption in the previous Section was that of a unit support. Unfortunately, results about convergence appear much more difficult to establish in the general case. We will restrict ourselves to the analysis of some examples. Interestingly enough, it will be possible in some situations to estimate the examinee's distribution just from the observed frequencies of the marker's states. However, a final example will illustrate the impossibility of such a derivation in general.

We use the same notations as in the previous Section, and assume that the stochastic assessment process is fair ($\gamma_q = 0$) and unitary. By Theorem 1.14.2, the Markov states of the chain $(\mathbf{M}_n)$ that contain more than one knowledge state are transient. Moreover, if K is a knowledge state in the support of π (that is, $\pi(K) > 0$), then $\{K\}$ is an ergodic Markov state of $(\mathbf{M}_n)$. As a matter of fact, the same holds for any knowledge state K satisfying $K' \subseteq K \subseteq K''$ for some knowledge states K' and K'' having $\pi(K') > 0$ and $\pi(K'') > 0$. More can be said using the subset Q^+ of questions that belong to at least one knowledge state from the support.

Notation 1.15.1 Set

$$Q^+ = \bigcap \{K \in \mathcal{K} \,|\, \pi(K) > 0\}.$$

(In the right-hand side, we have the intersection of a subfamily of knowledge states.) □

If $\beta_q > 0$ for each question q in Q^+ (i. e. the probability of a careless error is positive for each question in Q^+), then any Markov state $\{K\}$ with K included in some knowledge state from the support is ergodic. Let us look at the opposite case, that is the straight case.

Example 1.15.1 Take $Q = \{a, b, c, d, e\}$ and $\mathcal{K}$ to be the *chain* of knowledge states

$$\emptyset, \quad \{a\}, \quad \{a,b\}, \quad \{a,b,c\}, \quad \{a,b,c,d\}, \quad Q.$$

If $\beta_q = 0$ for all q in Q^+, the only ergodic Markov states $\{K\}$ will be obtained for K comprised between two knowledge states from the support (this can be checked directly, or derived from next Theorem 1.15.1). We further particularize our example, still assuming straightness, and let the support of π consists of the subchain of knowledge states

$$\{a,b\}, \{a,b,c\}, \quad \text{and} \quad \{a,b,c,d\}.$$

Consequently, there are exactly three ergodic Markov states, namely $\{K\}$ with K one of the three knowledge states in this subchain. The information we want to uncover amounts to the following three numbers:

$$\begin{aligned} \pi_1 &= \pi(\{a,b\}), \\ \pi_2 &= \pi(\{a,b,c\}), \\ \pi_3 &= \pi(\{a,b,c,d\}). \end{aligned}$$

The chain $(\mathbf{M}_n)$ has $2^6 - 1 = 63$ possible Markov states. Let us stick to the six Markov states $\{K\}$, with $K \in \mathcal{K}$, which altogether form a closed set. The direct reachability relation D, restricted to this closed set, is given in Figure 1.10.

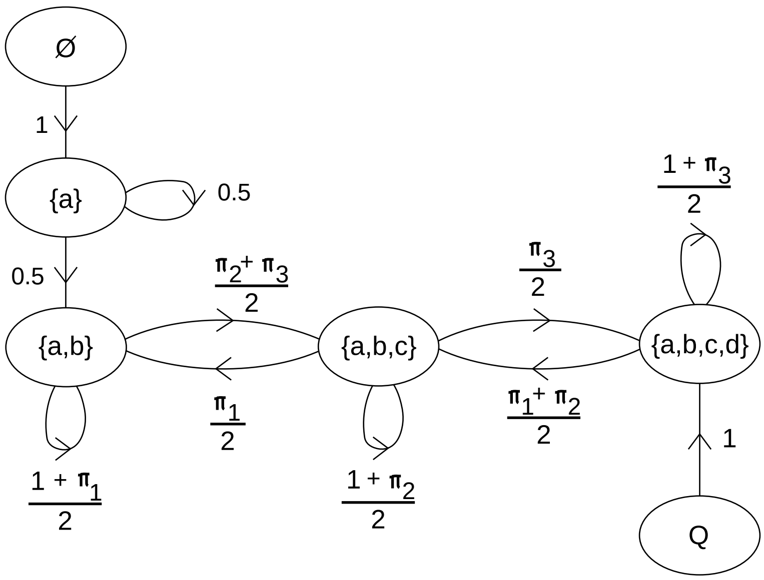


Figure 1.10. The direct reachability relation D on six of the Markov states of Example 1.15.1.

The unique ergodic set $\{\{a,b\}, \{a,b,c\}, \{a,b,c,d\}\}$ asymptotically captures the chain. The transition probabilities within this set are displayed in the matrix of Table 1.4.

One can compute the unique stationary probability distribution (p_1, p_2, p_3) as indicated in Section 1.13. Let us rather extract the unknown quantities

π_1, π_2, π_3 from the p_i. By stationarity:

$$\left\{\begin{array}{lllll} \frac{1+\pi_1}{2}p_1 & + & \frac{\pi_1}{2}p_2 & & = & p_1, \\ & & \frac{\pi_3}{2}p_2 & + & \frac{1+\pi_3}{2}p_3 & = & p_3, \end{array}\right.$$

from which we derive, using $\pi_1 + \pi_2 + \pi_3 = 1$:

$$\left\{\begin{array}{lll} \pi_1 & = & \frac{p_1}{p_1+p_2}, \\ \pi_2 & = & \frac{(p_2)^2 - p_1 p_3}{(p_1+p_2)(p_2+p_3)}, \\ \pi_3 & = & \frac{p_3}{p_2+p_3}. \end{array}\right.$$

Thus, observing a 'large' number of steps from the Markov chain $(\mathbf{M}_n)$ shows the presence of three ergodic Markov states $\{a,b\}$, $\{a,b,c\}$, and $\{a,b,c,d\}$ together with their actual frequencies that approximate p_1, p_2, and p_3. Through the above formulae, one then derives estimates of π_1, π_2, π_3. This example clearly shows how horizontal averaging may be useful for uncovering the unknown distribution. □

Of course, the method illustrated in Example 1.15.1 will require testing repeatedly a series of notions (a large stock of equivalent questions will be required). This is a drawback, due to the variation along time of the student's knowledge state; however, see Section 1.17 on computer simulations.

The assessment process we just described was based on a chain of knowledge states. Let us now take our favorite small example (pictured in Figure 1.6), which is well graded without being a chain.

Example 1.15.2 Take

$$\begin{aligned} \mathcal{K}_3 \;=\; & \{\emptyset, \{b\}, \{c\}, \{a,c\}, \{b,c\}, \{b,d\}, \{a,b,c\}, \{a,c,d\}, \\ & \{b,c,d\}, \{a,b,c,d\}, \{a,b,c,e\}, Q\}. \end{aligned}$$

We again assume that the stochastic assessment process is unitary. The Markov states of the form $\{K\}$, for all K in $\mathcal{K}_3$, constitute a closed set. Notice that all Markov states $\{K'\}$ directly reachable from the Markov state $\{K\}$ must satisfy $d(K,K') \leq 1$. A better control of the direct reachability relation D requires further assumptions. For instance, consider the straight case ($\beta_q = \gamma_q = 0$),

Table 1.4. The matrix of transition probabilities between the ergodic Markov states in Example 1.15.1; rows and columns are indexed by $\{a,b\}$, $\{a,b,c\}$, $\{a,b,c,d\}$ in that order.

$$\begin{pmatrix} \frac{1+\pi_1}{2} & \frac{\pi_1}{2} & 0 \\ \frac{\pi_2+\pi_3}{2} & \frac{1+\pi_2}{2} & \frac{\pi_1+\pi_2}{2} \\ 0 & \frac{\pi_3}{2} & \frac{1+\pi_3}{2} \end{pmatrix}$$

and take the distribution π to be concentrated on only two knowledge states, say $\{a, c\}$ and $\{b, c\}$. We thus have for some real number α:

$$\begin{aligned} \pi(\{a, c\}) &= \alpha > 0, \\ \pi(\{b, c\}) &= 1 - \alpha > 0. \end{aligned}$$

The resulting relation D on the set of Markov states $\{K\}$ is given in Figure 1.11. A unique ergodic set appears that collects four Markov states $\{K\}$, with K

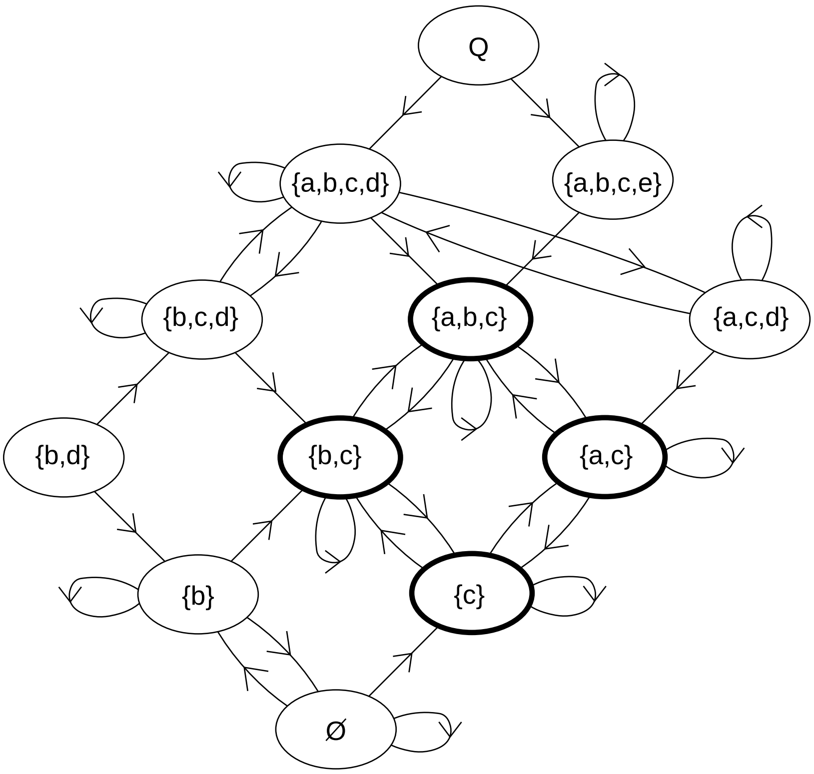


Figure 1.11. The direct reachability relation discussed in Example 1.15.2; the six ergodic Markov states are outlined.

equal to one of

$$\{c\}, \quad \{a, c\}, \quad \{b, c\}, \quad \{a, b, c\}. \qquad \square$$

In fact, this is a particular instance of the following result. (The set most to the left in Equation (1.10) is nothing else than Q^+, see Notation 1.15.1 and Exercise 1.15.1.)

THEOREM 1.15.1 *For a unitary, straight assessment process on the knowledge space $(Q, \mathcal{K})$ with parameter π, there is only one ergodic set of Markov states. It consists of all $\{K\}$, with K in $\mathcal{K}$ satisfying*

$$(1.10) \qquad \bigcap\{L \in \mathcal{K} \mid \pi(L) > 0\} \subseteq K \subseteq \bigcup\{L \in \mathcal{K} \mid \pi(L) > 0\}. \qquad \square$$

PROOF. Let us call J the set most to the right in Equation (1.10) and prove that $\{J\}$ is reachable from any Markov state $\{K\}$ (where $K \in \mathcal{K}$; notice $J \in \mathcal{K}$

because $\mathcal{K}$ is a space). By well-gradedness, and referring to the fringe $F(K)$ of K (see Definition 1.7.2 and Proposition 1.7.2(i)), we can pick a question q in $(J \bigtriangleup K) \cap F(K)$. If $\mathbf{M}_n = \{K\}$, there is a positive probability that question q will be asked. Now if $q \in J$ (resp. $q \notin J$), the answer will be correct (resp. incorrect) with a positive probability; hence the knowledge state $K \cup \{q\}$ (resp. $K \setminus \{q\}$) is directly reachable from K. In both situations, we have come closer to J (with respect to the symmetric difference distance). Repeating this argument with the new knowledge state, and again, shows that J is reachable from any Markov state $\{K\}$. There follows at once the uniqueness of the ergodic set, say E. More precisely, E consists of all Markov states $\{K\}$ reachable from $\{J\}$.

The first inclusion in Equation (1.10) is obviously necessary for $\{K\}$ to be reachable from $\{J\}$. To establish now its sufficiency, remember that well-gradedness implies the existence of a chain of knowledge states from K to J, growing by one question at a time (see Definition 1.7.1). And notice that the probability of an uncorrect answer to any question in $J \setminus \bigcap\{L \in \mathcal{K} \mid \pi(L) > 0\}$ is positive. ■

We return to our example, for which the unique ergodic set consists of four Markov states.

Example 1.15.3 (following Example 1.15.2) After a sufficient number of steps, the examiner will detect the ergodic set, with the relative frequencies of its states. Let us denote these as

$$\begin{array}{lll} \hat{p}_1 & \text{for} & \{c\}, \\ \hat{p}_2 & \text{for} & \{a,c\}, \\ \hat{p}_3 & \text{for} & \{b,c\}, \\ \hat{p}_4 & \text{for} & \{a,b,c\}. \end{array}$$

The discovery of the ergodic set reveals that each knowledge state K distinct from the four above has $\pi(K) = 0$ (using Theorem 1.15.1). Thus the observer is left with four unknown parameters, namely

$$\begin{array}{lll} \pi_1 & = & \pi(\{c\}), \\ \pi_2 & = & \pi(\{a,c\}), \\ \pi_3 & = & \pi(\{b,c\}), \\ \pi_4 & = & \pi(\{a,b,c\}), \end{array}$$

which govern the transition probabilities within the ergodic set. Figures 1.12 and 1.13 give a self-explaining sketch of the computations, and Table 1.5 the derived values of the transition probabilities. One can compute the theoretical values of the stationary distribution, and show that the π_i's can be derived from these. Thus the observer can infer estimates for π from estimates from p, the latter being obtained from observing the process. He will thus uncover the

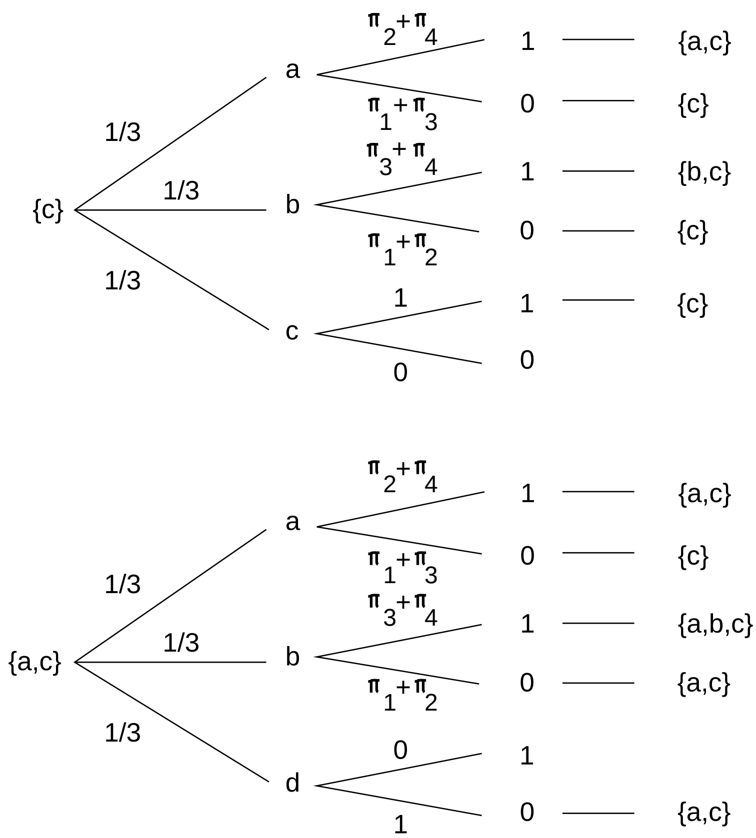


Figure 1.12. The first two diagrams for the computation of the transition probabilities in Example 1.15.3.

true values of π that we assumed from the start, namely

$$\begin{aligned}
\pi_1 &= \pi(\{c\}) &&= 0,\\
\pi_2 &= \pi(\{a,c\}) &&= \alpha > 0,\\
\pi_3 &= \pi(\{b,c\}) &&= 1-\alpha > 0,\\
\pi_4 &= \pi(\{a,b,c\}) &&= 0.
\end{aligned}$$

□

It should not be concluded from Example 1.15.3 that our model makes all distributions π uncoverable. Take the following simple example (first described by Mike Landy).

Table 1.5. Transition probabilities in the ergodic set of Example 1.15.3; rows and columns are indexed by $\{c\}$, $\{a,c\}$, $\{b,c\}$, and $\{a,b,c\}$ in that order.

$$\begin{pmatrix}
\frac{1+2\pi_1+\pi_2+\pi_3}{3} & \frac{\pi_1+\pi_3}{3} & \frac{\pi_1+\pi_2}{4} & 0\\
\frac{\pi_2+\pi_4}{3} & \frac{1+\pi_1+2\pi_2+\pi_4}{3} & 0 & \frac{\pi_1+\pi_2}{4}\\
\frac{\pi_3+\pi_4}{3} & 0 & \frac{2+\pi_1+2\pi_3+\pi_4}{4} & \frac{\pi_1+\pi_3}{4}\\
0 & \frac{\pi_3+\pi_4}{3} & \frac{\pi_2+\pi_4}{4} & \frac{2+\pi_2+\pi_3+2\pi_4}{4}
\end{pmatrix}$$

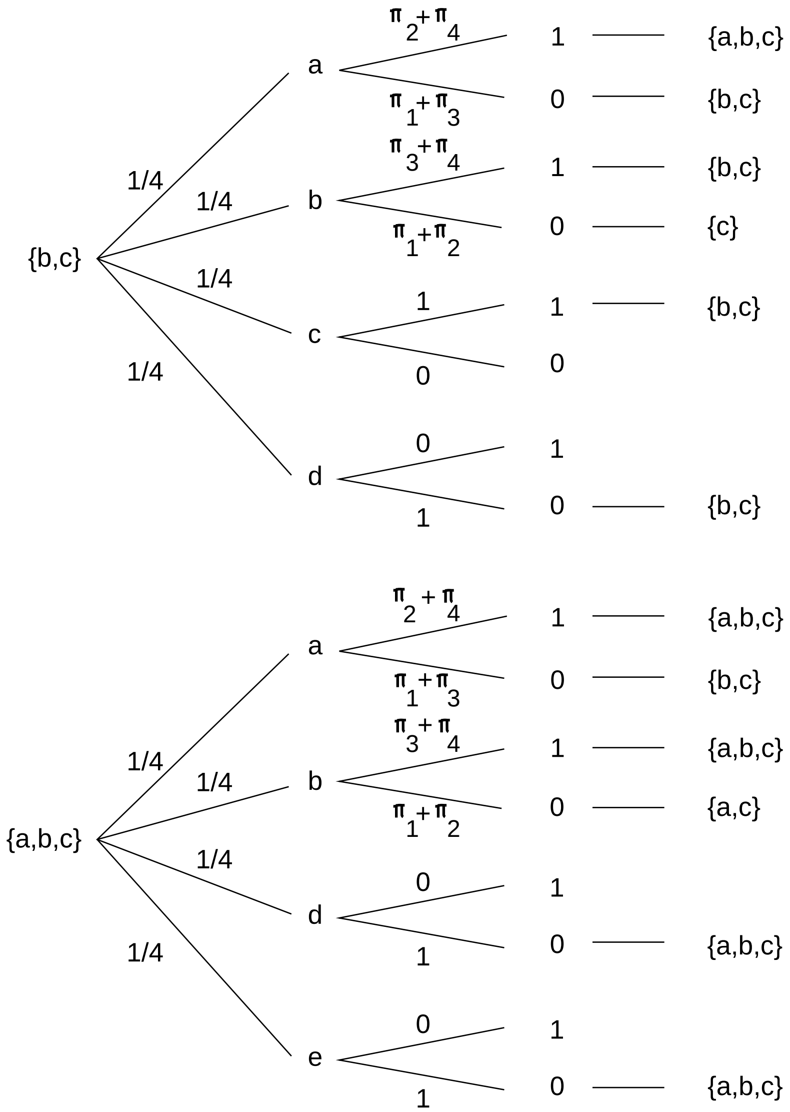


Figure 1.13. The two last diagrams for the computation of the transition probabilities in Example 1.15.3.

EXAMPLE 1.15.4 On the knowledge space

$$Q = \{a, b, c, d\}, \qquad \mathcal{K} = 2^Q,$$

define two different distributions π_1 and π_2 through

$$\begin{aligned} \pi_1(\{a,b\}) &= 0.5, & \pi_2(\{b,c\}) &= 0.5, \\ \pi_1(\{c,d\}) &= 0.5, & \pi_2(\{d,a\}) &= 0.5. \end{aligned}$$

In the straight case, it is easily checked that the probabilities of correct answers to each question from Q coincide for students characterized by the distributions π_1 and π_2, respectively. Hence, our unitary, straight stochastic assessment process cannot successfully ascribe the probabilistic knowledge in this example: it cannot discriminate between these two students. □

Let us take a second look at the phenomenon exhibited in Example 1.15.4. The student's answers (what we observe) are governed by the correctness probabilities, which means a number of parameters equal to the number of questions. On the other hand, the student's distribution (what we want to uncover) has as many parameters as the number of knowledge states. It is intuitively clear that for a number of questions less than the number of knowledge states, not all distributions π can be uncovered. Falmagne and Doignon (1988b) analyses the situation in terms of the rank of some matrix, and characterizes the cases in which any distribution is uncoverable. We guess that in practice the support of π is rather small, hence that in most cases π will be correctly estimated from observing one (or more) realization of our process.

This chapter provides a mathematical analysis of a Markovian process for knowledge assessment. Typical of the definition of such Markovian processes is the fact that they forget the past – in our case, only the information contained in the actual marker $\mathbf{M}_n$ is retained. In practice, however, the assesser may try to incorporate in his algorithm (part of) the information from previous steps. For instance, as soon as the marker contains only one knowledge state, he would record the relative frequencies of the knowledge states that are met along time. From Section 1.13, we know that these relative frequencies will converge to the probabilities of the Markov states of an ergodic set, and from this Section that the asymptotics may give access to the distribution of the student's knowledge state.

Exercise 1.15.1 Prove the equality $Q^+ = \{q \in Q \mid \pi(\mathcal{K}_q) > 0\}$, where $\pi(\mathcal{K}_q) = \sum_{K \in \mathcal{K}_q} \pi(K)$. □

1.16 Another model for probabilistic assessment

Another model of probabilistic assessment was described by Falmagne and Doignon (1988a). Based on a continuous stochastic process (taking its values in an infinite compact set, and again discrete in time), it was conceived chronologically before the Markov chain model on which this chapter centers. The rôle of the marker is replaced by a numerical likelihood for each knowledge state.

We will not go into details here, but just illustrate how the information evoluates over time. Consider again the knowledge space $(Q, \mathcal{K}_3)$. In Figure 1.14, the likelihoods of the various knowledge states are rendered by the areas of the dark rectangles. At the start, all knowledge states have the same plausibility.

After asking a question and checking the correctness of the answer, the likelihoods are consequently updated. Precise formulas for computing the new values are proposed by Falmagne and Doignon (1988a). The main result of the paper shows, in the unit support case, that the likelihoods will converge to the distribution π that is to be uncovered.

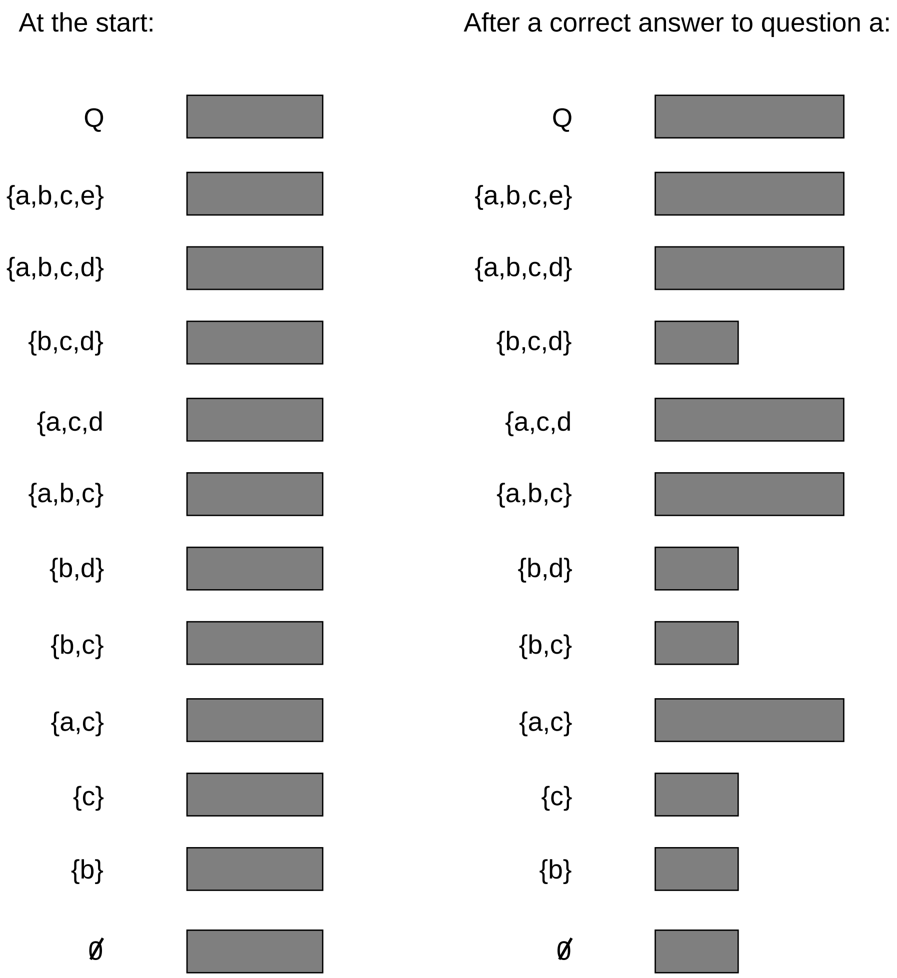


Figure 1.14. Illustration of the model in Section 1.16.

FOR FURTHER REFLECTION 1.16.1 List some properties you would like to specify on the way likelihoods are transformed (depending on the question asked and the validity of the answer). Then devise formulas satisfying your requirements. Compare your formulas with those of Falmagne and Doignon (1988a). □

1.17 Computer simulations

Both the unitary process model from Definition 1.12.1 and the stochastic model from Section 1.16 have been simulated on computers. They will be referred to as the discrete (D) and the continuous (C) models. In a first study (Villano, Falmagne, Johannesen, & Doignon, 1987), a set Q of 21 questions in high-school mathematics has been used. Information about dependencies among questions was obtained from an expert. Thus a surmise relation (Definition 1.3.1) was formed, and then the associated knowledge space $(Q, \mathcal{K})$ was constructed (Definition 1.4.1 and Theorem 1.4.1). It happened that among all the 2^{21} subsets of Q, only 302 belonged to the space $\mathcal{K}$.

A first simulation of assessment was done in the straight case (no careless error, no lucky guess). Let us summarize it by giving the average numbers of questions asked before obtaining a single knowledge state that belongs to the marker (D) or amasses the highest likelihood (C). As in the other simulations we will mention, 1,000 trials were performed for each of three student's states of knowledge containing respectively 5, 11 and 15 questions. The results are given in Table 1.6. It must be recalled that the knowledge space was on

Table 1.6. Average number of questions before isolating a single knowledge state.

Model	D	C
Student mastering 5 questions	8.498	9.000
Student mastering 11 questions	8.122	8.000
Student mastering 15 questions	8.234	9.000

21 questions; thus the assessment procedure needs to ask only a small fraction of the questions to isolate the correct knowledge state.

Other simulations were done in case the probabilities of lucky guesses and careless errors are positive. We just report here about the situation in the fair case (no lucky guess), after 12 answers collected. Table 1.7 shows the symmetric difference distance (in the average) between the correct state of knowledge and the knowledge state remaining in the marker (D) or having highest likelihood (C). (In the continuous case, if more than one knowledge state has highest likelihood, the average distance was used.) The lucky guess parameter β_q was set to a constant value β in each of three simulations, namely $\beta = 0.05$, $\beta = 0.10$, and $\beta = 0.20$. These results show that after only about

Table 1.7. Average distance between the correct knowledge state and the remaining knowledge state.

Model	D	C	D	C	D	C
β	0.05	0.05	0.10	0.10	0.20	0.20
5 q.	0.126	0.122	0.288	0. 238	0.568	0.558
11 q.	0.312	0.224	0.664	0.428	1.410	1.140
15 q.	0.664	0.221	1.270	0.533	2.710	1.730

half of all the questions are asked, both processes give quite reliable results.

A deeper study of the continuous model is to be found in Villano (1991). This work relies on the huge data set consisting of the answers of 80,722 students to the New York test on high-school mathematics. It first addresses the problem of the comparison of knowledge spaces built from five different experts or from empirical data. Then the reliability of these spaces for the assessment process is investigated through various simulations. After a few answers have been collected from a 'student', the model can predict his or her answers to remaining questions. It turns out that the correctness of these predictions are rather high. We will not give more details because this would require a more detailed exposition of the continuous model.

1.18 Conclusions

The aim of this chapter was to expose a theory of knowledge assessment. As was said in the Introduction (Section 1.1), interactive, computer-assisted courses should embody a module for assessing efficiently the actual knowledge of the trainee. In order to design the required software, a theoretical model was first needed. This meant not only a descriptive setting, but also the framing of assessment processes taking into account the inherent noise in the answers collected from the student.

First, the fairly general notion of a knowledge structure is defined in set-theoretic terms. A more constraining definition quickly appears to be necessary. On the basis of a mathematical link with the AND/OR graphs of artificial intelligence (recast here as surmise systems), knowledge spaces are introduced. Well-gradedness captures an interesting feature of a subclass of knowledge spaces. Pursuing along the deterministic point of view, we have described processes for knowledge assessment in the form of binary decision trees.

A major change in the framework is required to leave room for possible inconsistencies in the student's answers: probabilistic notions enter the pictures. Thus, a Markov chain model is devised (the reader may be aware of the use of Markov chains in learning theory). It allows a mathematical study of convergence, and also computer-simulations.

We think that all this preliminary theoretical work (pertaining to combinatorics, probability theory, and computer science) was necessary before starting the development of a well-founded software. Some other aspects have not been touched upon in this expository chapter. There are statistical issues, related to how our model should be tested on data; see Falmagne (1989a). The problem of building a particular knowledge space from data consisting in answers from a population of students to a test was also considered elsewhere (Villano, 1991; and Theuns, 1992). Another approach is to extract the relevant information from experts in the field. It led to more combinatorial investigations (Koppen & Doignon, 1990; Koppen, in press; and independently, Müller, 1989). The resulting algorithm minimizes the number of queries to the expert. It was used in a real-life application (Kambouri, 1991; see also Kambouri, Koppen, Villano, & Falmagne, 1991, or Villano, 1991): five experts were queried about the knowledge assessed by the New York test in high-school mathematics. In that way, five knowledge spaces on the same 50 questions were built. (Studies have been undertaken on how to aggregate the resulting spaces.) Remarkably enough, among the huge number of subsets (in this case, 2^{50}, that is a 15 digit number), the algorithm retained only a few thousands as possible knowledge states. Indeed, the figures vary with the experts, but do not exceed 3,903 for four of them. Although the fifth expert did not complete the task, only 7,932 subsets were remaining when she left. More recent works aim at constructing not only knowledge states, but also estimators for their probabilities, or for the probabilities of learning paths (Falmagne, 1989a; Falmagne, 1992; Falmagne, in press).

The final goal of the project, that is a general software for assessment routines based on our models, is not yet reached; we have mentioned in Section 1.17 that preliminary versions were used for computer simulations. Let us now propose some general reflections about the theoretical, preliminary steps of our work.

A question that could have bothered our readers abruptly formulates: why do we use such a mathematical machinery for investigating a psychological topic? The answer is multiple. First, we need to make precise statements, not only to ease communication, but also to put forth assertions whose correctness can be precisely evaluated (either by formal arguments, or by experiments, computer simulations, ...). As Henri Poincaré once said,

> Mathematics is a language in which one cannot express imprecise or nebulous thoughts.

Second, mathematical proofs are surely more convincing than vague arguments (at least for the kind of affirmation that we make, e. g. about the convergence of a process).

Third, the models of knowledge representation and assessment we introduced are testable, either in the sense of a scientific theory, or in a more narrow, technical sense: statistical techniques apply to obtain a measure of confidence of a particular knowledge space. Finally, and once again, these models allow an easy and faithful computer implementation.

A future investigation in the follow up of our project will be to compare the efficiency of our processes with the way human examiners act. We hope to benefit on both sides from this comparison: by incorporating teachers' attitudes and tricks in our software, and by getting a deeper understanding of the way teachers assess the knowledge of their students.

Bibliography

References on knowledge structures and assessment:

Degreef, E., Doignon, J.-P., Ducamp, A., & Falmagne, J.-C. (1986). Languages for the assessment of knowledge. *Journal of Mathematical Psychology*, **30**, 243–256.

Doignon, J.-P., & Falmagne, J.-C. (1985). Spaces for the assessment of knowledge. *International Journal of Man-Machine Studies*, **23**, 175–196.

Dowling, C. E. (1991a). Constructing knowledge spaces from judgements with differing degrees of certainty. In J.-P. Doignon & J.-C. Falmagne (Eds.), *Mathematical psychology: Current Developments* (pp. 221–231). New York: Springer.

Dowling, C. E. (1991b). *Constructing Knowledge Structures from the Judgements of Experts*. Habilitationsschrift, Technische Universität Carolo-Wilhelmina, Braunschweig, Germany.

Falmagne, J.-C. (1989a). A latent trait theory via stochastic learning theory for a knowledge space. *Psychometrika*, **53**, 283–303.

Falmagne, J.-C. (in press,a). Finite Markov learning models for knowledge structures. In G. H. Fischer & D. Laming (Eds.), *Contributions to Mathematical Psychology, Psychometrics, and Methodology*. New York: Springer. To appear.

Falmagne, J.-C. (in press,b). Stochastic learning paths in a knowledge structure. *Journal of Mathematical Psychology.*

Falmagne, J.-C., & Doignon, J.-P. (1988a). A class of stochastic procedures for the assessment of knowledge. *British Journal of Mathematical and Statistical Psychology*, **41**, 1–23.

Falmagne, J.-C., & Doignon, J.-P. (1988b). A Markovian procedure for assessing the state of a system. *Journal of Mathematical Psychology*, **32**, 232–258.

Falmagne, J.-C., Koppen, M., Villano, M., Doignon, J.-P., & Johannesen, L. (1990). Introduction to knowledge spaces: How to build, test and search them. *Psychological Review*, **97**, 201–224.

Kambouri, M. (1991). *Knowledge assessment: A comparison between human experts and computerized procedure.* Doctoral dissertation, New York University.

Kambouri, M., Koppen, M., Villano, M., & Falmagne, J.-C. (in press). Knowledge assessment: tapping human expertise by the QUERY routine. *International Journal of Man-Machine Studies.*

Koppen, M. (1991). *On alternative representations for knowledge spaces* Manuscript submitted for publication.

Koppen, M. (1993). Extracting human expertise for constructing knowledge spaces: An algorithm. *Journal of Mathematical Psychology*, **37**, 1–20.

Koppen, M., & Doignon, J.-P. (1990). How to build a knowledge space by querying an expert. *Journal of Mathematical Psychology*, **34**, 311–331.

Müller, C. E. (1989). A procedure for facilitating an expert's judgments on a set of rules. In E. E. Roskam (Ed.), *Mathematical psychology in progress* (pp. 157–170). Berlin: Springer.

Theuns, P. (1992). *Dichotomization methods in Boolean analysis of co-occurrence data.* Doctoral dissertation, Vrije Universiteit Brussel, Brussels, Belgium.

Villano, M. (1991). Computerized knowledge assessment: Building the knowledge structure and calibrating the assessment routine (Doctoral dissertation, New York University, New York). *Dissertation Abstracts International*, **552**, 12B.

Villano, M., Falmagne, J.-C., Johannesen, L., & Doignon, J.-P. (1987). Stochastic procedures for assessing an individual's state of knowledge. *Proceedings of the International Conference on Computer-assisted Learning in Post-Secondary Education, Calgary 1987* (pp. 369–371). Calgary: University of Calgary Press.

Other references for combinatorial results:

Birkhoff, G. (1937). Rings of sets. *Duke Mathematical Journal*, **3**, 443–454.

Hyafil, L., & Rivest, R. L. (1976). Constructing optimal decision trees is NP-complete. *Information Processing Letters*, **5**, 15–17.

Manuals on Markov chains:

Chung, K. L. (1974). *Elementary probability theory with stochastic processes.* New York: Springer.

Feller, W. (1970). *An introduction to probability theory and its applications.* New York: Wiley.

Kemeny, J. G., & Snell, J. L. (1965). *Finite Markov chains.* Princeton: Van Norstrand.

Manuals on artificial intelligence:

Barr, A., & Feigenbaum, E. A. (Eds.). (1981). *The handbook of artificial intelligence.* London: Pitman.

Rich, E. (1983). *Artificial intelligence.* Singapore: McGraw-Hill.

Further reading:

Albert, D., Schrepp, M., & Held, Th. (1992). Construction of knowledge spaces for problem solving in chess – two experimental investigations. *Bericht aus dem Psychologischen Institut der Universität Heidelberg*, 1–39.

Bloom, C., Villano, M., & VanLehn, K. (1992). *Application of artificial intelligence technologies to training systems: Computer-based diagnostic testing systems* (Contract No. F41624–91–C–5002). Brooks AFB, TX: Technical Training Research Division, Human Resources Directorate.

Doignon, J.-P. (in press). Knowledge spaces and skill assignments. In G. H. Fischer & D. Laming (Eds.), *Contributions to Mathematical Psychology, Psychometrics, and Methodology.* New York: Springer. To appear.

Doignon, J.-P., & Falmagne, J.-C. (1987). Knowledge assessment: A set theoretical framework. In B. Ganter, R. Wille, & K. E. Wolfe (Eds.), *Beiträge zur Begriffsanalyse, Vorträge der Arbeitstagung Begriffsanalyse, Darmstadt 1986* (pp. 129–140). Mannheim: B. I. Wissenschaftsverlag.

Doignon, J.-P., & Falmagne, J.-C. (1988). Parametrization of knowledge structures. *Discrete Applied Mathematics*, **21**, 87–100.

Dowling, C. E. (1993a). On the irredundant construction of knowledge spaces. *Journal of Mathematical Psychology*, **37**, 21–48.

Dowling, C. E. (1993b). Applying the basis of a knowledge space for controlling the questioning of an expert. *Journal of Mathematical Psychology*, **37**, 49–62.

Dowling, C. E. (in press). Integrating Different Knowledge Spaces. In G. H. Fischer & D. Laming (Eds.), *Contributions to Mathematical Psychology, Psychometrics, and Methodology.* New York. Springer. To appear.

Dowling, C. E., & Malinowski, U. (in preparation). Determining knowledge structures for a CAD tutorial.

Falmagne, J.-C. (1989b). Probabilistic knowledge spaces: A review. In F. S. Roberts (Ed.), *Applications of Combinatorics and Graph Theory to the Biological and Social Sciences* (IMA Vol. 17, pp. 283–303). New York: Springer.

Falmagne, J.-C. & Doignon, J.-P. (1993). A stochastic theory for system failure assessment. In B. Bouchon-Meunier, L. Valverde & R. R. Yager (Eds.), *Uncertainty in Intelligent Systems*. Amsterdam: North-Holland.

Koppen, M. (1989). *Ordinal Data Analysis: Biorder Representation and Knowledge Spaces*. Doctoral Dissertation, Katholieke Universiteit te Nijmegen, Nijmegen, The Netherlands.

Koppen, M. (in press). The construction of knowledge spaces by querying experts. In G. H. Fischer & D. Laming (Eds.), *Contributions to Mathematical Psychology, Psychometrics, and Methodology*. New York: Springer. To appear.

Lukas, J., & Albert, D. (1993). Knowledge assessment based on skill assignment and psychological task analysis. In G. Strube & K. F. Wender (Eds.), *The Cognitive Psychology of Knowledge* (pp. 139–159). Amsterdam: Elsevier.

Unnewehr, J. (1992). Benutzerhandbuch Prozeduren zur Wissendiagnose. *Bericht aus dem Psychologischen Institut der Universität Heidelberg* (pp. 1–39+5).

Unnewehr, J. (1993). Knowledge Assessment Procedures 2.0. *Arbeitsbericht aus dem Projekt "Wissensstruktur"* (pp. 1–7+2).

Villano, M. (1992). Probabilistic student models: Bayesian belief networks and knowledge space theory. *Proceedings of the Second International Conference on Intelligent Tutoring Systems* (491–498). New York, Springer, Lecture Notes in Computer Science.

Villano, M., & Bloom, C. (1992). *Probabilistic Student Modelilng with Knowledge Space Theory*. (Contract No. F33615-01-C-0002). Brooks AFB, TX: Technical Training Research Division Human Resources Directorate.

2 Combinatorial structures for the representation of knowledge

Cornelia E. Dowling
Technische Universität Braunschweig, Institut für Psychologie, Postfach 3329,
D–3300 Braunschweig, Germany
E-mail: i3160501@dbstu1.rz.tu-bs.de

2.1 Introduction

In schools and universities, student knowledge in some fields is often assessed by presenting the same small set of questions to each student in the class. A student's performance is then evaluated by assigning a score or grade. This procedure is too crude for the assessment of knowledge as required by a system for computer assisted instruction. Such a system needs information on which topics still have to be taught to an individual student, and which subjects can be omitted. Students may otherwise become either frustrated or bored. Doignon and Falmagne (1985) therefore suggest combining instructional systems with a procedure for the assessment of knowledge which models an oral examination, where the teacher's questioning strategy depends upon the student's answers to previous questions. Using such a questioning procedure the teacher will be able to diagnose a student's knowledge, often after having posed only a few questions to the student.

To create a computer aided assessment procedure modeling a teacher's way of selecting questions in an oral examination, we need a large pool of differentiated questions from the field of knowledge to be examined. To an individual student, we will, however, pose only a few, but highly pertinent questions. Those questions a student is not asked during the assessment procedure are successively to be determined on the basis of the student's answers to previous questions.

The information concerning which of the questions are to be omitted during the assessment procedure may be derived from an expert's judgments. (The expert may, for example, be a teacher experienced in the relevant field of knowledge.) The expert's judgments are so that statements of the following form are selected: "if a student does not master all questions of this given subset of questions, then it can be assumed that he or she does also not master this specific question."

In some applications we work with a set of skills rather than with a set of examination questions. The following example refers to one of these applications. It is designed to demonstrate the nature of the teachers' judgments, and will be referred to repeatedly in this chapter for other illustrations.

Example 2.1.1 In a continuing project a particularly well trained and experienced teacher named 48 skills that she considered to be important for the diagnosis and training of elementary reading and writing abilities. For simplicity, we only list the following four skills (a) to (d), namely,

(a) being able to pick out a printed character from a collection of similar characters,
(b) being able to write optically similar letters correctly,
(c) being able to identify similarly sounding initial phonemes,
(d) being able to write characters corresponding to similar phonemes correctly.

Another experienced teacher chose the following two statements on the four skills a, b, c, and d as being correct.

(1) *If a pupil cannot master skill (a), then she or he cannot master skill (b).*
(2) *If a pupil cannot master the skills (b) and (c), then she or he cannot master skill (d).*

The figure 2.1 below illustrates these two judgments. The "and" in figure 2.1 corresponds to "... (b) and (c)..." in statement (2). □

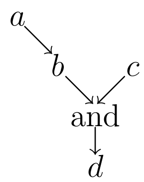


Figure 2.1. Graph illustrating the judgments from example 2.1.1

The expert's task of selecting the appropriate statements is not a trivial matter whenever the collection of questions is large. A major problem that could affect the consistency and the validity of the expert's judgments is the exponentially large number of possible judgments. For example, if only 50 questions are considered, then we have approximately $2.8 \cdot 10^{16}$ statements! There are, however, procedures available facilitating the expert's judgments (Dowling, in press; Koppen, in press; Koppen & Doignon, 1990; Müller, 1989). One of the ideas upon which such a procedure is based is to omit judgments on statements which can be inferred from previous judgments. An introduction to such a procedure will be given in the last section of this chapter. Such procedures do, of course, avoid iterating through the set of all possible statements.

An expert's judgments as described above can be considered as knowledge about students' knowledge, briefly called *meta-knowledge* from now on. In the following sections, we will introduce various formal structures for the representation of meta-knowledge, and we will investigate the relationship between these structures. We will compare structures differing in their psychological interpretation by proving their formal equivalence. For example, we will show in section 2.5 that a collection of an expert's judgments and those judgments which follow from the expert's judgments is equivalent to a specific collection of *knowledge states.* A knowledge state is defined by Doignon and Falmagne (1985) as a subset of questions a student is capable of solving.

By considering different but formally equivalent structures for the representation of meta-knowledge, we are able to choose that structure from a set of equivalent structures, which is most suitable for the use of algorithms required by an application. One of these structures introduced in the section 2.4 will turn out to be a useful basis for applying algorithms to assess students' knowledge. These assessment procedures are developed by Falmagne and Doignon (1988a, 1988b), and by Degreef, Doignon, Ducamp and Falmagne (1986). A survey of some of the probabilistic algorithms for the assessment of student's knowledge is presented in Jean-Paul Doignon's chapter in this volume.

2.2 Representing judgments with a relation

Let V be a fixed and finite set, and let $p_1, \cdots, p_k, q \in V$. In our application, V represents a collection of questions or skills from some knowledge domain. We assume that an expert has selected a set of statements each of which has the form

(2.1) if p_1 and $\cdots$ and p_k are not mastered, then assume q to be unknown.

In this section, we will represent a set of such statements as a relation. This kind of representation will, on the one hand turn out to be straight forward, and on the other, enable us to establish a connection with less obvious structures for the representation of meta-knowledge in the section 2.5.

DEFINITION 2.2.1 The *Cartesian product* $X \times Y$ of two sets X and Y is the set of all *ordered* pairs (x, y) with $x \in X$ and $y \in Y$. A subset $R \subseteq X \times Y$ is called a *binary relation* on X to Y. Then $(x, y) \in R$ is also written as xRy, and is read: "x stands in relation R to y". □

EXAMPLE 2.2.1 Let $X = \{a, b, c\}$ and $Y = \{d, e\}$. The Cartesian product of X and Y is

$$X \times Y = \{(a, d), (a, e), (b, d), (b, e), (c, d), (c, e)\}.$$

Examples of binary relations on X to Y are:

$$R = \{(a, d),\ (a, e),\ (b, d),\ (c, e)\}, \quad R = \emptyset, \quad \text{and } R = X \times Y. \qquad \square$$

With the following definition we introduce the specific binary relation which shall be used to represent a set of statements of the form (2.1).

DEFINITION 2.2.2 Let V be a set. The set of all subsets of the set V is written as 2^V. The set of all subsets of V without the family $\{\emptyset\}$ containing only the empty set is denoted by $2^V \setminus \emptyset$. A binary relation on $2^V \setminus \emptyset$ to V is called an *implication relation* on V. For $P \in 2^V \setminus \emptyset$, for $q \in V$, and for an implication relation,

$$I \subseteq (2^V \setminus \emptyset) \times V,$$

$(P, q) \in I$ is also written as PIq, and is read: "P implies q". □

We have introduced an implication relation $I \subseteq (2^V \setminus \emptyset) \times V$ with the following interpretation for the notation PIq: The question q can be assumed to be unknown whenever all the questions in P are not mastered. Thus, we can identify each statement of the form (2.1) with a member (P, q) of the Cartesian product $(2^V \setminus \emptyset) \times V$, and each subset of such statements as an implication relation. For a member $(P, q) \in I$ we call the set P the *premise* and the element q the consequence.

Example 2.2.2 Let $V = \{a, b, c, d\}$ be the set of questions from our introductory example 2.1.1. The expert's judgments (1) and (2) of this example can be represented as the implication relation

$$I = \{(\{a\},\ b),\quad (\{b, c\},\ d)\}.$$ □

The members of the implication relation $\mathbf{T}$ on V defined by setting

$$P\mathbf{T}q \text{ if and only if } q \in P \tag{2.2}$$

are called tautologies. For the set V from example 2.2.2 the statements $(\{a, b, c, \}, a)$ and $(\{b, d\}, d)$ are examples of tautologies. It is, of course, not necessary to ask an expert for his judgments on the members of the relation $\mathbf{T}$ since these statements must be assumed to be correct.

Exercise 2.2.1 Give the smallest and the largest implication relation $I \subseteq (2^V \setminus \emptyset) \times V$. □

Exercise 2.2.2 Doignon and Falmagne (1985) have introduced a "surmise" relation $S \subseteq V \times V$, with the following interpretation for the notation pSq: on observing a correct response to the question q, it can be surmised that a correct response would also be given to the question p. What is the relationship between a surmise relation S on a set V of questions and an implication relation on V? □

2.3 Representing of judgments by knowledge and failure spaces

In the introduction, we mentioned that algorithms for assessing students' knowledge in a specified field have been developed by Falmagne and Doignon (1988a, 1988b). These algorithms operate on a collection of "knowledge states". Doignon and Falmagne (1985) define a student's *knowledge state* as the subset of questions the student is capable of solving. In this section, we describe that collection of knowledge states which is compatible with an expert's judgments encoded as an implication relation.

Definition 2.3.1 Let 2^V be the set of all subsets of a set V, and let $\mathcal{F}$ be a family of subsets of V, i. e., $\mathcal{F} \subseteq 2^V$, and let $I \subseteq (2^V \setminus \emptyset) \times V$ be an implication relation on V. A subset X of V is called a *failure state* if and only if the following condition (2.3) is fulfilled for X:

$$\text{for all } (P, q) \in I, \text{ if } P \subseteq X, \text{ then } q \in X. \tag{2.3}$$

The family of all failure states is called the *failure space* $\mathcal{F}_I$ derived from I. □

Example 2.3.1 Let $V = \{a, b, c, d\}$ be the set of questions from the introductory example 2.1.1, and let

$$I = \{(\{a\},\ b),\ \ (\{b, c\},\ d)\}$$

be the implication relation from example 2.2.2 representing the statements (1) and (2) of the introductory example 2.1.1. Using the definition 2.3.1 we obtain the failure space derived from I,

$$\mathcal{F}_I = \{\emptyset, \{b\}, \{c\}, \{d\}, \{a, b\}, \{b, d\}, \{c, d\}, \{a, b, d\}, \{b, c, d\}, \{a, b, c, d\}\}.$$

For all members $X \in \mathcal{F}_I$ the condition (2.3) is fulfilled. Examples for which the condition (2.3) does not hold are the set $\{a, c, d\}$, since $P = \{a\} \subseteq \{a, c, d\}$ and $q = b \notin \{a, c, d\}$, the set $\{a, b, c\}$, since $P = \{b, c\} \subseteq \{a, b, c\}$ and $q = d \notin \{a, b, c\}$, and the set $\{a, c\}$, since $P = \{a\} \subseteq \{a, c\}$ and $q = b \notin \{a, c\}$. Condition (2.3) is, of course, not fulfilled for the set $X = \{a\}$ or the set $X = \{b, c\}$. □

By condition (2.3), a failure state can be determined as a subset of V which, for all $(P, q) \in I$, contains the inference q whenever the premise P is a subset of X. In our application a failure state can therefore be interpreted as a *subset of incorrectly answered questions*, which is *compatible* with the expert's statements encoded as an implication relation.

The next definition introduces two concepts which will be used frequently in the remainder of this chapter.

Definition 2.3.2 Let V be a set and let $\mathcal{S}$ be a family of subsets of V so that $\{X_1, \ldots, X_i, \ldots, X_n\} = \mathcal{S}$. The *intersection* of the members of $\mathcal{S}$ is written as $\bigcap \mathcal{S}$, the *union* of the members of $\mathcal{S}$ is written as $\bigcup \mathcal{S}$. The sets $\bigcap \mathcal{S}$ and $\bigcup \mathcal{S}$ are defined as

$$\bigcap \mathcal{S} = X_1 \cap \cdots \cap X_i \cap \cdots \cap X_n \text{ and } \bigcup \mathcal{S} = X_1 \cup \cdots \cup X_i \cup \cdots \cup X_n$$

A family $\mathcal{F} \subseteq 2^V$ is called *closed under intersection*, if, for all subsets $\mathcal{S} \subseteq \mathcal{F}$, the intersection $\bigcap \mathcal{S} \in \mathcal{F}$. Correspondingly, a family $\mathcal{F} \subseteq 2^V$ is called *closed under union*, if $\bigcup \mathcal{S} \in \mathcal{F}$ for all subsets $\mathcal{S} \subseteq \mathcal{F}$. □

Proposition 2.3.1 *The failure space, $\mathcal{F}_I$, derived from an implication relation $I \subseteq (2^V \setminus \emptyset) \times V$, is closed under intersection. The empty set and the total set V are elements of $\mathcal{F}_I$.* □

Proof. Suppose $(P, q) \in I$, and let $\mathcal{F}_{(P,q)} = \{X \subseteq V \mid P \nsubseteq X\} \cup \{X \subseteq V \mid q \in X\}$. With the definition of a failure space,

$$\mathcal{F}_I = \bigcap\{\mathcal{F}_{(P,q)} \mid (P, q) \in I\}.$$

We prove that $\mathcal{F}_{(P,q)}$ is closed under intersection. Suppose $\mathcal{S} \subseteq \mathcal{F}_{(P,q)}$. If $P \nsubseteq X$ for some $X \in \mathcal{S}$, then $P \nsubseteq \bigcap\mathcal{S}$, and $\bigcap\mathcal{S} \in \mathcal{F}_{(P,q)}$. Hence, suppose $P \subseteq X$ for all $X \in \mathcal{S}$. From the definition of $\mathcal{F}_{(P,q)}$, we have $q \in X$ whenever $P \subseteq X$; i.e., $q \in \bigcap\mathcal{S}$, and hence $\bigcap\mathcal{S} \in \mathcal{F}_{(P,q)}$. Since all families $\mathcal{F}_{(P,q)}$ are closed under intersection, the family $\mathcal{F}_I$ is also closed under intersection. Since $P \neq \emptyset$, it follows that $\emptyset \in \mathcal{F}_{(P,q)}$; and since $q \in V$, one obtains $V \in \mathcal{F}_{(P,q)}$ for all $(P, q) \in I$. ■

Since a failure space $\mathcal{F}_I$ is closed under intersection, we can assign to each subset Y of the set V the smallest member $Y^\bullet$ of $\mathcal{F}_I$ containing Y by the following equation:

$$Y^\bullet = \bigcap\{X \in \mathcal{F}_I \mid Y \subseteq X\}. \tag{2.4}$$

The proposition 2.3.1 can be applied to assess students' knowledge by assigning to any student, who has answered a subset Y of the set V of questions incorrectly, the temporary failure state $Y^\bullet$. The student's temporary failure state $Y^\bullet$ contains the questions in Y, which the student has answered incorrectly, together with those that are implied by the statements represented by the implication relation I. If the student answers all questions in $V \setminus Y^\bullet$ correctly, then the temporary failure state becomes the student's final state. If some questions in $V \setminus Y^\bullet$ are answered incorrectly, then the student's final failure state is some subset $Z^\bullet$ of V with the property

$$Y \subseteq Y^\bullet \subseteq Z^\bullet \text{ and } Y \subseteq Z$$

for $Z^\bullet = \bigcap\{X \in \mathcal{F}_I \mid Z \subseteq X\}$. In this manner the temporary failure state of a student can be regarded as a *minimal* set of questions the student is *incapable of solving*. A student's final failure state can be interpreted as the set of questions the student is incapable of solving. For a final failure state $Y \subseteq V$ we can interpret the set complement $\bar{Y} = V \setminus Y$ as the set of questions a student is *capable of solving*, that is, as a knowledge state in the sense of Doignon and Falmagne (1985).

Since the failure space $\mathcal{F}_I$ is closed under intersection by proposition 2.3.1, it follows that the family of the complements $\bar{Y} = V \setminus Y$ of the members Y of a failure space is closed under union.

Definition 2.3.3 A family $\mathcal{K}$ of subsets of a set V, which contains the empty set and the set V, and is closed under union, is called a *knowledge space*. For an implication relation I, a failure space $\mathcal{F}_I$, and $\bar{X} = V \setminus X$, the set

$$\mathcal{K}_I = \{\bar{X} \mid X \in \mathcal{F}_I\}$$

is called the *knowledge space derived from the relation I*. □

EXAMPLE 2.3.2 Let $\mathcal{F}_I$ be the failure space derived from the implication relation $I = \{(\{a\},\ b),\ \ (\{b,c\},\ d)\}$ in example 2.3.1. The knowledge space $\mathcal{K}_I$ is the set of the complements of the members of the failure space $\mathcal{F}_I$,

$$\mathcal{F}_I = \{\emptyset, \{b\}, \{c\}, \{d\}, \{a,b\}, \{b,d\}, \{c,d\}, \{a,b,d\}, \{b,c,d\}, \{a,b,c,d\}\},$$

$$\mathcal{K}_I = \{\{a,b,c,d\}, \{a,c,d\}, \{a,b,d\}, \{a,b,c\}, \{c,d\}, \{a,c\}, \{a,b\}, \{c\}, \{a\}, \emptyset\}.$$

□

We have introduced the concept of a failure space derived from an implication relation *before* the concept of a knowledge space since we assume that condition (2.3) can be understood more immediate than a condition defining a knowledge space derived from an implication relation directly. Such a condition is formulated as follows:

PROPOSITION 2.3.2 *Let I be an implication relation on V, and let Y be a subset of V which fulfills the property*

$$\text{(2.5)} \qquad \textit{for all } (P,q) \in I,\ \textit{if } q \in Y,\ \textit{then } P \cap Y \neq \emptyset.$$

The family of subsets Y fulfilling property (2.5) *is equal to the knowledge space $\mathcal{K}_I$ derived from I.* □

Note that, for a given implication relation I, property (2.5) holds for a set Y if and only if property (2.3) holds for the set $X = \bar{Y}$.

The idea of applying knowledge or failure spaces to the assessment of students' knowledge originates from Doignon and Falmagne (1985). The authors derive knowledge spaces from a representation of AND/OR graphs (Nilsson, 1971) called *surmise systems*, which are introduced in J.-P. Doignon's chapter in this volume. Koppen and Doignon (1990) derive knowledge spaces from a relation related to the implication relation, which we will refer to in the following chapter.

EXERCISE 2.3.1 Let $\mathcal{F}_I$ be the failure space from the example 2.3.2. Determine the smallest failure state in $\mathcal{F}_I$ containing X for the sets $X = \emptyset,\ X = \{a,c\},$ and $X = \{b,c\}$. □

EXERCISE 2.3.2 Let $\mathcal{K}_I$ be the knowledge space from the example 2.3.2. Determine the largest knowledge state in $\mathcal{K}_I$ contained in X for the sets $X = \{a,b,c,d\},\ X = \{b,d\},$ and $X = \{a,d\}$. □

EXERCISE 2.3.3 Let $V = \{a,b,c,d\}$ be a set and let $I = \{(\{c\},\ d), (\{a,b\},\ c)\}$ be an implication relation.

(a) Determine the failure space $\mathcal{F}_I$ derived from I.

(b) Determine the knowledge space $\mathcal{K}_I$ derived from I. □

EXERCISE 2.3.4 Let $\mathcal{S} \subseteq 2^V$. Determine the sets $\bigcap\mathcal{S}$ and $\bigcup\mathcal{S}$ for $\mathcal{S} = 2^V$ and for $\mathcal{S} = \emptyset$, considering the fact that both $\bigcap\mathcal{S}$ and $\bigcup\mathcal{S}$ can also be defined as

$$x \in \bigcup\mathcal{S} \quad \text{if and only if} \quad \text{there is a } X \in \mathcal{S} \text{ so that } x \in X.$$

$$x \in \bigcap\mathcal{S} \quad \text{if and only if} \quad \text{for all } X \in \mathcal{S},\ X \in \mathcal{S} \text{ implies } x \in X. \qquad \square$$

2.4 Combinatorial Galois connections

In this section, we introduce several concepts for elaborating our understanding of the relationship between implication relations and knowledge spaces. These prerequisites will eventually help us to prove a one-to-one correspondence between certain implication relations and knowledge spaces.

DEFINITION 2.4.1 A binary relation $R \subseteq X \times X$ is called a relation on the set X. The *converse* of a binary relation R on X is the relation R^{-1} defined as

$$yR^{-1}x \quad \text{if and only if} \quad xRy$$

A set X together with a binary relation R on X is written as the pair (X, R). A pair $(X, \leq)$ is a *partial order* if the following three properties are fulfilled for the relation $\leq$ on X:

(i) $x \leq x$ for all $x \in X$ (*reflexive*).
(ii) If $x \leq y$ and $y \leq x$, then $x = y$ for all $x, y \in X$ (*antisymmetric*).
(iii) If $x \leq y$ and $y \leq z$, then $x \leq z$ for all $x, y, z \in X$ (*transitive*).

For a partial order $(X, \leq)$ the converse $\leq^{-1}$ of the relation $\leq$ is written as $\geq$. $\square$

EXAMPLE 2.4.1 Let $\mathcal{F}$ be a family of subsets of a set V ordered by the inclusion relation $\subseteq$. The pair $(\mathcal{F}, \subseteq)$ is a partial order. $\square$

DEFINITION 2.4.2 Let $(X, \leq)$ be a partial order. The function c: $X \to X$ that assigns to each member $x \in X$ a member $c(x) = x^c \in X$ with the following properties,

(i) $x \leq x^c$,
(ii) $x^c = (x^c)^c$,
(iii) $y \leq x$ implies $y^c \leq x^c$.

is called a *closure operator on* $(X, \leq)$. The images x^c are called the *closed* members of the set X. $\square$

The following example demonstrates that the members $Y^\bullet$ of a failure space determined by the equation (2.4) can be considered as closed sets for a closure operator on the partial order $(2^V, \subseteq)$.

EXAMPLE 2.4.2 Let V be a set, let $\mathcal{F}_I$ be a failure space, and consider the partial order $(2^V, \subseteq)$. Then the function

$$\bullet\colon\ 2^V \to 2^V,$$

which assigns to each member $Y \in 2^V$ the smallest member $Y^\bullet \in \mathcal{F}_I$ containing Y is a closure operator on $(2^V, \subseteq)$. The closed members of 2^V are the members of the failure space $\mathcal{F}_I$. In particular, for the set $V = \{a, b, c, d\}$ of questions from our introductory example 2.1.1, and the failure space $\mathcal{F}_I$ determined in the example 2.3.1 we obtain:

$\emptyset^\bullet = \emptyset$, $\{a\}^\bullet = \{a, b\}$, $\{b\}^\bullet = \{b\}$, $\{c\}^\bullet = \{c\}$, $\{d\}^\bullet = \{d\}$,
$\{a, b\}^\bullet = \{a, b\}$, $\{a, c\}^\bullet = \{a, b, c, d\}$, $\{a, d\}^\bullet = \{a, b, d\}$,
$\{b, c\}^\bullet = \{b, c, d\}$, $\{b, d\}^\bullet = \{b, d\}$, $\{c, d\}^\bullet = \{c, d\}$,
$\{a, b, c\}^\bullet = \{a, b, c, d\}$, $\{a, b, d\}^\bullet = \{a, b, d\}$, $\{a, c, d\}^\bullet = \{a, b, c, d\}$,
$\{b, c, d\}^\bullet = \{b, c, d\}$, $\{a, b, c, d\}^\bullet = \{a, b, c, d\}$.

For these subsets of the set V we can see that the conditions (i), (ii), and (iii) of definition 2.4.2 are fulfilled. For all subsets $X, Y \subseteq V$ we have

(i) $X \subseteq X^\bullet$,
(ii) $(X \subseteq X^\bullet)^\bullet$, for example, $\{a, d\}^\bullet = \{a, b, d\} = \{a, b, d\}^\bullet$, and
(iii) $X \subseteq Y$ implies $X^\bullet \subseteq Y^\bullet$. □

The next example illustrates that there are closure operators on partial orders for which the failure spaces and the knowledge spaces can be determined as closed elements.

EXAMPLE 2.4.3 Let $2^{2^V} = \tilde{\mathcal{F}}$ be the set of all families of subsets of a set V. The function

$$i\colon\ \tilde{\mathcal{F}} \to \tilde{\mathcal{F}}$$

that assigns to each family $\mathcal{F} \subseteq 2^V$ the set $\mathcal{F}^i = \{\bigcap \mathcal{S} \mid \mathcal{S} \subseteq \mathcal{F}\}$ is a closure operator on $(\tilde{\mathcal{F}}, \subseteq)$. The closed members of $\tilde{\mathcal{F}}$ are the families of sets, which are closed under intersection and contain the empty set as well as the set V. Analogously, the function

$$u\colon\ \tilde{\mathcal{F}} \to \tilde{\mathcal{F}}$$

that assigns to each family $\mathcal{F} \subseteq 2^V$ the set $\mathcal{F}^u = \{\bigcup \mathcal{S} \mid \mathcal{S} \subseteq \mathcal{F}\}$ is a closure operator on $(\tilde{\mathcal{F}}, \subseteq)$. The closed sets are the families of sets which are closed under union, and contain the empty set as well as the set V. The families $\mathcal{F}^i$ and $\mathcal{F}^u$ are called the closure of the family $\mathcal{F}$ under intersection or under union, respectively. For example, the closure under intersection of the family,

$$\mathcal{F} = \{\{c\}, \{a, b\}, \{c, d\}, \{a, b, d\}, \{b, c, d\}\},$$

is equal to the failure space determined in the example 2.3.1. The closure under union of the family,

$$\mathcal{G} = \{\{a, b, d\}, \{c, d\}, \{a, b\}, \{c\}, \{a\}\},$$

is equal to the knowledge space from the example 2.3.2. Note that $V = \{a, b, c, d\} \in \mathcal{F}^i$ since $\mathcal{S} = \emptyset \subseteq \mathcal{F}$, and, by convention, $\bigcap(\mathcal{S} = \emptyset) = V$. By a

$$\begin{tikzcd}
Z & & X \arrow[ll, "h \circ g"'] \arrow[dl, "g"] \\
& Y \arrow[ul, "h"] &
\end{tikzcd}$$

Figure 2.2. Diagram illustrating the functions from Definition 2.4.3

similar convention we obtain $\emptyset \in \mathcal{G}^u$. (We have just answered the hardest part of exercise 2.3.4! There you can find hints explaining the conventions.) □

In the following, we will introduce a procedure for establishing correspondences between the closed members of two different partially ordered sets. For that purpose we require the notion of the composition of functions.

Definition 2.4.3 Let X, Y, and Z be sets. The *composite* of two given functions

$$g: X \to Y \quad h: Y \to Z$$

is the function $h \circ g : X \to Z$ with values given by $(h \circ g)(x) = h(g(x))$ for all $x \in X$. □

The definition 2.4.3 may be visualized by the diagram 2.2 above.

Definition 2.4.4 Let $(X, \leq_1)$ and $(Y, \leq_2)$ be two partial orders, and let (g, h) be a pair of mappings, $g: X \to Y$, $h: Y \to X$. The pair (g, h) is called a *Galois connection* between X and Y if the following four conditions hold, for all $x, x' \in X$, and for all $y, y' \in Y$:

(i) If $x \leq_1 x'$, then $g(x) \geq_2 g(x')$,
(ii) If $y \leq_2 y'$, then $h(y) \geq_1 h(y')$,
(iii) $x \leq_1 (h \circ g)(x)$,
(iv) $y \leq_2 (g \circ h)(y)$.

The elements, $(h \circ g)(x) \in X$, and the elements, $(g \circ h)(y) \in Y$, are called the *Galois closures* of x and y, respectively. The combined correspondences, $(h \circ g)$ and $(g \circ h)$, are called *Galois closure operators*. □

Example 2.4.4 Suppose (X, R_1) and (Y, R_2) are two partial orders,where

$$X = \{a, b, c, d\},$$

$$R_1 = \{(a, b), (b, c), (c, d), (a, c), (a, d), (b, d), (a, a), (b, b), (c, c), (d, d)\},$$

and

$$Y = \{x, y, z, u\},$$

$$R_2 = \{(x, y), (y, z), (z, u), (x, z), (x, u), (y, u), (x, x), (y, y), (z, z), (u, u)\}.$$

Let g: $X \to Y$, with $g(a) = u$, $g(b) = u$, $g(c) = z$, $g(d) = y$. Let h: $Y \to X$, with $h(u) = b$, $h(z) = c$, $h(y) = d$, $h(x) = d$. The assignments by the functions g and h are illustrated by the diagram 2.3 below substituting $\leq$ for R_1 and $\geq$ for ${R_2}^{-1}$. The Galois closures of the set X are the elements

$$(h \circ g)(a) = (h \circ g)(b) = b, \quad (h \circ g)(c) = c, \quad (h \circ g)(d) = d.$$

The Galois closures of the set Y are the elements u, z, and y. □

The following well known propositions (see, for example, Ore, 1962) will turn out to be useful in the sequel.

PROPOSITION 2.4.1 *Under a Galois connection (g, h) the following two conditions are fulfilled:*

$$\text{(i) } g(h \circ g)(x) = g(x). \qquad \text{(ii) } h(g \circ h)(y) = h(y).$$ □

PROOF. We prove (i). From a combination of the properties (iii) and (i) in the definition 2.4.4, it follows that $g(h \circ g)(x) \leq g(x)$. By the property (iv) from the same definition we obtain that $g(x) \leq g(h \circ g)(x)$. (ii) can be proven analogously. ■

Using the definition 2.4.4 of a Galois connection and the proposition 2.4.1, we obtain the following, immediate result.

PROPOSITION 2.4.2 *The Galois closure operators $(h \circ g)$ and $(g \circ h)$ are closure operators on $(X, \leq)$ and $(Y, \leq)$, respectively.* □

With the proposition 2.4.2 the elements $x^c = (h \circ g)(x)$ and $y^c = (g \circ h)(y)$ are closed members of X and Y, respectively.

PROPOSITION 2.4.3 *The closed elements x^c in X and y^c in Y can be characterized by the statements*

$$x^c = h(y) \text{ for some } y \in Y \qquad \text{and} \qquad y^c = g(x) \text{ for some } x \in X,$$

and the Galois connection defines one-to-one correspondences for the closed elements of the sets X and Y. □

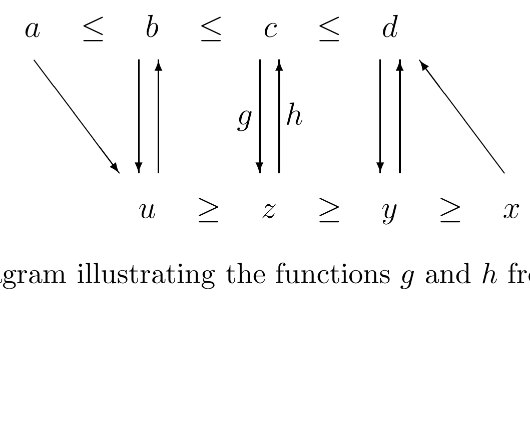


Figure 2.3. Diagram illustrating the functions g and h from example 2.4.4

PROOF. If an element $x^c = h(y)$ for some $y \in Y$, then it is closed according to the proposition 2.4.1. Conversely, if x^c is a closed element, then it is an image under h, namely $x^c = h(g(x))$. Finally, using the proposition 2.4.1 we obtain that each of the relations

$$x^c = h\ (y^c) \quad \text{and} \quad y^c = g\ (x^c)$$

implies the other. ■

EXAMPLE 2.4.5 In the example 2.4.4 the one-to-one correspondence between the closed members of the sets X and Y associates the closed element $b \in X$ with the closed element $u \in Y$, the element $c \in X$ with $z \in Y$, and $d \in X$ with $y \in Y$. □

2.5 The relationship between implication relations and knowledge spaces

In this section, we will establish a Galois connection between implication relations and families of subsets of a finite set. We will prove that there is a one-to-one correspondence between knowledge spaces and those implication relations which contain all members which can be inferred from other members of the implication relation. In the last section of this chapter we show that this relationship between knowledge spaces and implication relations can be applied in procedures for constructing a knowledge space from the judgments of an expert.

We first introduce a theorem by Koppen and Doignon (1990) and by Müller (1989), and subsequently report related results by Birkhoff (1937), Monjardet (1979), by Doignon and Falmagne (1985).

THEOREM 2.5.1 *Let V be a set, let $\tilde{I}$ be the set of implication relations on V, and let $\tilde{\mathcal{F}}$ be the set of all families of subsets of the set V, both ordered by the inclusion relation. Define a mapping, $r\colon \tilde{\mathcal{F}} \to \tilde{I}$ by requiring that*

$$(P,q) \in r(\mathcal{F}) \quad \textit{if and only if} \quad \textit{for every } X \in \mathcal{F},\ \textit{if } P \subseteq X,\ \textit{then } q \in X.$$

Similarly, define a mapping $f\colon \tilde{I} \to \tilde{\mathcal{F}}$ by setting

$$X \in f(I) \quad \textit{if and only if} \quad \textit{for every } (P,q) \in I,\ \textit{if } P \subseteq X,\ \textit{then } q \in X.$$

Then the pair (r, f) is a Galois connection. The Galois closure $(f \circ r)(\mathcal{F})$ is the smallest failure space containing $\mathcal{F}$. □

PROOF. (i) To prove $\mathcal{F} \subseteq \mathcal{F}'$ implies $r(\mathcal{F}) \supseteq r(\mathcal{F}')$, suppose $\mathcal{F} \subseteq \mathcal{F}'$, and let $P \subseteq V$. Since $\mathcal{F} \subseteq \mathcal{F}'$, it follows that $\bigcap\{X \in \mathcal{F} \mid P \subseteq X\} \supseteq \bigcap\{X \in \mathcal{F}' \mid P \subseteq X\}$. Hence, for all $q \in \bigcap\{X \in \mathcal{F}' \mid P \subseteq X\}$, we obtain that $(P,q) \in r(\mathcal{F})$ whenever $(P,q) \in r(\mathcal{F}')$ by definition of the map r.

(ii) To prove $I \subseteq I'$ implies $f(I) \supseteq f(I')$, suppose $I \subseteq I'$. Using the definition of the map f, $f(I) = 2^V \setminus \{X \subseteq V \mid (P,q) \in I \text{ and } P \subseteq X \text{ and } q \notin X\}$, whereas $f(I') = f(I) \setminus \{X \subseteq V \mid (P,q) \in (I' \setminus I) \text{ and } P \subseteq X \text{ and } q \notin X\}$. Hence, $f(I) \supseteq f(I')$.

(iii) To prove $\mathcal{F} \subseteq (f \circ r)(\mathcal{F})$, suppose $Y \in \mathcal{F}$. If $P \subseteq Y$, then $\bigcap\{X \in \mathcal{F} \mid P \subseteq X\} \subseteq Y$. Hence, if $q \in \bigcap\{X \in \mathcal{F} \mid P \subseteq X\}$ and $P \subseteq Y$, then $q \in Y$. By the definition of the map r, it follows that $q \in Y$ whenever $(P, q) \in r(\mathcal{F})$ and $P \subseteq Y$. By the definition of the map f, we have $Y \in (f \circ r)(\mathcal{F})$.

(iv) To prove $I \subseteq (r \circ f)(I)$, suppose $(P, q) \in I$. Using the definition of the map f, we have $q \in X$ for all $X \in f(I)$ with $P \subseteq X$, and hence $q \in \bigcap\{X \in f(I) \mid P \subseteq X\}$. By definition of the map r, we obtain $(P, q) \in (r \circ f)(I)$. ■

Note, that the failure space $f(I)$ is defined so that it is equal to the failure space derived from I in definition 2.3.1.

Corollary 2.5.1 *The Galois connection* (r, f) *induces a one-to-one correspondence between the closed implication relations on* V *and the failure spaces, derived from implication relations on* V. □

The corollary to theorem 2.5.1 follows from the proposition 2.4.3. Since there is a one-to-one correspondence between the knowledge and failure spaces derived from the same implication relation I, the Galois connection (r, f) also induces a one-to-one correspondence between knowledge spaces and implication relations.

Theorem 2.5.1 characterizes the Galois closure $(f \circ r)(\mathcal{F})$, but not the Galois closure $(r \circ f)(I)$. The following proposition by Koppen and Doignon (1990) gives a compact description of the closed implication relations.

Proposition 2.5.1 *Let* I *be an implication relation on* V. *If* $I^\diamond$ *is the smallest implication relation containing* I, *and fulfilling the properties* (i) *and* (ii) *below, then*

$$I^\diamond = (r \circ f)(I).$$

(i) *For each* $P \subseteq Q$, *and each* $p \in P$, *we have* $(P, p) \in I^\diamond$.

(ii) *Let* $O, P \subseteq V$, *and let* $p, q \in V$. *If* $(O, p) \in I^\diamond$ *for all* $p \in P$, *and if* $(P, q) \in A_l^\diamond$, *then it follows that* $(O, q) \in A_l^\diamond$. □

The elements (P, q) of the closed implication relation $(r \circ f)(I)$ can be interpreted as the statements of the form (2.1), which follow logically from the correct statements encoded by the implication relation I. The closed implication relation $(r \circ f)(I)$ contains all statements which can be inferred as being correct.

The smallest closed implication relation on V is the relation $\mathbf{T}$ which is defined by expression (2.2) and represents the set of tautological statements of the form (2.1). To the relation $\mathbf{T}$, the largest closed failure space, 2^V, is assigned by the function f defined in theorem 2.5.1,

$$f(\mathbf{T}) \;=\; 2^V, \text{ and } r(2^V) = \mathbf{T}. \tag{2.6}$$

The smallest closed failure space is $\{\emptyset, T\}$. By the function r, the largest closed implication relation, $(2^V \setminus \emptyset) \times V$, is assigned to the smallest failure space,

$$r(\{\emptyset, V\}) \;=\; (2^V \setminus \emptyset) \times V, \text{ and } f((2^V \setminus \emptyset) \times V) = \{\emptyset, V\}. \tag{2.7}$$

$$\begin{tikzcd}
\mathcal{F} \arrow[d, "f \circ r"'] & I \arrow[d, "r \circ f"] \\
\mathcal{F}_I \arrow[r, shift left, "r"] & I_{\mathcal{F}} \arrow[l, shift left, "f"]
\end{tikzcd}$$

Figure 2.4. The correspondences between the implication relations I and $I_{\mathcal{F}}$ and the families of subsets $\mathcal{F}$ and $\mathcal{F}_I$.

Example 2.5.1 Let $\mathcal{F}$ be the family of subsets of the set $V = \{a, b, c, d\}$ given in the example 2.4.3,

$$\mathcal{F} = \{\{c\}, \{a, b\}, \{c, d\}, \{a, b, d\}, \{b, c, d\}\},$$

let I be the implication relation on V from the example 2.2.2, which represents the expert's statements from the introductory example 2.1.1.

$$I = \{(\{a\},\ b),\quad (\{b, c\},\ d)\}.$$

Let $\mathcal{F}_I$ be the failure space from the example 2.3.1,

$$\mathcal{F}_I = \{\emptyset, \{b\}, \{c\}, \{d\}, \{a, b\}, \{b, d\}, \{c, d\}, \{a, b, d\}, \{b, c, d\}, \{a, b, c, d\}\}.$$

Let $I_{\mathcal{F}}$ be the following implication relation on V:

$$\begin{aligned} I_{\mathcal{F}} \quad = \quad & \{(\{a\},\ b),\quad (\{b, c\},\ d),\quad (\{a, c\},\ b),\quad (\{a, c\},\ d),\quad (\{a, d\},\ b), \\ & (\{a, b, c\},\ d),\quad (\{a, c, d\},\ b)\} \ \cup\ \mathbf{T}, \end{aligned}$$

where $\mathbf{T}$ is defined by (2.2). Using the definition of the function f in the theorem 2.5.1, we obtain that

$$f(I) = \mathcal{F}_I \quad \text{and} \quad f(I_{\mathcal{F}}) = \mathcal{F}_I.$$

From the definition of the function r in the theorem 2.5.1, it follows that

$$r(\mathcal{F}) = I_{\mathcal{F}} \quad \text{and} \quad r(\mathcal{F}_I) = I_{\mathcal{F}}.$$

By proposition 2.4.3, the relation $I_{\mathcal{F}}$ is a closed implication relation, and the family $\mathcal{F}_I$ is a closed family of sets, since they are images under r and f, respectively. The one-to-one correspondence established by the Galois connection (r, f) is illustrated in this example by the fact that the closed relation $I_{\mathcal{F}}$ is assigned to the closed family $\mathcal{F}_I$ by the function r, and, conversely, that $\mathcal{F}_I$ is associated to $I_{\mathcal{F}}$ by the function f. The correspondences between the implication relations I and $I_{\mathcal{F}}$ and the families of subsets $\mathcal{F}$ and $\mathcal{F}_I$ are visualized by figure 2.4. □

In many applications, a set V of questions is presented to a group of students and the students are asked to answer all the questions in V. In this manner the set of questions which are answered incorrectly by a student can be regarded as the student's failure state. The failure states of all students in the group can be represented by a family $\mathcal{F}$ of subsets of the set V. We can thus apply the theorem 2.5.1 to determine the largest implication relation consistent with the student's data and to derive the largest set of failure states which can be inferred from the students' data represented by $\mathcal{F}$.

The theorem 2.5.1 is related to a variety of results on Galois connections between relations and families of sets. Let us consider the special case of an implication relation S on V which is defined as

$$PSq \quad \text{if and only if} \quad |P| = 1,$$

where $|P|$ denotes the *size* or the *cardinality* of P. In that manner we can identify a relation S with a binary relation $S \subseteq V \times V$. (Note that we have given the solution to exercise 2.2.2!) Birkhoff (1937) has proven that there is a one-to-one correspondence between the binary relations on V which are reflexive and transitive, and the families of subsets of V which are closures under union and intersection. Monjardet (1970) has shown that Birkhoff's result can be considered to be an inference of the fact that a Galois connection can be established between the binary relations on a set V and the families of subsets of the set V. Koppen and Doignon (1990) extend Monjardet's work by constructing a Galois connection between the binary relations on the set 2^V and the families of subsets of V. The resulting Galois closures are the families of subsets, which are closed under union, and relations E on 2^V which the authors call "entail" relations. An entail relation E on 2^V is reflexive and transitive.

The idea of generalizing the results of Birkhoff (1937) and Monjardet (1970) and of applying the generalized result to the assessment of knowledge is due to Doignon and Falmagne (1985). Doignon and Falmagne establish a Galois connection between representations of AND/OR graphs called "surmise systems" and families of subsets. As closed families of subsets they obtain the knowledge spaces. The "surmise systems" and the closed "surmise systems" are introduced in Jean-Paul Doignon's chapter in this volume.

EXERCISE 2.5.1 Let V be the set $\{a, b, c, d\}$ and $I = \{(\{c\}, d),\ (\{a, b\}, c)\}$ the implication relation from exercise 2.3.3. Determine the closed implication relation $(r \circ f)(I)$. □

EXERCISE 2.5.2 Let $V = \{a, b, c, d\}$ and $I = \{(\{c\}, c),\ (\{a, b\}, b)\}$ be an implication relation on V.

(a) Determine the failure space $f(I)$.

(b) Determine the closed implication relation $(r \circ f)(I)$. □

EXERCISE 2.5.3 Consider the family $\mathcal{F} = \{\{a, b, c\},\ \{a, b, d\},\ \{a, c, d\},\ \{b, c, d\}\}$ of subsets of the set $V = \{a, b, c, d\}$.

(a) Determine the closed implication relation $r(\mathcal{F})$.

(b) Determine the failure space $(f \circ r)(\mathcal{F})$. □

EXERCISE 2.5.4 Let $I = \{(\{a\}, b),\ (\{b\}, c),\ (\{c\}, d),\ (\{d\}, a)\}$ be an implication relation on $2^V \setminus \emptyset$ to $V = \{a, b, c, d\}$.

(a) Determine the failure space $f(I)$.

(b) Determine the closed implication relation $(r \circ f)(I)$. □

Exercise 2.5.5 Let $\mathcal{F}$ be the family $\{\emptyset\}$ of subsets of a set V. Determine the closed implication relation $r(\mathcal{F})$ (Hint: use the definition of the intersection of a family $\mathcal{S} \subseteq 2^V$ given in the exercise 2.3.4). □

For further reading 2.5.1 For further reading on the topics of closure operators and Galois connections we recommend Ore (1962) and Stanat and McAllister (1977) as introductions. For deeper understanding we recommend Birkhoff (1979), and the articles cited in this section. □

2.6 A procedure facilitating an expert's judgments

A procedure that interactively questions an expert for his judgments on statements of the form (2.1) is introduced in this section. This procedure will allow an expert to omit judgments on statements that logically follow from previous judgments, and to avoid contradictory judgments. The questioning procedure is based on the Galois connection between the implication relations and the families of subsets of the set V established by theorem 2.5.1. Somewhat different variants of such a procedure for querying an expert are given by Dowling (in press), Koppen (in press), Koppen and Doignon (1990), and by Müller (1989).

The procedure for questioning an expert works step wise. At each step l with $l \geq 1$, exactly one statement is presented to the expert. The expert judges, i. e., *accepts* or *rejects* this statement. A new statement presented for judgment at step $l+1$ must fulfill two restrictions which ensure that judgments which follow from previous judgments are not presented to the expert at any given step. These two restrictions will be introduced in the definition 2.6.2 below. The procedure terminates, at step n, say, whenever no new statement fulfills these restrictions.

In the following we will differentiate between three implication relations. An implication relation representing the set of statements of the form (2.1) judged by the expert up to the step l, an implication relation whose members are interpreted as the statements accepted by the expert up to step l, and an implication relation characterizing the statements rejected by the expert up to the step l.

Definition 2.6.1 Let $\emptyset = J_0 \subseteq J_1 \subseteq \cdots \subseteq J_l \subseteq \cdots \subseteq J_n$ be the sequence of implication relations on V judged up to step l. Such a sequence has the property that $|J_{l+1} \backslash J_l| = 1$. For $l > 0$, the member $(P, q) \in J_l \backslash J_{l-1}$ is called the *statement judged at the step l*. Let $\emptyset = A_0 \subseteq A_1 \subseteq \cdots \subseteq A_l \subseteq \cdots \subseteq A_n$ be a sequence of implication relations on V such that $A_l \subseteq J_l$, and either $A_{l+1} = A_l$ or $A_{l+1} \not\subseteq J_l$ for all $0 \leq l \leq n$. If $(P, q) \in A_l \setminus A_{l-1}$, then (P, q) is called the *statement accepted at the step l*.

If $(P, q) \in J_l \setminus (J_{l-1} \cup A_l)$, then (P, q) is called the *statement rejected at the step* l. The set $J_l \setminus A_l$ of statements rejected at the steps $j \leq l$ is denoted by R_l. □

The following definition by Müller (1989) differentiates between sets of statements of the form (2.1) the acceptance of which is implied by the expert's previous judgments, and sets of statements the rejection of which follows from the expert's previous judgments.

DEFINITION 2.6.2 Let r and f be the functions defined in theorem 2.5.1. The closed relation $(r \circ f)(A_l)$ is called the set of the *positive inferences* of the members of A_l. The members of the set

$$R_l^* = \{(O, s) \in (2^V \setminus \emptyset) \times V \mid R_l \cap (r \circ f)(A_l \cup \{(O, s)\}) \neq \emptyset\}$$

are called the *negative inferences* of R_l *with respect to* A_l or, briefly, the *negative inferences* of R_l. The sets A_l and R_l are called *contradictory* if and only if

$$R_l^* \;\cap\; (r \circ f)(A_l) \;\neq\; \emptyset.$$

The implication relations J_l are called *consistent,* if and only if, for all steps $0 \leq l \leq n$, one has whenever $(P, q) \in J_{l+1} \setminus J_l$, then the conditions [a] and [b] below are fulfilled.

[a] $(P, q) \notin (r \circ f)(A_l)$.
[b] $(P, q) \notin R^*$; i. e., there is no $(O, s) \in R_l$ such that $(O, s) \in (r \circ f)(A_l \cup \{(P, q)\})$. □

By theorem 2.5.1, the positive inferences $(r \circ f)(A_l)$ of the set of statements A_l accepted up to step l have been shown to be those members of $(2^V \setminus \emptyset) \times V$ the acceptance of which logically follows from the acceptance of the members of A_l. The negative inferences R_l^* of the statements R_l rejected up to step l are defined to be those members of $(2^V \setminus \emptyset) \times V$ the rejection of which logically follows from the rejection of the members of R_l and from the acceptance of the members of A_l. Hence we have defined a set A_l of accepted statements and a set R_l of rejected statements to be contradictory, if there is a member $(P, q) \in (2^V \setminus \emptyset) \times V$ the rejection as well as the acceptance of which follows from R_l and A_l.

The following proposition by Müller (1989) summarizes the properties of a procedure presenting only members of consistent sets J_l to an expert for judgment.

PROPOSITION 2.6.1 *If the implication relations* J_l, *with* $0 \leq l \leq m$, *are consistent, then the conditions* (i), (ii), (iii), *and* (iv) *below are fulfilled:*

(i) *No statement accepted at a step* l *is an inference of a set of statement accepted at steps* $0 \leq j < l$.
(ii) *No statement rejected at a step* l *is an inference of a set of statements accepted at steps* $0 \leq j < l$.
(iii) *No statement rejected at a step* l *is an inference of a set of statements accepted at steps* $m \geq j > l$.

(iv) *The set of statements accepted at the steps $j \leq l$ and the set of statements rejected at the steps $j \leq l$ are not contradictory for all steps $0 \leq l \leq m$* $\square$

An interactive procedure presenting only those statements to an expert for judgment which fulfill the conditions [a] and [b] of definition 2.6.2 omits all judgments which are redundant in the sense that a statement would be judged which is already a positive inference of statements accepted at previous steps, or a negative inference of statements rejected at previous steps. An interactive questioning procedure restricted by the conditions [a] and [b] is so that the sets of accepted statements and the sets of rejected statements are not contradictory at all steps.

A first way of testing the conditions [a] and [b] of the definition 2.6.2 is to store and to update the set $(r \circ f)(A_l)$ of positive inferences and the set R_l^* of negative inferences at each step l. Then we can generate a new statement (P, q) at step $l+1$, and test if the restrictions [a] and [b] of the definition 2.6.2 are fulfilled. If both conditions hold, then the statement (P, q) is presented to the expert for judgment. If these restrictions are not fulfilled, then a new statement (P', q') is generated, we check if the restrictions [a] and [b] hold for the new statement (P', q') and so on until all members $(P, q) \in (2^V \times V) \setminus \emptyset$ are judged or inferred.

The updating of the implication relations $(r \circ f)(A_l)$ and R_l^* at each step l is not a trivial task, particularly when the number of inferences is large. There are algorithms available, however, that compute the new positive and negative inferences at each step $l+1$, and add these inferences to the old sets $(r \circ f)(A_l)$ and R_l^*. Koppen and Doignon (1990) have developed iterative updating rules which have been applied in practice. Different, non-iterative updating rules have been suggested by Dowling (1991).

Storing the complete sets $(r \circ f)(A_l)$ and R_l^* until all possible statements (P, q) are judged or inferred is not feasible for larger sets V. From the introduction, we know that we would have to store up to approximately $2.8 \cdot 10^{16}$ statements if only 50 questions are considered. In this predicament, we obtain help from theorem 2.5.1, i. e., from the fact that implication relations and families of subsets of a finite set are related by a Galois connection. This enables us to replace each closed set $(r \circ f)(A_l)$ of accepted assertions by an equivalent structure, the failure space $f(A_l)$, or the corresponding knowledge space. This replacement is feasible since the sizes of the sets $(r \circ f)(A_l)$ and $f(A_l)$ are inversely related, the larger the size of the set $(r \circ f)(A_l)$ of positive inferences, the smaller the size of the failure space $f(A_l)$. As soon as the set $(r \circ f)(A_l)$ becomes too large, we can replace it by the failure space $f(A_l)$. From theorem 2.5.1, it follows that $(P, q) \notin (r \circ f)(A_l)$ if and only if

$$q \notin P^\bullet = \bigcap\{X \in f(A_l) \mid P \subseteq X\}); \tag{2.8}$$

that is, condition (2.8) can be tested instead of the restriction [a] of definition 2.6.2.

Conditions for replacing the condition [b] of definition 2.6.2 can be found in

Dowling (in press), and in Koppen (in press). Both articles suggest algorithms which control the questioning of an expert by families of sets representing the sets of positive and negative inferences at each step l, and both algorithms have been applied successfully in practice. The algorithms do, however, differ in various aspects which are discussed in Dowling (in press).

The following example illustrates the interactive procedure for questioning an expert. We exemplify the updating of the inferences $(r \circ f)(A_l)$ of the statements accepted up to a step l, the failure space $f(A_l)$, and the negative inferences R_l^*.

EXAMPLE 2.6.1 Let $V = \{a, b, c, d\}$ be the set of questions from the introductory example 2.1.1 and let $\mathbf{T}$ be the implication relation representing the "tautological" statements of the form (2.1) defined by (2.2). For simplification we will write the premises P of the pairs $(P, q) \in (2^V \setminus \emptyset) \times V$ without surrounding braces, and without separating the elements of a premise P by commas. For example, the pair $(\{a, b\}, d)$ with the premise $P = \{a, b\}$ will be written as (ab, d).

The steps of the procedure for querying an expert will again be denoted by the index l. The procedure is so that the sets of statements presented for judgment are consistent; i. e., a member $(P, q) \in (2^V \setminus \emptyset) \times V$ presented for judgment at a step $l + 1$ is neither a inference in $(r \circ f)(A_l)$ nor a negative inference in R_l^*. The procedure terminates as soon as

$$(r \circ f)(A_l) \cup R_l^* = (2^V \setminus \emptyset) \times V.$$

Whenever a statement (P, q) is accepted by the expert at step $l+1$ we write $(P, q) \in A_{l+1} \setminus A_l$. If a statement (P, q) is rejected by the expert at step $l + 1$, then we write $(P, q) \in R_{l+1} \setminus R_l$.

- Step $l = 0$, $A_0 = \emptyset$, $R_0 = \emptyset$:
  $f(A_0) = 2^{\{a,b,c,d\}}, \quad (r \circ f)(A_0) = \mathbf{T}, \quad R_0^* = \emptyset.$
- Step $l = 1$, $(ab, c) \in R_1 \setminus R_0$:
  $f(A_1) = f(A_0), \quad (r \circ f)(A_1) = (r \circ f)(A_0),$
  $R_1^* = R_0^* \cup \{(ab, c), (a, c), (b, c)\}.$
- Step $l = 2$, $(ab, d) \in R_2 \setminus R_1$:
  $f(A_2) = f(A_1), \quad (r \circ f)(A_2) = (r \circ f)(A_1),$
  $R_2^* = R_1^* \cup \{(ab, d), (a, d), (b, d)\}.$
- Step $l = 3$, $(ac, b) \in A_3 \setminus A_2$:
  $f(A_3) = \{\emptyset, \{a\}, \{b\}, \{c\}, \{d\}, \{a, b\}, \{a, d\}, \{b, c\}, \{b, d\}, \{c, d\},$
  $\{a, b, c\}, \{a, b, d\}, \{b, c, d\}, V\},$
  $(r \circ f)(A_3) = (r \circ f)(A_2) \cup \{(ac, b), (acd, b)\}, \quad R_3^* = R_2^*.$
- Step $l = 4$, $(a, b) \in A_4 \setminus A_3$:
  $f(A_4) = \{\emptyset, \{b\}, \{c\}, \{d\}, \{a, b\}, \{b, c\}, \{b, d\}, \{c, d\},$
  $\{a, b, c\}, \{a, b, d\}, \{b, c, d\}, V\},$
  $(r \circ f)(A_4) = (r \circ f)(A_3) \cup \{(a, b), (ad, b)\}, \quad R_4^* = R_3^*.$
- Step $l = 5$, $(ac, d) \in A_5 \setminus A_4$:
  $f(A_5) = \{\emptyset, \{b\}, \{c\}, \{d\}, \{a, b\}, \{b, c\}, \{b, d\}, \{c, d\},$

$\{a,b,d\},\{b,c,d\},V\}$,
$(r\circ f)(A_5)=(r\circ f)(A_4)\cup\{(ac,d),(abc,d)\}, \quad R_5^*=R_4^*.$

- Step $l=6$, $(ad,c)\in R_6\setminus R_5$:
  $f(A_6)=f(A_5), \quad (r\circ f)(A_6)=(r\circ f)(A_5),$
  $R_6^*=R_5^*\cup\{(ad,c),(abd,c),(bd,c),(d,c)\}.$
- Step $l=7$, $(bc,a)\in R_7\setminus R_6$:
  $f(A_7)=f(A_6), \quad (r\circ f)(A_7)=(r\circ f)(A_6),$
  $R_7^*=R_6^*\cup\{(bc,a),(b,a),(c,a)\}.$
- Step $l=8$, $(bc,d)\in A_8\setminus A_7$:
  $f(A_5)=\{\emptyset,\{b\},\{c\},\{d\},\{a,b\},\{b,d\},\{c,d\},\{a,b,d\},\{b,c,d\},V\},$
  $(r\circ f)(A_8)=(r\circ f)(A_7)\cup\{(bc,d)\},$
  $R_8^*=R_7^*\cup\{bcd,a),(bd,a),(cd,a),(d,a)\}.$
- Step $l=9$, $(c,d)\in R_9\setminus R_8$:
  $f(A_9)=f(A_8), \quad (r\circ f)(A_9)=(r\circ f)(A_8), \quad R_9^*=R_8^*\cup\{(c,d)\}.$
- Step $l=10$, $(cd,b)\in R_{10}\setminus R_9$:
  $f(A_{10})=f(A_9), \quad (r\circ f)(A_{10})=(r\circ f)(A_9),$
  $R_{10}^*=R_9^*\cup\{(cd,b),(c,b),(d,b)\}.$

At the last step $l=10$ we obtain that

$$(r\circ f)(A_{10})\cup R_{10}^*=(2^V\setminus\emptyset)\times V. \qquad \square$$

In this example the expert has given ten judgments instead of the possible twenty eight judgments on statements which do not correspond to tautologies.

A procedure selecting the statements presented to an expert for judgment may be restricted by additional conditions which further reduce the number of statements to be judged. The conditions [a] and [b] of definition 2.6.2 are chosen so that they do not yet predetermine the sequence in which the statements are selected for judgment. Ordering the statements to be judged prior to presentation could influence the efficiency of the procedure. Whenever the set V of questions is large, a means of limiting the size of the premise of the statements to be judged to some upper bound will be necessary. For example, were a statement to fill a terminal screen, it would be unlikely that the expert could make a reliable judgment. A condition minimizing the size of the premise of the judged statements is given by Müller (1989).

Such a procedure for questioning experts has been applied in three projects. In the first project teachers judge statements on 50 problems from the field of U. S. high school mathematics (Falmagne et al, 1990). In the second project tutors of a system for computer aided design (CAD) judged statements on 28 skills required to use the CAD system. The third project is the one mentioned in the introductory example.

For further reading 2.6.1 We recommend the articles cited in this section. For readers who want to implement an algorithm for questioning experts we suggest the articles Dowling (in press), and Koppen (in press). $\square$

References

Birkhoff, G. (1979). *Lattice theory.* Providence: American Mathematical Society.

Degreef, E., Doignon, J.-P., Ducamp, A., & Falmagne, J.-C. (1986). Languages for the assessment of knowledge. *Journal of Mathematical Psychology*, **30**, 234–256.

Doignon, J.-P., & Falmagne, J.-C. (1985). Spaces for the assessment of knowledge. *International Journal of Man-Machine Studies*, **23**, 175–196.

Dowling, C. E. (1993). Applying the basis of a knowledge space for controlling the questioning of an expert. *Journal of Mathematical Psychology*, **37**, 21–48.

Falmagne, J.-C., Koppen, M., Villano, M., Doignon, J.-P., & Johannesen, L. (1988a). Introduction to knowledge spaces: How to build, test and search them. *Psychological Review*, **97**, 201–224.

Falmagne, J.-C., & Doignon, J.-P. (1988a). A class of stochastic procedures for the assessment of knowledge. *British Journal of Mathematical and Statistical Psychology*, **41**, 1–23.

Falmagne, J.-C., & Doignon, J.-P. (1988b). A Markovian procedure for assessing the state of a system. *Journal of Mathematical Psychology*, **32**, 232–258.

Koppen, M. (in press). Extracting human expertise for constructing knowledge spaces: An algorithm. *Journal of Mathematical Psychology.*

Koppen, M., & Doignon, J.-P. (1990). How to build a knowledge space by querying an expert. *Journal of Mathematical Psychology*, **34**, 311–331.

Monjardet, B. (1970). Tresses, fuseaux, préordres et topologies. [Lattices, intersections, preorders and topologies] *Mathematiques et Sciences Humaines*, **30**, 11–22.

Müller, C. E. (1989). A procedure for facilitating an expert's judgments on a set of rules. In E. E. Roskam (Ed.), *Mathematical psychology in progress.* Heidelberg, New York: Springer.

Nilsson, N. J. (1971). *Problem solving methods in artificial intelligence.* New York: McGraw Hill.

Ore, O. (1962). *Theory of graphs.* Providence: American Mathematical Society Colloquium publications Vol. XXXVIII.

Stanat, D. F., & McAllister, D. F. (1977). *Discrete mathematics in computer science.* Englewood Cliffs: Prentice-Hall.

3 Establishing knowledge spaces by systematical problem construction

D. Albert[1] and T. Held
Universität Heidelberg, Psychologisches Institut, Hauptstraße 47–51,
D-6900 Heidelberg, Germany
E-mail: a92@vm.urz.uni-heidelberg.de

3.1 Introduction

Procedures which are to test a subject's knowledge concerning a specific domain obviously require (in addition to other prerequisites) a set of problems.

The answers to these problems may serve as a basis for a hypothesis about the subject's actual knowledge. A teacher might assume that a student possesses all of the knowledge necessary to solve the problems. There are at least two different methods of questioning:

- All available problems are presented and the set of problems which have been solved correctly is assumed to represent the student's knowledge concerning the investigated domain. This method seems to be rather uneconomical, particularly if the set of problems is quite large.
- The problems which are presented are selected adaptively from a problem set. If a teacher presents a problem which is solved correctly by a student, the next problem will probably be more difficult because the teacher will suppose that the student is capable of solving all easier problems.

Certainly the second method of knowledge assessment requires an a-priori hypothesis about a *structure* on the problem set. Such a hypothesis may, for example, be: "If a student succeeds in multiplying two fractions, she or he will also be able to multiply two natural numbers". The manner in which a teacher will conduct an assessment procedure depends largely on his or her own experience and knowledge. This experience and knowledge are implicitly used for structuring a knowledge domain. We would like to investigate these hypotheses of a domain's structure in a formal way. First we will take a look at various types of relations that may be defined on a set of problems. This overview

[1]The research reported in this paper is based on Albert (1989, 1991); it was supported by Grant Lu 385/1 of the Deutsche Forschungsgemeinschaft to J. Lukas and D. Albert at the University of Heidelberg. We are grateful to P. Hellriegel, B. Hierholz, J. Ptucha and M. Wölk (University of Heidelberg) for collecting the data, to J. Heller (University of Regensburg), and to J. Lukas and H. Rodenhausen (University of Heidelberg) for their invaluable comments on an earlier draft of this paper. We also thank M. Sauter (University of Heidelberg) for the proof of Proposition 3.3.1.

serves as a prerequisite for a short introduction to the theory of knowledge spaces put forward by Doignon and Falmagne (1985) (see Section 3.2).

We will focus on the question, how a relation on a set of problems can be established by systematical problem construction (Section 3.3).

First we will give some examples of relations on sets of problems. For this purpose, we must introduce a few basic concepts of ordering theory, e. g. how can statements like "problem x is more difficult than problem y" or "problem x is at least as difficult as problem y" be denoted?

Our examples will involve quasi-orders, linear orders (chains) and anti-chains. Similar examples can be found for other types of orders such as weak orders and partial orders. For a general introduction to ordering theory we refer to Davey and Priestley (1990). Let us first give some definitions.

Definition 3.1.1 $M_1 \times \ldots \times M_n := \{(x_1, \ldots, x_n) \mid x_i \in M_i\}$ is called the *Cartesian product* of the sets $M_1, \ldots, M_n$. □

Definition 3.1.2 A subset $P \subseteq M_1 \times M_2$ is called *binary relation*. □

Definition 3.1.3 Let S be a set and P a binary relation on S. $\langle S, P\rangle$ is a *quasi-order* if for all $x, y, z \in S$,[2]

(i) xPx (reflexive),
(ii) $xPy \wedge yPz \Rightarrow xPz$ (transitive). □

Quasi-orders can be depicted as *Hasse diagrams*. In these diagrams, the relation is shown in a very economical way, i. e., lines for 'reflexive' ordered pairs such as (x, x), and for ordered pairs which can be derived through transitivity (e. g., if $(x, y), (y, z) \in P$, then $(x, z) \in P$) are omitted. For a detailed definition see, e. g. Davey and Priestley (1990, p. 7).

Example 3.1.1 We have a set $Q = \{x, y, z\}$ of problems on which a quasi-order $\{(x, x), (y, y), (z, z), (y, x), (z, x)\}$ is defined. Relation P is shown in Figure 3.1(a). For our set of questions, this means that problem x is should be more 'difficult' than problem y and problem z. It is assumed that problems y and z cannot be compared. □

Definition 3.1.4 Let S be a set and P a binary relation on S. $\langle S, P\rangle$ is a *linear order* if for all $x, y, z \in S$,

(i) $xPy \vee yPx$ (connected),
(ii) $xPy \wedge yPx \Rightarrow x = y$ (antisymmetric),
(iii) $xPy \wedge yPz \Rightarrow xPz$ (transitive). □

Note that each linear order is also reflexive. Because reflexivity can be derived from the properties stated above, we do not have to mention it explicitly. In general, a linear order is a special case of a quasi-order with the additional properties of connectedness and antisymmetry.

[2]The expression xPy denotes that the ordered pair (x, y) is an element of the relation $P \subseteq S \times S$.

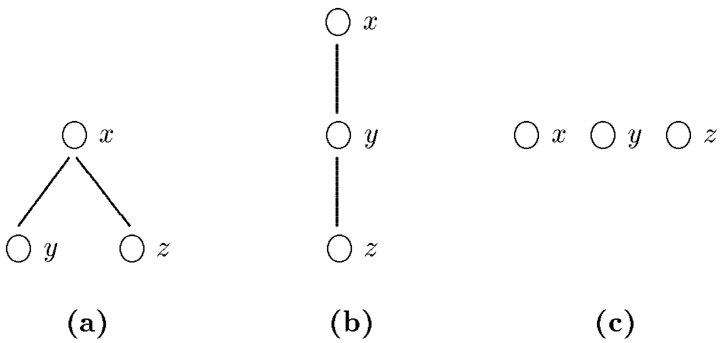


Figure 3.1. Hasse diagrams for Examples 3.1.1, 3.1.2 and 3.1.3.

EXAMPLE 3.1.2 On a set $Q = \{x, y, z\}$ of problems a linear order $\{(x, x), (y, y), (z, z), (y, x), (z, y), (z, x)\}$ is defined. First let us look at the Hasse diagram in Figure 3.1(b). We can see that this order is a special case of a quasi-order, because every problem is comparable to all other problems. Problem x, for example, is supposed to be more 'difficult' than problems y and z. This type of problem ordering is known as a *Guttman scale* (Guttman, 1947, 1950). □

DEFINITION 3.1.5 Let S be a set and P a binary relation on S. $\langle S, P\rangle$ is an *antichain* if xPy in S if and only if $x = y$. □

Note that the antichain is also a special case of a quasi-order. Its defining properties are reflexivity, transitivity, symmetry and antisymmetry.

EXAMPLE 3.1.3 On a set $Q = \{x, y, z\}$ of problems an antichain order $\{(x, x), (y, y), (z, z)\}$ is defined. The Hasse diagram in Figure 3.1(c) shows that the problems of Q are not connected. The postulation of an antichain order may be adequate for sets of heterogeneous and completely incomparable problems. However, it is clear that such a set cannot be used for an economical adaptive questioning procedure because no conclusions can be drawn from a subject's answers. □

An important topic in problem ordering is the interpretation of the binary relation that is defined on the problem set. Up until now, we have mentioned only a few unspecified differences in 'difficulty'. An interpretation will in addition to further theoretical considerations be introduced in the following section.

3.2 Knowledge spaces

The theory of knowledge spaces was first introduced by Doignon and Falmagne (1985). We will begin with a presentation of the basic concepts of this theoretical approach.

Suppose we must test a student's knowledge of elementary algebra. It is advisable to begin with a problem of medium difficulty. Depending on the answer, we can go on with a more difficult or with an easier problem. We will proceed in this manner until we have acquired sufficient information about the student's knowledge.

An important prerequisite for such a procedure is a hypothesis about a structure on the problem set. The theory of Doignon and Falmagne shows, how the structure of problems can be represented in a formal way (for an introduction see Falmagne, Koppen, Johannesen, Villano, and Doignon, 1990).

Let $Q = \{x, y, z\}$ be a set of problems which is used for an examination. For some of these problems, a statement such as *"if a student is able to solve a specific problem in Q, he or she will also be able to solve other questions belonging to Q"* may be plausible. This can be formalized in terms of a binary relation $\preceq$. The expression $y \preceq x$ is interpreted as follows: Given a correct response to problem x, we *surmise* a correct answer to problem y. The relation $\preceq$ is called *surmise relation*. It is assumed that the surmise relation is a quasi-order on Q.

A surmise relation can be depicted as a Hasse diagram. Figure 3.2(a) shows a hypothetical order on the problem set Q. According to this order we assume

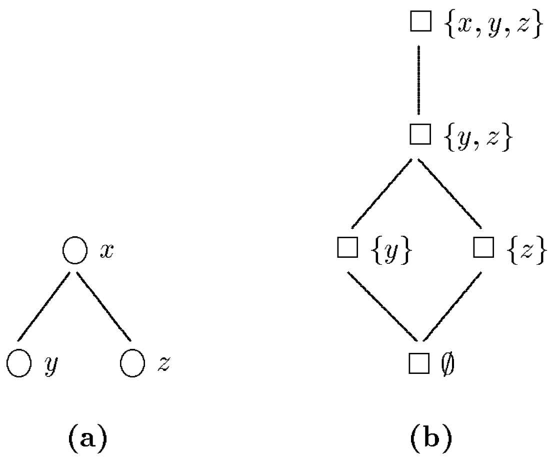


Figure 3.2. Surmise relation and knowledge states for the problems in Q (problems are marked by circles, states are marked by squares).

that each of the students capable of solving problem x, will also be able to solve problem y *and* problem z. Based on this assumption we can collect all subsets of Q which agree with the surmise relation. These subsets are called *knowledge states.*

Definition 3.2.1 (see Falmagne et al., 1990) Let Q be a set of problems. $K \subseteq Q$ is a *state* $\Leftrightarrow$ $(\forall q, t \in Q, q \preceq t \wedge t \in K \Rightarrow q \in K)$. □

The family of all possible states with respect to a set of problems is a *knowledge structure.* For our example, we obtain the structure $\mathcal{F}$:

$$\mathcal{F} = \{\emptyset, \{y\}, \{z\}, \{y, z\}, \{x, y, z\}\}.$$

This knowledge structure contains all subsets of Q which are expected to occur as results of diagnostic procedures. The purpose of such procedures is to assign subjects to one of these states without presenting all problems in Q (see Falmagne et al., 1990). Figure 3.2(b) shows the Hasse diagram for this knowledge structure. We can see that $\mathcal{F}$ is closed under union and intersection, so for all states S, S' in $\mathcal{F}$ the following properties hold:

$$\text{if } S, S' \in \mathcal{F} \text{ then } S \cup S' \in \mathcal{F},$$
$$\text{if } S, S' \in \mathcal{F} \text{ then } S \cap S' \in \mathcal{F}.$$

A knowledge structure with these properties is called a *quasi-ordinal knowledge space*. A one-to-one correspondence between transitive and reflexive orders and families of (knowledge) states which are closed under union and intersection is established by a theorem by Birkhoff (1937) (see Doignon and Falmagne, 1985).

The restriction of closure under union and intersection is very great and somewhat unrealistic for many knowledge domains.

Therefore Doignon and Falmagne introduced, as a generalization of quasi-ordinal knowledge spaces, the concept of *knowledge spaces*. Knowledge spaces are families of states which are closed under union, but do not have to be closed under intersection. Hence, every quasi-ordinal knowledge space is also a knowledge space. Doignon and Falmagne showed that there is a one-to-one correspondence between knowledge spaces and the so-called *surmise-systems*. This will not be discussed in detail here. Our further considerations will deal solely with quasi-ordinal knowledge spaces. The contributions of C. E. Dowling and J.-P. Doignon in this volume use the more general concept of knowledge spaces.

Quasi-ordinal knowledge spaces can also be derived from 'special cases' of quasi-ordered problem sets such as sets with an antichain order or linearly ordered sets. The following examples will illustrate these cases.

EXAMPLE 3.2.1 We have a set $M = \{x, y, z\}$ with three problems. If there is an antichain order $\{(x, x), (y, y), (z, z)\}$ defined on M, the following knowledge states can appear:

$$\emptyset, \{x\}, \{y\}, \{z\}, \{x, y\}, \{x, z\}, \{y, z\}, \{x, y, z\}.$$

The set of these knowledge states is equal to the *power set* of M. Figure 3.3(a) shows the surmise relation and the knowledge states. □

Now we will take a look at a set of problems on which a linear order is defined.

EXAMPLE 3.2.2 Suppose, we have a set $M = \{x, y, z\}$ of three problems. If a linear order $\{(x, x), (y, y), (z, z), (y, x), (z, y), (z, x)\}$ is defined on M, the following knowledge states are assumed:

$$\emptyset, \{z\}, \{y, z\}, \{x, y, z\}.$$

The surmise relation and the corresponding knowledge states are shown in Figure 3.3(b). □

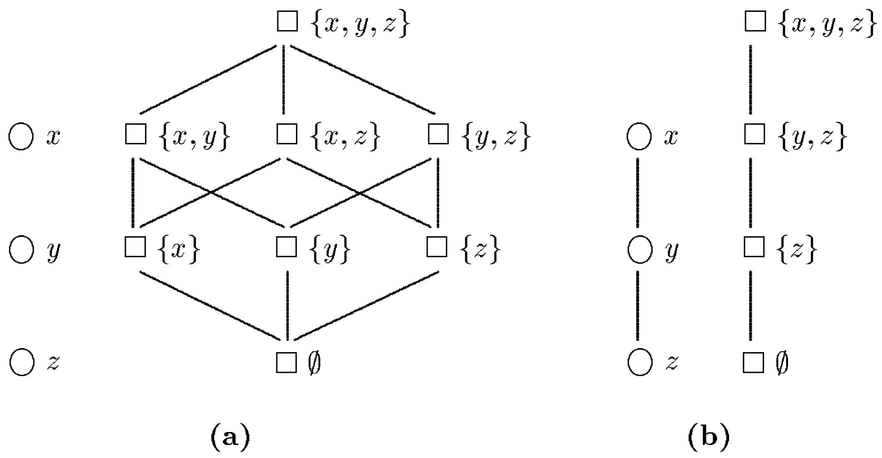


Figure 3.3. Surmise relation and knowledge states for Examples 3.2.1 and 3.2.2.

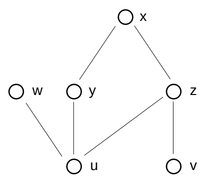
EXERCISE 3.2.1 Show that linear orders are reflexive. Note: try to use the given properties of the order (see Definition 3.1.4) for the proof. □

EXERCISE 3.2.2 Is the family of states

$$\begin{aligned}\mathcal{F}_{ex} = \; & \{\emptyset, \{x\}, \{y\}, \{z\}, \{w,x\}, \{w,z\}, \{x,y,z\}, \\ & \{w,x,y\}, \{w,x,z\}, \{w,z,y\}, \{w,x,y,z\}\}\end{aligned}$$

closed under union? Is it closed under intersection? Check, if a corresponding surmise relation can be found. If this is true, draw the Hasse diagram and list the elements of the relation. □

EXERCISE 3.2.3 Which quasi-ordinal knowledge space corresponds to the following problem structure?



Note: use Definition 3.2.1 and the fact that a quasi-ordinal knowledge space is closed under union. The result must consist of 16 knowledge states. □

EXERCISE 3.2.4 How many possible knowledge states can be derived from a set of 10 questions on which (a) an antichain order and (b) a linear order is defined? □

3.3 'Component-based' establishment of surmise relations

We will expand our considerations about problems by a topic we will call *problem component* or simply *component*. One way to facilitate problem comparison is by *systematical problem construction*. Construction principles are applied on well-defined sets of problem components. Further, by means of their associated component structures, we can both provide a precise description of problems and the class of possible problem variations. Certainly, components have to be equipped with properties which are prerequisites for a successful combination. We will discuss this later (Section 3.3.3).

Before we introduce two construction principles, a short sketch of the concept that we call a problem component should be drawn. As an example, let us imagine we are asked to solve an algebraic problem, e. g. the multiplication of two fractions. Although this is a simple task, we will not be able to give the solution, if we do not know some basics of algebra. Some of these basics may be 'multiplication of natural numbers', 'division of natural numbers' and 'rules for the multiplication of fractions'.

These items can be seen as *cognitive demands* on a subject confronted with the problem. If the subject does not have the knowledge at his or her disposal which is 'demanded' or if the subject is not able to apply this knowledge, it is supposed that the answer to the problem will be incorrect — assuming the guessing probability is equal to zero.

3.3.1 Union and intersection based rules

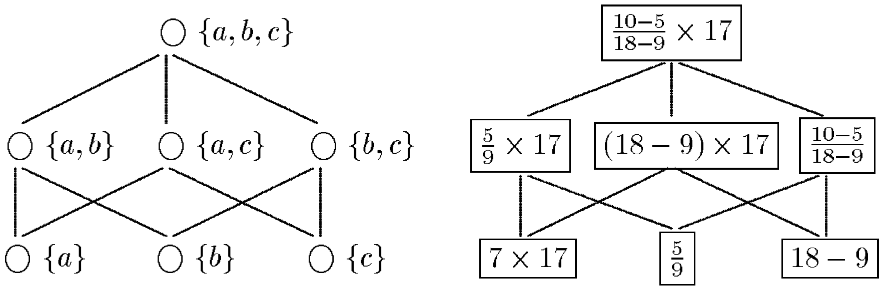
Our further considerations will make use of the basics of ordering theory introduced in Sections 3.1 and 3.2. We will now also take into account the representation of problems as sets of components. The following examples will give a first idea of how problems can be constructed from components.

EXAMPLE 3.3.1 Let $C = \{a, b, c\}$ be a set of problem components. We assume that these components are unconnected, i. e. that an antichain order is defined on C. Let us identify problems with subsets of C. With respect to the antichain order defined on the components, we assume that no dependencies between components exist. Hence, every subset of C can be identified with a potential problem and thus with an element of a problem set Q (in this case subsets of C denote problems):

$$Q = \{\emptyset, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}.$$

The combination of the components a, b and c has led to seven problems. The 'empty problem' $\emptyset$ is left out, because it cannot be shown. We assume that a problem is more difficult than another problem if it is characterized by all components of the other problem and by at least one more component. According to this assumption we can state a hypothetical order as shown in Figure 3.4(a). The next step is the application of the construction to concrete problem components and the ordering principle on concrete problems. For example, we define for a, b, c: $a\widehat{=}$ multiplication of numbers, $b\widehat{=}$ division of numbers, $c\widehat{=}$ subtraction of numbers. Hence, $\{a, b, c\}$ is a problem which contains multiplication, division and subtraction, e. g.

$$\frac{10-5}{18-9} \times 17.$$



(a) **(b)**

Figure 3.4. Problem structure for the questions in Q and an example concerning calculation problems.

The hypothetical order for a set of problems is shown in Figure 3.4(b). □

We would like to stress that a surmise relation was established by constructing and ordering the problems in this way. Unconnectedness, however, is not a necessary property of the component set. As we will show, this method is also applicable to linearly ordered and quasi-ordered component sets. Examples 3.3.2 and 3.3.3 give an idea of this method.

EXAMPLE 3.3.2 We have a set $C = \{a, b, c\}$ of linearly ordered problem components. Figure 3.5 shows on the left a possible Hasse diagram for the component structure and on the right the structure of the resulting problems (components are marked by triangles). Taking into account that the elements

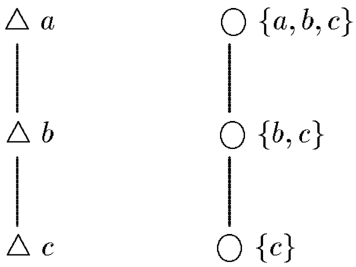


Figure 3.5. Component structure and problem structure for Example 3.3.2.

of C are linearly ordered, only three questions can be produced. This linear order may, for example, be induced by constraints on combining the components. Such constraints exist, for instance, for sets of non-independent components. In our example, a may be a problem component which also contains b and c in some way. Therefore, if one part of a problem is associated with a, then b and c are automatically involved. To illustrate we will assume that component a corresponds to the addition of natural numbers within the hundreds, e. g. $619 + 347$. Further, we will assume that b corresponds to the addition of the numbers between one an ten. We therefore see that b is also necessarily an element of a. Thus b is an element of problems containing a. □

EXAMPLE 3.3.3 We assume that a quasi-order is defined on a set $C = \{a, b, c\}$ of problem components. Figure 3.6 shows one of the possible Hasse diagrams (left) and the corresponding problem structure (right). From this quasi-

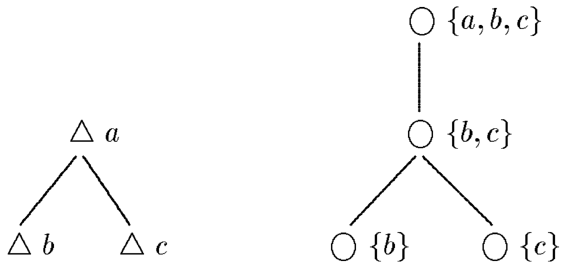


Figure 3.6. Component structure and problem structure for Example 3.3.3.

ordered problem set, single-component problems can be constructed which consist either of b or c, so we may suppose, that b and c are thematically independent, but are both involved in a in some way. □

After this brief introduction to one possible method of constructing and ordering problems by means of problem components, we will state these principles formally:

DEFINITION 3.3.1 Let C be a set of components and $\preceq$ a quasi-order on C. The *component space* $\mathcal{F}_C$ is the family of all subsets T of C for which

$$x \in T, y \preceq x \Rightarrow y \in T$$

holds. □

Given a component space $\mathcal{F}_C$, according to each element T of $\mathcal{F}_C$ a problem q_T is formulated. A *surmise relation* R on the problem set $Q = \{q_T \mid T \in \mathcal{F}_C\}$ is defined by the following condition:

$$q_T \; R \; q_{T'} :\Leftrightarrow T \subseteq T'.$$

This means that the problems are identified with the elements of $\mathcal{F}_C$, while the relation R is identified with $\subseteq$.

It is easy to verify that R is a transitive relation: Let M, M', M'' be sets with $M \subseteq M'$ and $M' \subseteq M''$. M' which contains M is a subset of M'', thus $M \subseteq M''$, which means that '$\subseteq$' is transitive (see Definition 3.1.3 (ii)).

This ordering principle of set inclusion is based on the plausible assumption that a subject succeeding in the solution of a given problem will also be able to solve all the partial items of this problem. We would like to note here that a reversed statement such as "if someone is able to solve the partial items, she or he will also be able to solve the superset item" is not expected to hold true. The combination of problem components may lead to some additional difficulties which might not appear within the single components. An empirical example will be given in in Section 3.4.1.

As an ordering method, set inclusion can be applied to very different theoretical approaches of the field of knowledge assessment. For examples, we refer to the investigations of Korossy (in preparation).

Exercise 3.3.1 Why does the correct solution of two problems not necessarily imply the correct solution of a problem which is built from the union of components of these problems? □

3.3.2 Product formation based rules

Up until now, we have focused only on single sets of components which were characterized by an order which was defined on the component set. In this section, we will turn our attention to the construction of problems which consist of components with variable attributes. Here every problem is equipped with the same number of components. New problems are constructed by varying the components' attributes. The problems' order will be derived from relations which are defined on the set of attributes. The following example gives an idea of this method.

Example 3.3.4 Let $A = \{a_1, a_2, a_3\}$ and $B = \{b_1, b_2\}$ be problem components; a_1, a_2, a_3 and b_1, b_2 are the attributes of these components. On both sets A and B a linear order is defined (see Figure 3.7; attributes are marked by black triangles).

Suppose we want to construct simple algebra problems. One component may be the set of numbers which is used within a calculation, the other component is characterized by the operations which are to be applied on the set of numbers. We define: $a_1 \widehat{=}$ use of real numbers,$a_2 \widehat{=}$ use of integers, $a_3 \widehat{=}$ use of natural numbers, $b_1 \widehat{=}$ calculation of powers, $b_2 \widehat{=}$ addition.

Both operations of B can be applied on the sets of numbers of A. Therefore, we can construct problems which contain one property of A and one property of B. The problem $(-5)^2$, for instance, corresponds to the combination of a_2 and b_1. From A and B we can construct a set $\mathcal{F}_p$ of six problems:

$$\mathcal{F}_p = \{(a_1, b_1), (a_1, b_2), (a_2, b_1), (a_2, b_2), (a_3, b_1), (a_3, b_2)\}.$$

We see that all problems of $\mathcal{F}_p$ consist of two components which are represented by their attributes $a_1, \ldots, a_i, \ldots, a_n$ and $b_1, \ldots, b_j, \ldots, b_m$. The problem structure for this example is shown in Figure 3.7. □

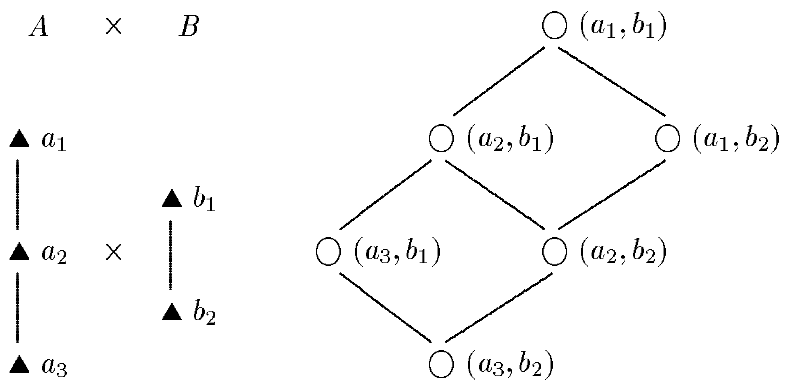


Figure 3.7. Attributes and problem structure for Example 3.3.4.

Now we must determine the principles by which these problems were constructed and ordered. Let us look at the steps in detail. Components are sets of attributes. It is supposed that the attributes in a component cannot be combined with one another. In our example we have two components whose attributes are to be combined. This combination has been established by forming the *Cartesian product* of A and B (see Definition 3.1.1). Looking at Example 3.3.4, we can easily check that $\mathcal{F}_p$ is a set which contains the product of A and B.

DEFINITION 3.3.2 Let S be a set and P a binary relation on S. $\langle S, P\rangle$ is a *partial order* if for all $x, y, z \in S$,

(i) xPx (reflexive),
(ii) $xPy \wedge yPz \Rightarrow xPz$ (transitive),
(iii) $xPy \wedge yPx \Rightarrow x = y$ (antisymmetric). □

In order to establish a problem structure (surmise relation) as shown in Figure 3.7, it is necessary to compare the generated problems in pairs with respect to the components' attributes. Formally, the ordering rule we applied was:

> Let $C_1 \dots C_n$ be component sets on which partial orders $R_1, \dots, R_n$ are defined. On the Cartesian product $C_1 \times \dots \times C_n$ an order $\preceq$ is imposed by defining
>
> $$(x_1, \dots, x_n) \preceq (y_1, \dots, y_n) \Longleftrightarrow (\forall i)\ x_i\ R_i\ y_i.$$

Expressed in words: We surmise that a problem q_1 is at least as difficult to solve as a problem q_2, if all attributes of q_1 are at least as difficult as the corresponding attributes in q_2 with respect to the relations R_i defined on the attribute sets. This principle is known as 'coordinatewise order'; for a description see Davey and Priestley (1990, p. 18). According to Birkhoff (1973), $\preceq$ is a partial order. Note that this method is also known from decision theory where the choice heuristic called *dominance rule* corresponds to coordinatewise orders. For a more detailed discussion of parallels to choice heuristics, we refer to Section 3.5 in this paper.

Extensions of this ordering method applied to problems of elementary probability calculus are introduced in Held (1992, 1993). Here an approach to the component based establishment of surmise systems can also be found.

Since the attributes of the components must be compared, it is necessary to define an order on each set of attributes. Example 3.3.4 showed the case of linearly ordered attributes. Example 3.3.5 demonstrates the ordering of problems which were constructed from quasi-ordered sets of attributes.

EXAMPLE 3.3.5 Let $A = \{a_1, a_2, a_3\}$ and $B = \{b_1, b_2, b_3\}$ be quasi-ordered sets of attributes. Figure 3.8 shows the Hasse diagrams for these sets and the corresponding problem structure. A procedure for the graphical construction of such products is given in Davey and Priestley (1990, p. 19). □

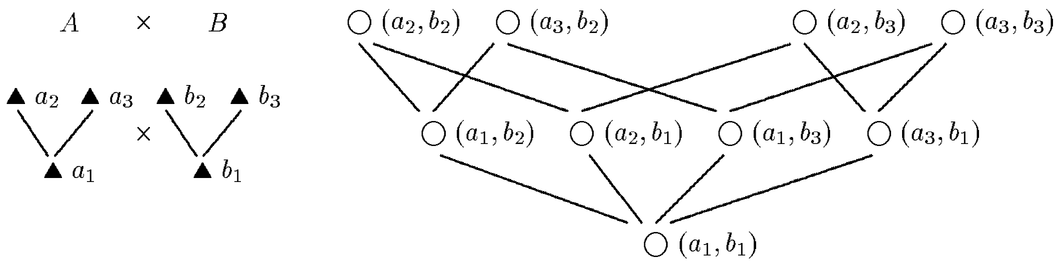


Figure 3.8. Attributes and problem structure for Example 3.3.5.

Problems which were constructed by product formation can also be ordered *lexicographically*. Lexicographic orders are also known from decision theory (see e. g. Fishburn, 1974).

EXAMPLE 3.3.6 As in Examples 3.3.4 and 3.3.5, we have two components $A = \{a_1, a_2, a_3\}$ and $B = \{b_1, b_2\}$. It is assumed that component A is 'more important' than component B. Figure 3.9 shows the lexicographic order of the product $A \times B$. How was this order established? First, we will describe the

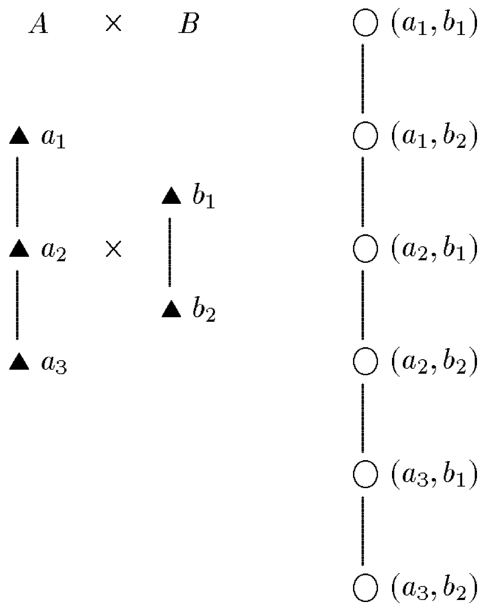


Figure 3.9. Lexicographic order.

general principle. The n-tuples which are to be ordered are compared pairwise beginning with the first elements (here: a_i). Since it is assumed that A is the most 'important' component it is also assumed that if these elements are not identical, the n-tuple which contains the subordinate element with respect to the order on A is subordinate to the other n-tuple. In Figure 3.9 we see that this is the case for all tuples (a_1, b_i) and (a_2, b_j). If the first elements are identical, the second pair of elements will be compared and the n-tuple with the subordinate element is subordinate (see all tuples (a_i, b_1) and (a_i, b_2)). This procedure which is known from dictionaries continues on until two different elements are found or until there are no more elements left to compare. □

Now we will describe the lexicographic ordering in a formal way and prove that it imposes a linear order on a problem set (i. e., the proof of transitivity is shown below, while proving reflexivity and connectedness is left to the reader as an exercise).

For establishing a lexicographic order we need sets A_i of attributes and relations P_i which are defined on the sets A_i. $\langle A_i, P_i\rangle$ with $i = 1, \ldots, n$ have to be strict linear orders.

Definition 3.3.3 Let A be a set. A *strict linear order* on A is a binary relation P, such that for all $a, b, c \in A$,

(i) $\neg(aPa)$ for all $a \in A$ (irreflexive),
(ii) $aPb \wedge bPc \Rightarrow aPc$ (transitive),
(iii) for all $a \neq b \in A$, $aPb \vee bPa$ (weakly connected). □

Definition 3.3.4 Let $\langle A_i, P_i\rangle, i = 1, \ldots, n$, be strict linear orders. $\langle A_1 \times A_2 \times \ldots \times A_n, Q\rangle$ with $Q \subseteq ((A_1 \times A_2 \times \ldots \times A_n) \times (A_1 \times A_2 \times \ldots \times A_n))$ is a *lexicographic order*, i. e. for

$$(a_1, a_2, \ldots, a_n), (b_1, b_2, \ldots, b_n) \in A_1 \times A_2 \times \ldots \times A_n$$

holds

$$(a_1, a_2, \ldots, a_n)\ Q\ (b_1, b_2, \ldots, b_n)$$

if and only if,

$$\begin{aligned} &a_1P_1b_1 \ \vee\ (a_1 = b_1 \ \wedge\ a_2P_2b_2) \ \vee\ (a_1 = b_1 \ \wedge\ a_2 = b_2 \ \wedge\ a_3P_3b_3) \ \vee\ \ldots \\ &\qquad \ldots \vee\ (a_1 = b_1 \ \wedge\ a_2 = b_2 \ \wedge\ \ldots \ \wedge\ a_{n-1} = b_{n-1} \ \wedge\ a_nP_nb_n) \\ &\qquad\qquad \vee\ (a_1 = b_1 \ \wedge\ a_2 = b_2 \ \wedge\ \ldots \ \wedge\ a_n = b_n). \end{aligned}$$ □

Proposition 3.3.1 *A lexicographic order is transitive.* □

Proof. Let $(a_1, a_2, \ldots, a_n)\ Q\ (b_1, b_2, \ldots, b_n)$ and $(b_1, b_2, \ldots, b_n)\ Q\ (c_1, c_2, \ldots, c_n)$ with $(a_1, \ldots, a_n), (b_1, \ldots, b_n), (c_1, \ldots, c_n) \in A_1 \times \ldots \times A_n$. Without loss of generality we can assume $\exists i \in \{1, \ldots, n\}\ (a_i \neq b_i) \wedge \exists i \in \{1, \ldots, n\}\ (b_i \neq c_i)$. Then there exists an $i \in \mathrm{N},\ 1 \leq i \leq n$ with $a_iP_ib_i$ and $a_k = b_k$ for all $k \in \mathrm{N}, k < i$, and a $j \in \mathrm{N},\ 1 \leq j \leq n$ with $b_jP_jc_j$ and $b_l = c_l$ for all $l \in \mathrm{N},\ l < j$.
Let $m = \min\{i, j\}$. Then:

$$a_m = \begin{array}{ll} b_m \wedge b_mP_mc_m & [\text{if } m = j] \quad \text{or} \\ a_mP_mb_m \wedge b_m = c_m & [\text{if } m = i] \quad \text{or} \\ a_mP_mb_m \wedge b_mP_mc_m & [\text{if } i = j] \end{array}$$

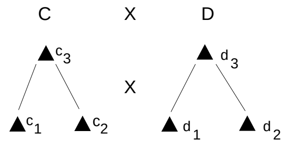
and also $a_q = b_q = c_q$ for all $q \in \mathrm{N}, q < m$; so $a_mP_mc_m$ and $a_q = c_q$ for all $q \in \mathrm{N},\ q < m$. Therefore, $(a_1, a_2, \ldots, a_n)\ Q\ (c_1, c_2, \ldots, c_n)$ ■

We have now introduced several important concepts for the construction of problems from components and for the establishment of a surmise relation on these problems. In the following section we will summarize aspects of component properties.

After these considerations we will present two empirical investigations which make use of the principles introduced for constructing and ordering problems.

Exercise 3.3.2 Form the product $M_1 \times M_2 \times M_3$ of the following sets: $M_1 = \{a_1, a_2\},\ M_2 = \{b_1, b_2, b_3\},\ M_3 = \{c_1, c_2\}$. □

Exercise 3.3.3 Give the problem structure for the product of the attribute sets C and D.



□

Exercise 3.3.4 Prove that a lexicographic order — as introduced in Definition 3.3.4 — is reflexive and connected. □

3.3.3 Comments and reflections on the concept of problem components

It is evident that the utility of the rules for problem construction depends largely on the properties of components. The selection of suitable components and attributes is surely the most important prerequisite for the application of our approach.

The examples in Sections 3.3.1 and 3.3.2 showed some simple types of components, by which the general rules for combination were demonstrated. We must be aware of the fact, however, that problems which are presented to a subject during an examination procedure do not consist solely of elements which are directly related to the considered knowledge domain. The way in which a question is phrased, for instance, may influence the problem's difficulty. Another variation in complexity can be caused by the way in which prerequisites for the solution are presented.

We will not be able to provide an exhaustive list of factors which can potentially influence the difficulty of problems. We will, however, present a few comments and examples which should serve as a basis for further reflections. We will begin with a simple example.

Example 3.3.7 We are going to construct some simple problems, concerning elementary algebra. The components at our disposal are *domain of numbers* with attributes (1) use of real numbers, (2) integers, and (3) natural numbers, respectively *operations* with (1) calculation of powers and (2) addition as attributes (see Example 3.3.4). A problem which is constructed by 'use of natural numbers' and 'calculation of powers' (in terms of Example 3.3.4, this is problem (a_3, b_1)) may look like the following:

Problem a: $12^3 = x$

or

Problem b: Mr X is playing roulette at the casino. He starts playing with a total amount of \$ 12. After five hours, he has as much money as his starting amount to the power of three. How much money does he own now? □

This example attempts to give an idea of the superficial differences between problems which contain the same components with respect to some knowledge domain. It seems plausible that these formal aspects can contribute to a problem's difficulty.

We know that complete problem sets can be built by systematically combining the components taken from one or more component sets. It is clear that components cannot be selected arbitrarily. Restrictions on combinations are likely to arise if dependencies between components which may lead to unwelcome results exist. The following two examples will give an impression.

EXAMPLE 3.3.8 We are going to construct another problem from the component properties introduced in Example 3.3.7: a combination of 'use of real numbers' and 'calculation of powers'. One of the possible problems may be $x = -2^{\frac{1}{2}}$. There are two main difficulties with this problem: (1) for $-2^{\frac{1}{2}} = \sqrt{-2}$, this problem involves the calculation of roots and (2) the result of $\sqrt{-2}$ might — as a complex number — not belong to the domain of numbers we are considering. □

EXAMPLE 3.3.9 Suppose we must compare two problems, constructed from the following component properties:

Problem 1: (calculation of fractions, division, domain: real numbers)

Problem 2: (calculation of fractions, domain: real numbers)

In this case, 'calculation of fractions' and 'division' are not independent because in a large class of problems on fractions division has to be performed. □

Another difficulty may arise when we look at the 'differences' between the attributes of a component. The question is how similar attributes should be treated. If the difference between some of the attributes does not seem to influence the difficulty of problems to a meaningful extent, the consideration of 'threshold values' might be appropriate. From the area of decision theory the *lexicographic semi-order* introduced by Tversky (1969) is well known. According to this rule the 'better' of two alternatives is preferred, if the alternatives differ by more than some threshold. The problem with this rule, however, is that it may lead to intransitive results. The result is not necessarily intransitive, but it must in all cases be checked for transitivity. For a more detailed discussion of the application of choice heuristics within our ordering approach see Section 3.5.

3.4 Empirical examples

Our two empirical examples report on experimental investigation which make use of the methods introduced for problem construction and problem ordering. The first investigation belongs to the area of psychology of thinking. It deals with the solution of chess problems. In the second experiment we focus on types of problems related to the field of inductive reasoning: the continuation of number series.

3.4.1 Construction and solution of chess problems

Chess playing is surely one of the most complex and demanding knowledge domains. This complexity makes the domain particularly interesting for cognitive scientists and psychologists. Not only the game of chess itself, but also the construction of chess problems requires a large amount of knowledge and experience. The immense number of possible moves which can be made even from a very simple constellation, makes the decision, whether one move is better than another very difficult. Grandmasters are often unable to 'proof' what move is the best in a particular situation; therefore they have to act intuitively.

An important book about the psychology of chess playing was written by De Groot (1965). He attempts to investigate the thought processes of highly trained chess players by means of introspective methods. De Groot also provides a proof scheme for objectively solvable positions, but the proof only works, if someone is able to differentiate between 'good' and 'less good' moves. This differentiation has, for complex positions, to be intuitive.

We can already see that for the construction of chess problems, we should not attempt to focus on such demanding constellations which in addition to requiring highly evolved skills are also very time consuming. In our example, we use the classical form of 'three move problems' which are familiar to every chess player. In Figure 3.10 we provide a typical example. The task is to

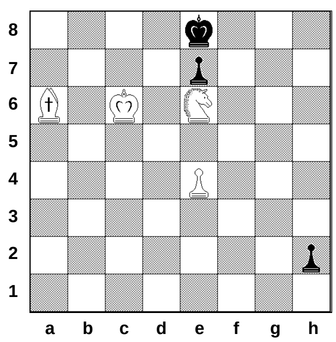


Figure 3.10. A typical three move problem.

show the moves to reach a 'winning position in three moves'. Supposing white starts, the solution is: 1. Be2 h1Q; 2. Bh5+ Qh5:; 3. Ng7+. Experienced chess players can show that for this type of problems there is only one optimal solution. Further, the time needed for handling such a position is expected to be much shorter than for a complex constellation in a real chess game.

As a next step, we have to find a way for the construction of such problems. Before we can apply one of our construction rules (see Sections 3.3.1 and 3.3.2), components have to be introduced. A basic concept in chess playing are '*motives*' which are tactical standard situations. In terms of problem solving, motives can be seen as 'subgoals' of a problem's solution. Figure 3.11 shows examples for positions in which the motives 'fork', 'pin', 'guidance', and 'deflection' occur.[3] To illustrate we will give a short description of these special

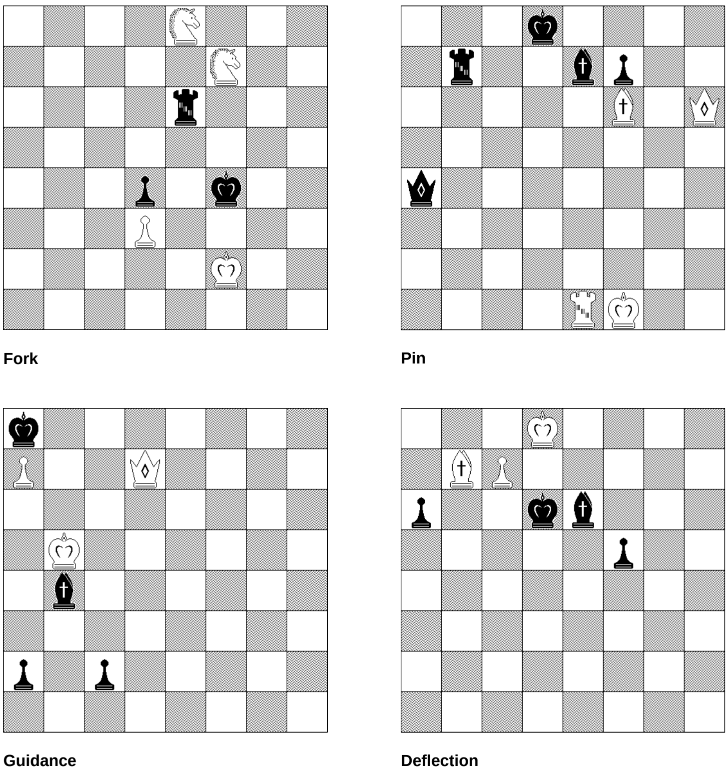


Figure 3.11. Positions in which the motives 'fork', 'pin', 'guidance', and 'deflection' occur.

situations:

- *Fork:* One piece simultaneously attacks two opposing pieces of higher value. Solution:[4] 1. Nc7 Rg6/c6; 2. Nd5+ arbitrary[5]; 3. Ne7+/Ne5+. If we take a look at one of the possible final positions (*Black:* Kf5, Rc6, ... *White:* Ne7, ...), we see that White's Knight attacks both Kf5 and Rc6.

[3]The example positions are identical to problems of the experimental investigation. Therefore these examples may appear to be rather complex.

[4]The 'solution' provides the sequence of three moves which a chess expert has considered as optimal for reaching a winning position.

[5]'Arbitrary' means that this move (in this case Black's move) is not relevant to the solution.

- *Pin:* An opposing piece is prevented from moving. Solution: 1. Qf8+ Qe8; 2. Rd1+ Rd7; 3. Be7:+. We see that the black Bishop cannot move away from e7 because of Bf6+.
- *Guidance:* An opposing piece is forced to a disadvantageous square. Solution: 1. Kb6 Ba5+/c5+; 2. Ka6c6 arbitrary; 3. Qb7/c6 mate; Black's Bishop is forced to ... a5+/c5+, otherwise 2. Qb7 mate.
- *Deflection:* An opposing piece is forced to leave an important line or square. Solution: 1. Bc8 Bd5; 2. Bf5: Bb7; 3. Be4. Black's Bishop is forced to leave e6, otherwise 2. Be6: ...

Motives can appear in a large variety of combinations and belong to the basic repertory of even only moderately experienced chess players. A complete list of all problems used in our investigation can be found in Table 3.6 in the appendix.

For the construction of problems, these motives present one possible type of problem components. As a principle of construction, we will select a small number of motives and then produce three move problems which contain *combinations* of them. In the following an investigation which makes use of this idea, will be reported.[6]

Problem construction and hypothesis As we have already indicated, the construction of problems and the establishment of the surmise relation are based on the combination of motives. The motives — symbolized by a, b, c and d — are elements of a single component set C. We assume that an antichain order is defined on C. Hence, the principle of set inclusion can be applied.

The component space $\mathcal{F}_C$ (see Definition 3.3.1) is as follows:

$$\begin{aligned}\mathcal{F}_C = \;& \{\emptyset, \{a\}, \{b\}, \{c\}, \{d\}, \{a,b\}, \{a,c\}, \{a,d\}, \{b,c\}, \{b,d\}, \\ & \{c,d\}, \{a,b,c\}, \{a,b,d\}, \{a,c,d\}, \{b,c,d\}, \{a,b,c,d\}\}.\end{aligned}$$

By means of the ordering principle of set inclusionas introduced in Section 3.3.1 we can infer a surmise relation R on the set Q of problems which are identified with the elements of $\mathcal{F}_C$. Figure 3.12 shows this relation as a Hasse diagram. Expressed in words, the hypothesis for the investigation is:

> *If a problem q identified with a component set $T \in \mathcal{F}_C$ is solved by a subject, then all problems q' which are identified with a component set $T' \in \mathcal{F}_C$ with $T' \subseteq T$ will also be solved by this subject.*

Concerning solution frequencies we expect that none of the problems will be solved more frequently than each of the corresponding subordinate problems. If, for example, problem $\{a, c\}$ is solved by n subjects we expect for problems $\{a\}$ and $\{c\}$ solution frequencies which are greater or equal to n.

Exercise 3.4.1 Suppose the data obtained in an experimental investigation are fully congruent with the expectations concerning solution frequencies.

[6] The investigation was conducted in 1988 by B. Hierholz at the University of Heidelberg under direction of the first author.

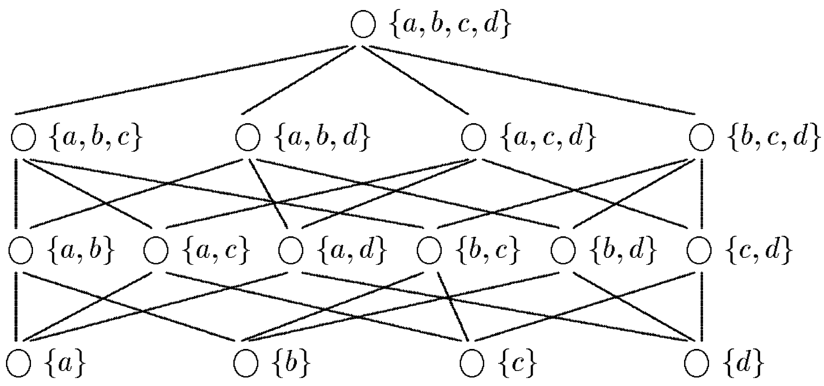


Figure 3.12. Hasse diagram for the problems identified with the elements of the component space $\mathcal{F}_C$.

Does this fact indicate that the data are also congruent with the surmise relation? Give reasons for your answer. □

Method For the investigation, four motives were selected and combined as shown in Figure 3.12. These motives are once again 'fork', 'pin', 'deflection' and 'guidance'. The combinations of these four motives form 15 problems with the problem containing all four motives being the most difficult one. A complete list of the problems is given in Table 3.6 in the appendix. Figure 3.13 shows a problem with the motives 'fork', 'pin' and 'deflection' (for the solution see problem 4 of Table 3.6 in the appendix). The positions of Figure 3.11 are examples for the problems with one motive.

These problems were presented to 13 subjects who are all members of the

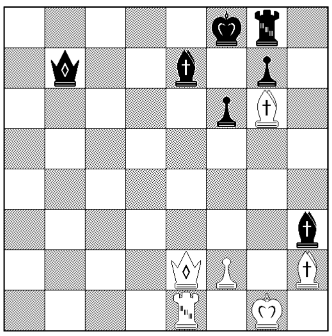


Figure 3.13. Example for a problem (motives a, b, c).

chess club in Ladenburg, Germany.

First, the subjects were asked to read the instructions for the experimental procedure, then they were permitted to begin working on the problems. Each problem was printed on a single card as a diagram (see Figures 3.10, 3.11 and 3.13). The subjects had to write down the solution in the usual form.

The time, needed for the solution was controlled by the subjects themselves with the aid of a chess clock. There was no time limit. The subjects were asked only to answer 'as accurately and as quickly as possible'. The problems were presented in the order of hypothesized difficulty, so problem $\{a, b, c, d\}$ with four motives was the first to be presented and the one motive problems $\{a\}$, $\{b\}$, $\{c\}$, and $\{d\}$ were the last to be presented.

Results Figure 3.14 shows the total solution frequencies for all problems. We can see that several contradictions to our assumption concerning solution frequencies arose. Problem $\{c, d\}$, for example, was solved less frequently than the (hypothetically) most difficult problem $\{a, b, c, d\}$. Figure 3.15 (solid circles denote correct answers, open circles denote incorrect answers) demonstrates structures for both an 'inconsistent' (1) and a 'consistent' subject (2).

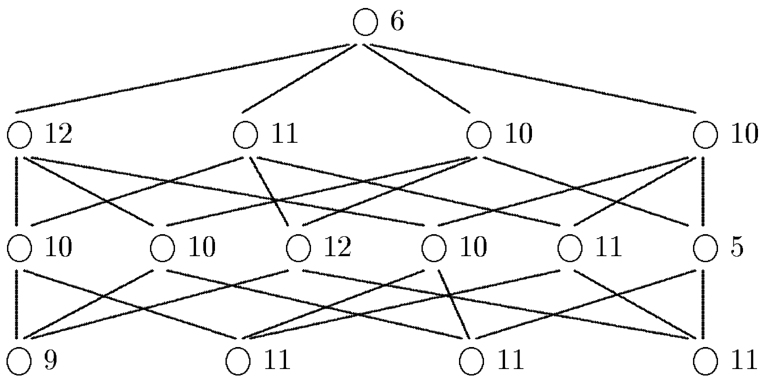


Figure 3.14. Chess problems: solution frequencies.

Table 3.1 shows the results for the 13 subjects, '+' stands for a correct and '–' for an incorrect answer. Note that in this table the problems are named by the symbols for the associated motives, i. e. the problem identified with $\{a, b, c, d\}$ is named 'abcd'.

If we look at the hypothetical problem structure (Figure 3.12), we can see that the hypothesis holds only for the three subjects (2, 4, 5), who solved all problems and for subject 3 who failed only in solving problem $\{a, b, c, d\}$. Subjects 7 and 10 each show inconsistencies for only one problem.

Discussion The results clearly contradict our deterministic hypothesis, since the response patterns of only four subjects agree with it.

The reasons for the unsatisfactory results may be found both in the theoretical approach and the experimental design. First of all, the difficulty of the chess problems is probably not solely influenced by the type and number of included motives. An investigation by Albert, Schrepp, and Held (1993) showed that taking the sequence of motives within problems into consideration can contribute to a more adequate problem structure.

Another problem is common to investigations dealing with chess playing is that the work on chess problems requires great concentration over a large period of time. Thus we suspect that the order of problem presentation (beginning with $\{a, b, c, d\}$) might not have been the best choice.

The experimental setting as a group experiment and the lack of a limit on solution times may have caused a decrease in motivation with some of the

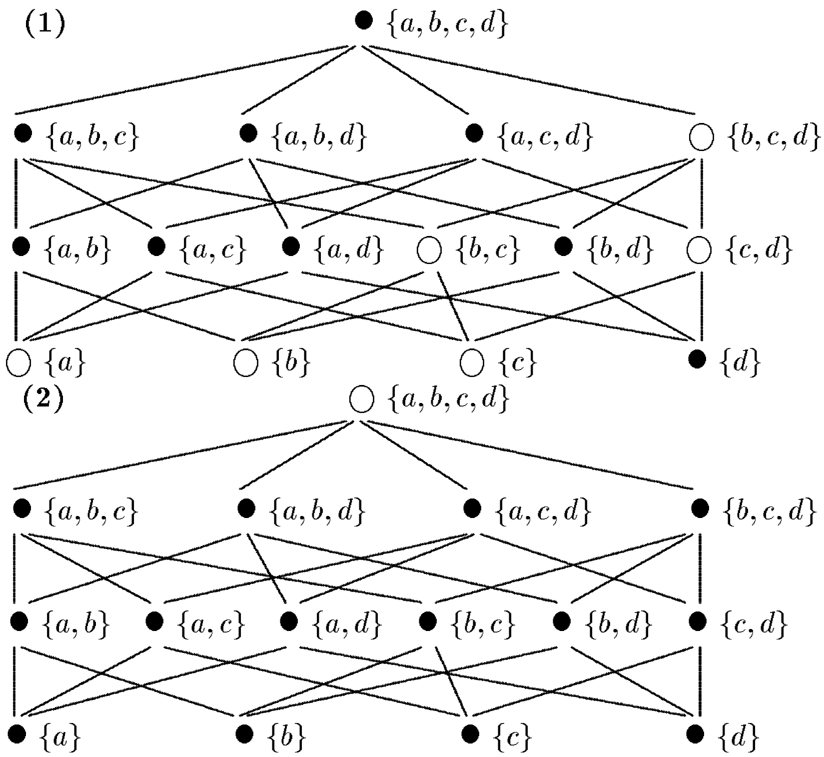


Figure 3.15. Chess problems: individual results of two subjects. Solid circles denote correct answers, open circles denote wrong answers.

Table 3.1. Chess problems: correct and incorrect answers

Sub-	Problem													
ject	abcd	bcd	abc	acd	abd	bc	ad	bd	ac	cd	ab	d	c	b	a
1	+	−	+	+	+	−	+	+	+	−	+	+	−	−	−
2	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
3	−	+	+	+	+	+	+	+	+	+	+	+	+	+	+
4	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
5	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
6	+	+	+	+	+	+	+	+	+	−	+	−	+	+	−
7	−	+	+	−	+	+	+	+	−	+	+	+	+	+	+
8	−	+	+	+	−	+	+	+	+	−	−	+	+	+	−
9	−	−	−	+	−	+	−	−	−	−	−	+	−	+	+
10	−	+	+	+	+	+	+	+	+	−	+	+	+	+	+
11	+	−	+	+	+	+	+	+	+	−	+	+	+	+	−
12	−	+	+	−	+	−	+	−	+	−	+	−	+	+	+
13	−	+	+	−	+	−	+	+	−	−	−	+	+	−	+

subjects who required longer solution times. Further, the problem set was possibly too inhomogeneous for some of the subjects. The degree of familiarity with a specific problem type may also influence the solution process.

In the investigation of Albert, Schrepp, and Held mentioned above, these problems were taken into account. A computerized experimental laboratory setting was used. Further, the uniqueness of motive assignment was optimized. Due to these improvements the results of this investigation are much more conclusive than the ones reported here.

3.4.2 Continuing a series of numbers

Our second empirical example deals with a type of task which is commonly found in diagnostic instruments in psychology.[7] It is typical vor inductive reasoning. A series of numbers constructed according to an algebraical rule is to be continued by one or more numbers. Subjects are required to infer the rule from the number series presented and to calculate the missing number with the help of this rule. The following example demonstrates a very simple task:

$$30\ 32\ 36\ 44\ 60\ \ldots ?$$

One possible rule is: $x_n = x_{n-1} + 2^n$. Of course, we can find other formulas which correspond to the example, e. g. $x_n = 3x_{n-1} - 2x_{n-2}$, where x_n is the number, we are trying to find, x_{n-1} is the preceding number (here: 60), and so on. Our example shows that both formulas use preceding elements of the given series for the calculation of x_n. We will call the number of immediate predecessors which are used for the solution of the problem *level of recursion.* The first formula has recursion level '1', the second is of level '2'. Krause (1985) used this type of recursively connected number series in an investigation of mental processes and rule detection. He attempted to classify the various methods subjects use to solve this type of problem.

Some types of number series problems possess properties which make them suitable for our component based method of problem construction. The level of recursion is one of them. Generally, we assume that the following cognitive demands are covered by number series problems: (1) the subject has to recognize properties and regularities of the presented sequence (e. g. the level of recursion), (2) a hypothesis concerning the underlying rule has to be established, applied and tested.

Problem construction and hypothesis Number series problems are extremely variable, so the question is, what types of components can be combined in which ways. In this investigation, three distinct components M_1, M_2, M_3 were used. Their attributes are shown in Table 3.2. Concerning attributes b_2 and c_2, we must note that the definition of the factors $f = 1$ and $g = 0$ is included for 'technical' reasons: although a recognition of a multiplicative or additive factor is not necessary for a solution of the problems which are characterized by b_2 or c_2, giving such 'zero values' is appropriate for a complete problem definition. We assume that a linear order is defined on the attributes

[7]The investigation was conducted in 1989 by P. Hellriegel, J. Ptucha and M. Wölk at the University of Heidelberg under direction of the first author.

Table 3.2. Number series: problem components

Components	Attributes		
$\mathbf{M_1}$	$\mathbf{a_1}$ level of rec.: 3	$\mathbf{a_2}$ level of rec.: 2	$\mathbf{a_3}$ level of rec.: 1
$\mathbf{M_2}$	$\mathbf{b_1}$ multiplicative factor $\mathrm{f} > 1 \wedge \mathrm{f} \in \mathrm{N}$	$\mathbf{b_2}$ multiplicative factor $\mathrm{f} = 1$	
$\mathbf{M_3}$	$\mathbf{c_1}$ additive factor $\mathrm{g} > 1 \wedge \mathrm{g} \in \mathrm{N}$	$\mathbf{c_2}$ additive factor $\mathrm{g} = 0$	

of each component. The Hasse diagrams of Figure 3.16 (left) illustrate this fact. This assumption means that, for example, recursion level 3 makes a problem

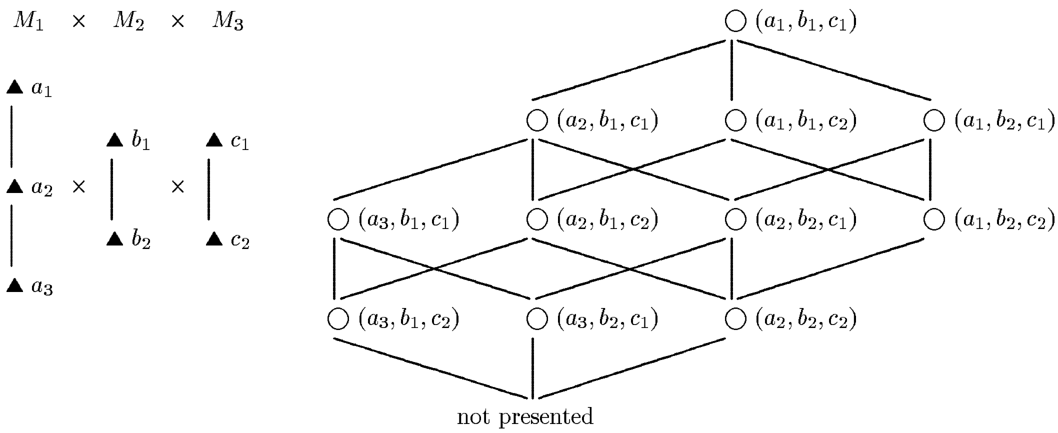


Figure 3.16. Number series: Order of attributes and problems.

more difficult than recursion level 2, or the existence of a multiplicative factor which is greater than 1 provides more complication than factor 1.

Now we must find a problem construction rule for these components. In Section 3.3.2, we demonstrated how product formation can be applied to sets of components. We will apply this rule to M_1, M_2, M_3. The product $M_1 \times M_2 \times M_3$ provides twelve combinations of attributes of the type (a_n, b_n, c_n). We will call the set of these combinations *problem set* Q_t.

The next step is the application of the *coordinatewise ordering rule* as described in Section 3.3.2. This leads to the structure of the twelve problems, where problem (a_1, b_1, c_1) is assumed to be the most difficult and (a_3, b_2, c_2) the simplest. Table 3.3 shows the complete problem set. On the right side of Figure 3.16, we can see the problem structure.

The construction and ordering of the number series problems which was based on product formation and coordinatewise ordering is the motivation for the investigated hypothesis:

> *If a subject solves any problem included in Q_t, then this subject is capable of solving all subordinate problems with respect to the postulated problem structure.*

Method As already mentioned, the problems were constructed from three components: recursion level, multiplicative constant and additive constant. The multiplicative constant is either 2 or 0, the additive constant is always a single- or two-digit element of $\mathbb{N}_0$, the maximal recursion level is 3. To avoid successful guessing of the solution, all solutions are numbers greater than 100. Table 3.3 shows all calculation rules and the corresponding problems.

Table 3.3. Number series: calculation rules and problems

Attributes		c_1	c_2
a_1	b_1	$x_n = 2x_{n-3} + x_{n-2} + x_{n-1} + 4$ $1, 5, 9, 20, 43, 85 \Rightarrow 172$	$x_n = 2x_{n-3} + x_{n-2} + x_{n-1}$ $6, 6, 7, 25, 44, 83 \Rightarrow 177$
	b_2	$x_n = x_{n-3} + x_{n-2} + x_{n-1} + 1$ $16, 16, 17, 50, 84, 152 \Rightarrow 287$	$x_n = x_{n-3} + x_{n-2} + x_{n-1}$ $26, 34, 41, 101, 176, 318 \Rightarrow 595$
a_2	b_1	$x_n = x_{n-2} + 2x_{n-1} + 2$ $1, 4, 11, 28, 69, 168 \Rightarrow 407$	$x_n = 2x_{n-2} + x_{n-1}$ $5, 11, 21, 43, 85, 171 \Rightarrow 341$
	b_2	$x_n = x_{n-2} + x_{n-1} + 5$ $12, 17, 34, 56, 95, 156 \Rightarrow 256$	$x_n = x_{n-2} + x_{n-1}$ $25, 34, 59, 93, 152, 245 \Rightarrow 397$
a_3	b_1	$x_n = 2x_{n-1} + 1$ $7, 15, 31, 63, 127, 255 \Rightarrow 511$	$x_n = 2x_{n-1}$ $4, 8, 16, 32, 64, 128 \Rightarrow 256$
	b_2	$x_n = x_{n-1} + 13$ $33, 46, 59, 72, 85, 98 \Rightarrow 111$	$x_n = x_{n-1}$ $113, 113, 113, 113, 113 \Rightarrow 113$ (not presented!)

Problem (a_3, b_2, c_2) was not used for the investigation, because it is not really a 'problem' and would possibly have confused the subjects. Before the remaining eleven problems were presented, the eighteen subjects, who took part in the investigation, were asked to read instructions which introduced the problem type. The subjects were also told that the only mathematical operations to be used were addition and multiplication. Then they were asked to solve three simple example problems.

The problems of our structure were then presented in a randomized order. The subjects had to write down the solution on a sheet of paper. If the solution was not given within seven minutes, the next problem was presented. Subjects who either gave the wrong solution or wanted to 'give up' before the seven minutes had expired, were asked to go on thinking about the problem. After the last problem the subject was asked for the rules he or she used for solving the problems.

Results Figure 3.17 shows the total solution frequencies for each problem in the structure. In Figure 3.17, we can see that two problems were solved by all

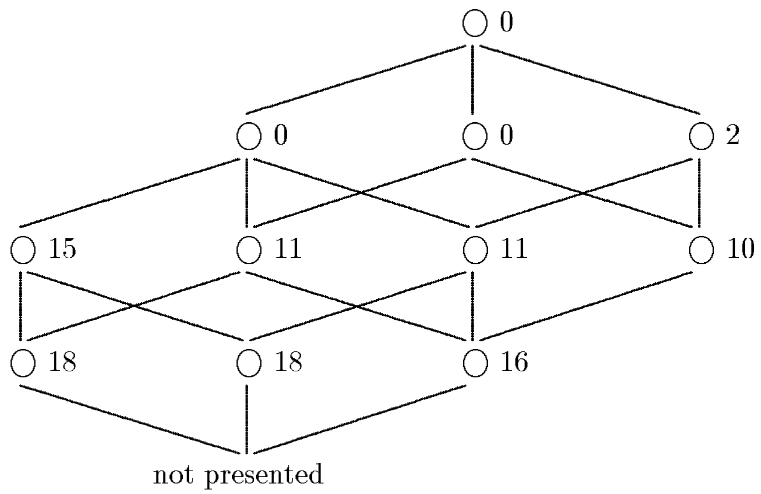


Figure 3.17. Number series: solution frequencies.

subjects and three problems were solved by no subject. In not one case was a subordinate problem solved less frequently than a superordinate problem.

The results of only two subjects (13 and 15) do not correspond to the hypothesis. In these cases, the set of solved problems is not an element of the postulated quasi-ordinal knowledge space. The result of one of these subjects (15) and that of the 'non-contradicting' subjects 1 and 5 can be seen in Figures 3.18 (1) and (2). Table 3.4 shows the results for the 18 subjects; '+' stands for a correct and '–' for an incorrect answer.

Discussion The results of this investigation show that the hypothetical conclusions we drew about the coordinatewise ordering rule were rather accurately. Our structure corresponds to a total number of 49 knowledge states, where the cardinality of the power set of the problem set is equal to 1024. Therefore, only 4.8 % of the potential response patterns are states with respect to the structure.

An alternative and 'more economical' theory for the data could be stated by a lexicographic order on the problem set. For this purpose we assume that component M_1 (recursion level) is the 'most important' component, M_2 (multiplicative factor) the second most important and M_3 the least important component. With respect to this order, none of the observed response patterns agrees with a state. In this case, only 11 knowledge states are assumed to exist — this is 1.1 % of the potential response patterns. Although the lexicographic order is much more 'restrictive' than the coordinatewise order, these results may also be a product of the assumption concerning the importance of the components. The analysis of other possible lexicographic orders is left to the reader as an exercise.

We assumed that a subject, who is able to solve a problem, will use one

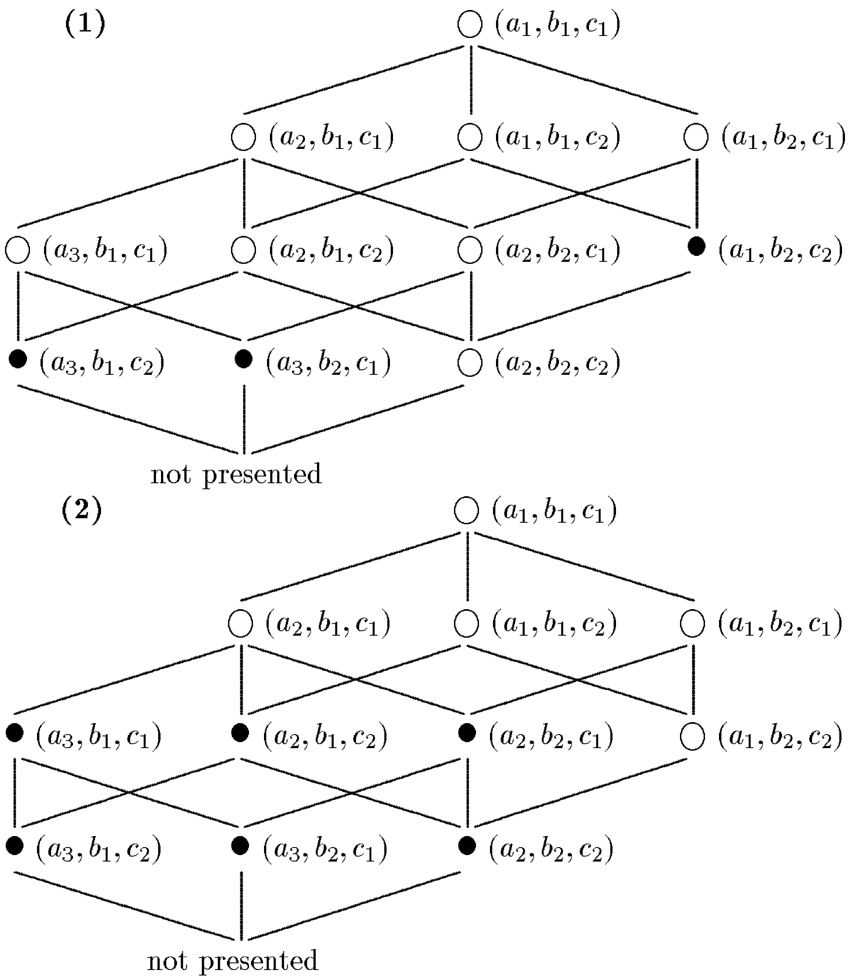


Figure 3.18. Number series: individual results of two subjects. Solid circles denote correct answers, open circles denote incorrect answers.

particular calculation rule. This is not always realistic, because for every number series problem, alternative solutions can be found. These alternatives are frequently also plausible. The next section deals with alternative solutions.

Ambiguity of number series problems In correspondence with problem construction, the calculation rule for the series $5, 11, 21, 43, 85, 171, \ldots ?$ is $x_n = 2x_{n-2} + x_{n-1}$. This is problem (a_2, b_1, c_2). We can easily see that the rule $x_n = 2x_{n-1} + (-1)^n$ will also provide a correct solution.

Although the subjects were told that only positive constants are to be added in the problems, we cannot exclude the possibility that a subject will use such an alternative rule. In the reported investigation, eleven subjects provided a correct answer, whereby eight subjects used the alternative rule as shown above.

We can see that the construction and ordering of number series problems must be based on an exact analysis of the problems' uniqueness, especially if the problems are constructed from components which include principles of so-

Table 3.4. Number series: correct and incorrect answers

Subject	Problem										
	(a_2, b_2, c_2)	(a_1, b_2, c_2)	(a_3, b_1, c_2)	(a_2, b_1, c_2)	(a_1, b_1, c_2)	(a_3, b_2, c_1)	(a_2, b_2, c_1)	(a_1, b_2, c_1)	(a_3, b_1, c_1)	(a_2, b_1, c_1)	(a_1, b_1, c_1)
1	+	−	+	+	−	+	+	−	+	−	−
2	+	−	+	+	−	+	−	−	+	−	−
3	+	+	+	+	−	+	+	−	+	−	−
4	+	+	+	+	−	+	+	−	−	−	−
5	+	−	+	+	−	+	+	−	+	−	−
6	+	+	+	+	−	+	+	−	+	−	−
7	+	+	+	+	−	+	+	−	+	−	−
8	+	−	+	+	−	+	−	−	+	−	−
9	+	+	+	+	−	+	+	−	+	−	−
10	+	+	+	−	−	+	−	−	+	−	−
11	+	+	+	−	−	+	+	−	+	−	−
12	+	+	+	−	−	+	+	+	+	−	−
13	−	−	+	+	−	+	−	−	+	−	−
14	+	+	+	−	−	+	+	+	−	−	−
15	−	+	+	−	−	+	−	−	−	−	−
16	+	−	+	−	−	+	+	−	+	−	−
17	+	−	+	−	−	+	−	−	+	−	−
18	+	−	+	+	−	+	−	−	+	−	−

lution. Korossy (1990) examined the phenomenon of ambiguous number series problems with special reference to the case of linear recursive series. He developed a method, which allows the uniqueness of the solution to be determined. This method is based on the theory of linear equation systems. One of the main results of his study is that only heavy restrictions on the domains of the recursive formulas lead to less ambiguous ranges for the solutions.

As an overall conclusion, we can say that it is impossible to construct number series problems which have only one correct solution. However it is possible to minimize the number of alternatives to a degree which allows one to work with this type of problem. Further, if the manner in which an ambiguous problem has been solved is known, it may be possible to infer which of the assumed cognitive demands has been mastered by the subject.

Exercise 3.4.2 With respect to the coordinatewise order of the number series problems, two response patter have been observed which do not correspond to our hypothesis. For each of these patterns identify the problems which do not conform to the hypothesis. □

Exercise 3.4.3 Draw the Hasse diagram for the lexicographic order corresponding to the hypothesis of component importance stated above (M_1: most important; M_2: second most important; M_3: least important). For each subject determine the number of correct or incorrect answers given which do not comply with the hypothesis. □

EXERCISE 3.4.4 Consider the following assumptions concerning the importance of the components of our number series problems:

most important	*second most important*	*least important*
M_2	M_1	M_3
M_1	M_3	M_2

Draw the Hasse diagrams for the lexicographic orders corresponding to these assumptions. Check the agreement of the resulting knowledge states with the data of the number series experiment. □

3.5 Relation to decision theory

As we have already noted in Section 3.3, the rules introduced for problem construction are closely related to the concepts of *choice heuristics* from decision theory. In the following, we will demonstrate that comparable mathematical structures can be relevant for different areas of psychological research (i. e. *psychology of knowledge* and *decision theory*). Further it will be shown that an area such as decision theory provides approaches which are also suitable for application in the field of knowledge assessment. However, it will be illustrated that not each of the known choice heuristics can be applied to knowledge assessment.

Therefore, we will give a brief introduction to some basics of choice heuristics[8] and we will illustrate parallel aspects of the formal description of choice behavior and the ordering of systematically constructed problems. We will focus on the preference relation defined on a set of alternatives
indexalternative, -s, and on the surmise relation defined on a set of problems. We will first take a look at a typical decision problem.

EXAMPLE 3.5.1 Consider, someone would like to buy a new car. The dealer offers five different models with attributes as shown in Table 3.5. The customer now has to decide which of these five cars is the most suitable. □

With the help of this example we will introduce some of the basic concepts for the representation of a decision task.

First we need a set $A = \{a, b, c, d, \ldots\}$ of *alternatives*. In our example this is $A_{ex} = \{\mathit{car\,1},\ \mathit{car\,2},\ \mathit{car\,3},\ \mathit{car\,4},\ \mathit{car\,5}\}$. These alternatives may be

[8] A detailed introduction to the mathematical aspects of decision theory can be found in Fishburn (1972). A number of different choice heuristics is formalized in Aschenbrenner (1980,1981).

Table 3.5. Examples for alternatives described by the attributes on five dimensions

Alternatives	*Dimensions*				
	V_{max}	fuel consumption	length	luggage space	purchase price
Car 1	100 mph	low	4.2 m	$1.6\,m^3$	\$ 22,000
Car 2	140 mph	medium	4.5 m	$1.4\,m^3$	\$ 24,000
Car 3	90 mph	low	4.1 m	$1.2\,m^3$	\$ 18,000
Car 4	120 mph	high	5.0 m	$1.9\,m^3$	\$ 21,000
Car 5	115 mph	medium	4.8 m	$1.6\,m^3$	\$ 30,000

characterized by *dimensions*. $D = \{d_1, d_2, \ldots, d_i, \ldots, d_n\}$ is the set of dimensions. The set $D_{ex} = \{V_{max}$, *fuel consumption, length, luggage space, purchase price*$\}$ is known from Example 3.5.1. Every dimension d_i is a set of *attributes*: $d_i = \{d_{i1}, d_{i2}, \ldots, d_{in}\}$. The dimension V_{max}, for example, has the attributes *100 mph, 140 mph, 90 mph, 120 mph*, and *115 mph* as elements. An alternative x can be treated as n-tuple: $x = (x_1, x_2, \ldots, x_n)$, where $x_1 \ldots x_n$ are attributes of the dimension d_i which belongs to alternative x. For example: *car 2 = (110 mph, medium fuel consumption, 4.5 m, 1.4 m*^{3}*, \$ 24,000).*

Let us take a look at a decision experiment in which all possible pairs of alternatives (including the pairs which consist of identical alternatives) in all possible sequences are presented to the subjects. The question to the subjects is: "Is the first of the presented alternatives at least as attractive to you as the second alternative?"

The fact that alternative x is preferred to alternative y is denoted by the *preference relation* $\mathcal{P}_R$. Therefore, we can write $x\mathcal{P}_R y$ for the preference of alternative x to y regarding rule R. The properties of this relation $\mathcal{P}_R$ depend on the *choice heuristic* R used that establishes the preference relation and on the manner in which the subject is asked (see Exercise 3.5.2). Because the preference of alternatives is based on the preference of attributes, the application of a choice heuristic requires the existence of a 'preference order' on each dimension.

Two choice heuristics have, in principle, already been introduced in Section 3.3: the *dominance rule* and the *lexicographic rule*. The definitions of these rules are analogous to those given in Section 3.3 (p. 88 and p. 90). We can therefore omit them here.

Exercise 3.5.1 Take a customer who decides according to the dominance rule. If we consider comparisons between car 1 and car 2 and between car 2 and car 4, which cars will be preferred by this customer? □

Exercise 3.5.2 Consider a choice experiment in which questions of the following type are asked: "Is the first of the presented alternatives *more* attractive to you than the second alternative?" In addition is assumed that the subjects answer in agreement with the dominance rule. Which property of a quasi-order may as a result of the manner of questioning not be met by the resulting preference relation? □

Another well known choice heuristic is the majority rule.

DEFINITION 3.5.1 A choice heuristic M is a *majority rule* if and only if

$$x\mathcal{P}_M y \iff \mathrm{card}(X_+) \geq \mathrm{card}(Y_+),$$

with x and y alternatives, X_+ the set of dimensions with preferred attribute in x and Y_+ the set of dimensions with preferred attribute in y. $card(X_+)$ is the cardinality of the set X_+. □

EXERCISE 3.5.3 We refer to Table 3.5 on page 106. Suppose that the alternatives x ($\widehat{=}$ car 1) and y ($\widehat{=}$ car 2) are to be compared by a customer who prefers small fast cars which have low fuel consumption, are equipped with a large luggage space, and are inexpensive. Further, we assume that the customer bases his or her decision on the majority rule. Which of the alternatives x and y will be preferred by the customer?

The customer then compares alternatives y ($\widehat{=}$ car 2) and z ($\widehat{=}$ car 4). Which one will be preferred in this case?

Next we will determine whether $\mathcal{P}_M$ is a transitive relation. For this purpose suppose that $\mathcal{P}_M$ is transitive. Which decision by the customer can as a result of this assumption be inferred from the decisions stated above? Check the inferred decision by comparing the attributes provided in Table 3.5. What conclusion can be drawn concerning the transitivity of $\mathcal{P}_M$? □

If attributes of different components are comparable (i. e., if it is possible to state that attribute a_i is more difficult than attribute b_j), then rules such as the *minimax heuristic* and the *maximax heuristic* (see Svenson, 1979; Huber, 1982) may be appropriate for knowledge assessment.

Decision theory offers the advantage of a well developed inventory of properly formalized choice heuristics which are potential candidates for problem ordering rules. As general principle, choice heuristics do not necessarily impose a surmise relation on a problem set. It can, however, be determined whether the properties of such a relation are met.

3.6 Summary

In this article, methods for the generation of ordered problem sets are introduced. Our theoretical results are motivated by the theory of knowledge spaces which was introduced by Doignon and Falmagne (1985). A basic concept of this theory is the *surmise relation*, a transitive and reflexive binary relation defined on a set of problems. By this relation, a set of *knowledge states* (i. e. subsets of the problem set) is determined. Although the step from surmise relations to the more general concept of *surmise systems* is the main achievement of the theory of knowledge spaces, we restrict our considerations concerning this theory to surmise relations.

The question we are focusing on is how surmise relations can be derived from a systematically constructed set of problems. Both problem construction and problem ordering are based on *domain specific theories* which are prerequisites for the definition of *problem components* and the establishment of problem structures which are derived from these components.

Problem components may, for example, be operations necessary for a problem solution or subgoals during the solution process.

The methods introduced for the establishment of ordered problem sets are in principle known from elementary ordering theory: *set inclusion* and *componentwise ordering of product sets*.

In the experimental investigations, we show applications of the methods introduced for problem construction and problem ordering. The first investigation deals with the domain of *problem solving*. Chess problems are constructed on the basis of *motives*. These tactical elements of the game of chess are viewed as subgoals for the solution process. A surmise relation on the problem set is established by inclusion of motive sets. The second investigation belongs to the domain of *inductive reasoning* with the solution of number series problems. Problem construction is done by product formation. The surmise relation is a result of the componentwise ordering of products. In this case, the components are parts of the *rules* which have to be found for problem solution.

Additionally, we show that parallels exist between the ordering principles introduced and *choice heuristics* known from decision theory. The applicability of these heuristics to knowledge assessment is discussed.

In the meantime, further principles for the establishment of knowledge structures which are based on problem components or skills have been developed. Lukas and Micka (1993) consider the assignment of skills to elementary chess-endgame problems. In Lukas (1991) *misconceptions* for the solution of problems on basic electricity circuits are modeled by *information systems*. This approach also focuses on *incompatibility relations* between skills. These results are also important for the definition of component based problems as introduced in this article.

The investigations of Korossy (1993) are based on modeling *competencies* and *performances* within assessment processes. The domain under investigation is the field of *geometric constructions and calculations*. In Held (1992, 1993), (quasi-ordinal) knowledge spaces are derived from component based problems on *elementary combinatorics and probability calculus*. Some of the theoretical approaches introduced there are extensions of the methods of this paper. Further, the assignment of 'problem demands' to problem components is discussed.

Albert, Schrepp and Held (1993) provide the principle of *sequence inclusion* for ordering motive based chess problems. This method is an extension of set inclusion which has been used for the chess experiment reported here.

Appendix

Table 3.6. Complete list of chess problems.

Number	Type	Position	Solution	Motives
1	abcd	*White*: Ka7 Qh3 Re5 Nd6 *Black*: Kh8 Qg6 Rg8 Bf7 Ph7	1. Rg5 Qf6 2. Qc3 Qc3: 3. Nf7 mate	deflection, guidance, pin, fork
2	bcd	*White*: Kh2 Bf3 Nh5 Pg3,g7 *Black*: Kh7 Qe6 Ph3	1. g8Q+ Kg8: 2. Bd5 Qd5: 3. Nf6+	guidance, pin, fork
3	abc	*White*: Kg1 Qe2 Re1 Bg6,h2 Pf2 *Black*: Kf8 Qb7 Rg8 Be7,h3 Pg7,f6	1. Qe7:+ Qe7: 2. Bd6 Qd6: 3. Re8 mate	deflection, guidance, pin
4	acd	*White*: Kg1 Qc2 Rf2 Bb1 Nf8 Pb2,c6,g2 *Black*: Kd8 Qg7 Rd6 Be4 Nd3 Pb7,e7	1. cb: Bb7: 2. Qd3: Rd3: 3. Ne6+	deflection, pin, fork
5	abd	*White*: Ka2 Qf4 Be3 Pb2,b3,h3,c7 *Black*: Ka5 Qe7 Nb6 Pa6,b5,c5,b4,h4	1. Qb4:+ cb: 2. Bb6:+ Kb6: 3. c8N+	deflection, guidance, fork
6	bc	*White*: Kf1 Qa6 Re1 Nh3 Pg2,f2,d4 *Black*: Ke8 Qd6 Rh8 Nc6 Pe6,f7,g7	1. d5 Qd5: 2. Qa8+ arbitrary 3. Qc6:+/Qd5:/Qh8:	 guidance, pin
7	ad	*White*: Kd6 Nf5 Pe7 *Black*: Kf7 Ng4 Ph7	1. Nh6+ Nh6: 2. Ke2 arbitrary 3. e8Q	 deflection, fork
8	bd	*White*: Kc6 Ba6 Ne6 Pe4 *Black*: Ke8 Pe7,h2	1. Be2 h1Q 2. Bh5+ Qh5: 3. Ng7+	 guidance, fork
9	ac	*White*: Kh2 Bb6 Pf3,g2 *Black*: Kh4 Rc2 Ph7,h5,g5	1. Bc7 Rg2:+ 2. Kg2: arbitrary 3. Bd8/f2 mate	 deflection, pin
10	cd	*White*: Kf3 Rc6 Ne5 Pg5 *Black*: Kg8 Rd4 Be7 Pf4	1. Rc8+ Kg7 2. Rc7 Kf8 3. Ng6+	 pin, fork
11	ab	*White*: Kh2 Qd1 Re2 Pd7,f2,h4 *Black*: Kg8 Qb5 Rd8 Pa4,g7,h7	1. Re8+ Re8: 2. Qd5+ Qd5: 3. deQ mate	 deflection, guidance
12	d	*White*: Kf2 Ne8,f7 Pd3 *Black*: Kf4 Re6 Pd4	1. Nc7 Rg6/c6 2. Nd5+ arbitrary 3. Ne7+/Ne5+	 fork
13	c	*White*: Kf1 Qh6 Re1 Bf6 *Black*: Kd8 Qa4 Rb7 Be7 Pf7	1. Qf8+ Qe8 2. Rd1+ Rd7 3. Be7:+	 pin
14	b	*White*: Kb5 Qd7 Pa7 *Black*: Ka8 Bb4 Pa2,c2	1. Kb6 Ba5+/c5+ 2. Ka6/c6 arbitrary 3. Qb7/c6 mate	 guidance
15	a	*White*: Kd8 Bb7 Pc7 *Black*: Kd6 Be6 Pf5,a6	1. Bc8 Bd5 2. Bf5: Bb7 3. Be4	 deflection

Problem 8 by Maiselis and Judowitsch (1966); problem 10 by Geisdorf (1984); problem 12 by Chéron (1960); problem 14 by Speckmann (1958).

References

Albert, D. (1989) *Knowledge assessment: choice heuristics as strategies for constructing questions and problems.* Paper read at the 20th European Mathematical Psychology Group Meeting, Nijmegen.

Albert, D. (1991) *Principles of problem construction for the assessment of knowledge.* Paper read at the 2nd European Congress of Psychology, Budapest.

Albert, D., Schrepp, M., & Held, T. (1993). Construction of knowledge spaces for problem solving in chess. In G. Fischer, & D. Laming (Eds.), *Contributions to Mathematical Psychology, Psychometrics, and Methodology.* New York: Springer.

Aschenbrenner, K. M. (1980). *Eingipflige Bevorzugung: Aufgabencharakteristika und Entscheidungsheuristiken als Bedingungen ihrer Entstehung.* [Single-peaked preference: problem characteristics and decision heuristics as conditions for its emergence.] Freiburg: Hochschulverlag.

Aschenbrenner, K. M. (1981). Efficient Sets, Decision Heuristics, and Single-Peaked Preferences. *Journal of Mathematical Psychology*, **23**(3), 227–256.

Birkhoff, G. (1937). Rings of sets. *Duke Mathematical Journal*, **3**, 443–454.

Birkhoff, G. (1973). *Lattice Theory*, 3rd edn. Providence: American Mathematical Society.

Chéron, A. (1960). *Lehr- und Handbuch der Endspiele Bd. 1.* [Text- and handbook of endgames Vol. 1.] Berlin: S. Engelhardt-Verlag.

Davey, B. A., & Priestley, H. A. (1990). *Introduction to lattices and orders.* Cambridge: Cambridge University Press.

Doignon, J.-P., & Falmagne, J.-C. (1985). Spaces for the assessment of knowledge. *International Journal of Man-Machine Studies*, **23**, 175–196.

Falmagne, J.-C., Koppen, M., Johannesen, L., Villano, M., & Doignon, J.-P. (1990). Introduction to knowledge spaces: how to build, test and search them. *Psychological Review*, **97**(2), 201–224.

Fishburn, P. C. (1972). *Mathematics of decision theory.* Princeton: Princeton University Press.

Fishburn, P. C. (1974). Lexicographic orders, utilities and decision rules: a survey. *Management science*, **20**, 1442–1471.

Geisdorf, H. (1984). *Der Schachfreund, Bd. 1: Auf den Flügeln der Kunst.* [The Chess buff.] Mannheim: Selbstverlag H. Geisdorf.

de Groot, A. D. (1965). *Thought and choice in chess.* The Hague : Mouton & Co.

Guttman, L. A. (1947). A basis for scaling qualitative data. *American Sociological Review*, **9**, 139–150.

Guttman, L. A. (1950). The basis for scalogram analysis. In S. A. Stouffer, L. A. Guttman, E. A. Suchman, P. F. Lazarsfeld, S. A. Star, & J. A. Clausen (Eds.), *Volume 4: Measurement and prediction* (pp. 60–90). London: Princeton University Press.

Held, T. (1992). *Systematische Konstruktion und Ordnung von Aufgabenmengen zur elementaren Wahrscheinlichkeitsrechnung.* [Systematical construction and ordering of problem sets of elementary stochastics.] Paper read at the 34th Tagung experimentell arbeitender Psychologen, Osnabrück, FRG, 12.-16. April 1992.

Held, T. (1993). *Establishment and empirical validation of problem structures based on domain specific skills and textual properties — A contribution to the 'Theory of Knowledge Spaces'.* Dissertation Universität Heidelberg.

Huber, O. (1982). *Entscheiden als Problemlösen.* [Deciding as problem solving.] Bern: Huber.

Korossy, K. (1990). *Zum Problem der eindeutigen Lösbarkeit bei linear-rekursiven Zahlenfolgen-Aufgaben.* [On the problem of unique solvability with linear-recursive number series.] Arbeitsbericht Universität Heidelberg.

Korossy, K. (1993). *Modellierung von Wissensstrukturen in Hinblick auf eine ef-*

fiziente Wissensdiagnose und Wissensvermittlung. [Modeling knowledge structures with regard to efficient knowledge diagnosis and knowledge impartment.] Dissertation Universität Heidelberg.

Krause, B. (1985). Zum Erkennen rekursiver Regularitäten. [Concerning detection of recursive regularities.] *Zeitschrift für Psychologie*, **193**, 71–86.

Lukas, J. (1990). *Algorithmen zur "Berechnung bestimmter Eigenschaften von Relationen und eine Anwendung auf die Untersuchung von Wissensstrukturen".* [Algorithms for "computation of certain properties of relations and an application to the investigation of knowledge structures.] Arbeitsbericht Universität Heidelberg.

Lukas, J. (1991). *Knowledge structures and information systems.* Paper read at the 22nd European Mathematical Psychology Group Meeting, Vienna.

Lukas, J., & Micka, R. (1993) *Zur Diagnose von Wissen über einfache Schachendspiele: Formale Theorie und empirische Ergebnisse.* [Diagnosis of knowledge about simple chess endgames: formal theory and empirical results.] Paper read at the 35th Tagung experimentell arbeitender Psychologen, Trier, FRG, 4.-8. April 1993.

Maiselis, I. L., & Judowitsch, M. M. (1966). *Lehrbuch des Schachspiels.* [Textbook of chess endgame.] Berlin: Sportverlag.

Speckmann, W. (1958). *Strategie im Schachproblem.* [Strategy in chess problem.] Berlin: de Gruyter.

Svenson, O. (1979). Process description of decision making. *Organizational Behavior and Human Performance*, **23**, 86–112.

Tversky, A. (1969). Intransitivity of preferences. *Psychological Review*, **76**, 31–48.

4 Semantic structures

Jürgen Heller
Institut für Psychologie, Universität Regensburg, Universitätsstr. 31, Gebäude PT, D-93040 Regensburg, Germany
E-mail: juergen.heller@rpss3.psychologie.uni-regensburg.de

4.1 Introduction

Semantics may be defined as the study of meaning. As a part of linguistics, semantics investigates the meaning of lexical items, e. g. words (Lyons, 1968). This paper, however, considers the meaning of lexical items from a psychological point of view. The subject of psycholinguistic research is the behavior of language speakers rather than the language itself. The psycholinguists try to uncover what people know about the meaning of lexical items, or to put it in other words, the psycholinguists investigate the verbal concepts people have in mind. From this viewpoint, a semantic structure represents the knowledge on the meaning of lexical items.

The main contribution to linguistic semantics comes from componential analysis (cf. Lyons, 1968). According to componential analysis the meaning of a word can be factorized into *semantic components*, which are established by considering semantic relations between words. Probably the most important semantic relation is the relation of superordination the converse of which is called hyponymy. In general a lexical item X is a hyponym to a lexical item Y, and Y is a superordinate to X, whenever the statement *'this is an X'* entails the statement *'this is a Y'*. For instance, *animal* is superordinated to *dog*, because *'this is a dog'* entails *'this is an animal'*. Conversely, *dog* is said to be hyponymous (or subordinated) to *animal*. The pairs of nouns *flower* – *rose*, *mammal* – *dog*, *animal* – *mammal* are related by superordination too. These examples taken from the taxonomy of species in flora and fauna demonstrate that superordination is a fundamental ordering principle in hierarchies of concepts.

Psychological studies of semantic structures also try to identify semantic components of lexical items. However, these components result from analyzing (often numerical) measures of similarity of meaning. A review of the relevant literature (see Section 4.2) puts emphasis on a critical discussion of the methods applied. Although they are widely used, they tend to be very strong, and their structural assumptions are usually not tested in the applications. The development of measurement theory (Krantz, Luce, Suppes & Tversky, 1971), however, influenced the way more recently proposed methods are formulated (Tversky, 1977; Colonius & Schulze, 1981). A measurement-theoretic formulation is based on a strict distinction between theory and data. The structural assumptions the theory relies on are explicitly stated, and are based on qualitative observations rather than some numerical indices. They may thus be tested in the applications.

The representation of semantic structures that we will propose below even goes one step farther. We shall argue that a global measure of similarity of meaning does not provide enough information for uniquely identifying semantic components. The assessment of semantic structures is therefore based on semantic generalization, and an empirically determined relation of superordination. Section 4.3 introduces the mathematical notions to which the theory refers. The experimental paradigm used to assess a semantic structure is a modification of a well-known task in mental testing. Section 4.4 formulates a representation theorem for this situation. The assumptions of the proposed join semilattice representation of semantic structures are empirically tested with data from two experiments.

4.2 Previous work

Psychological studies of semantic structures are usually based on measures of similarity of meaning. These (often numerical) measures result — either directly or as derived quantities — from a variety of experimental paradigms (cf. Fillenbaum & Rapoport, 1971). Some examples will be presented below. The data usually consist of a map assigning a number to each pair out of a set of lexical items. This number is interpreted as the corresponding measure of similarity of meaning. By employing certain algorithms to the data, the psycholinguists try to uncover semantic components (called *dimensions*, *features*, or *attributes*) which may explain the similarities. These procedures 'represent' the lexical items as points in some space, and the pairwise similarities as distances between the corresponding points: The higher the similarity, the shorter the representing distance should be. The intuitive properties of distances between points are explicitly stated in the following definition.

Definition 4.2.1 Let S be a set and let $d\colon S\times S\longrightarrow \mathrm{R}$ be a real valued map on pairs of elements of S. d is called a *metric* on S, if for all $x, y, z \in S$

1. $d(x, y) = 0$ iff $x = y$ (minimality)
2. $d(x, y) = d(y, x)$ (symmetry)
3. $d(x, z) \leq d(x, y) + d(y, z)$ (triangle inequality)

A set S together with a metric d on S is called a *metric space*. □

Exercise 4.2.1 Let d be a metric on a set S. Show that d is non-negative, i. e. $d(x, y) \geq 0$ for all $x, y \in S$. □

Remark 4.2.1 Let S be a nonempty set and let the map $d : S \times S \longrightarrow \mathrm{R}$ be defined by $d(x, y) = 0$ if $x = y$, and $d(x, y) = 1$ if $x \neq y$. Then d is a metric on S. This demonstrates that a metric is an abstract notion, which not only captures the idea of interpoint distance in physical space. □

The subsequently described methods are based on the same rationale, since they all represent the stimuli as points of a metric space. As a consequence of the generality of the notion of a metric, however, they look very different at first sight.

4.2.1 Dimensional representations: Semantic space

Osgood, Suci & Tannenbaum (1957) propose the *semantic differential* as a method providing an objective measure of meaning. A semantic differential is a list of pairs of polar adjectives, like *good – bad*, *black – white*, *large – small* or *fast – slow*. The polar adjectives define the extreme positions of rating scales consisting of a number of intermediary categories. The meaning of a given lexical item is determined by a list of ratings on these scales. For instance, the kinship term *father* may be rated to be quite good, slightly white, extremely large, and quite fast. The correlations of these so-called *profiles* are interpreted as measures of the similarity of meaning. By applying factor analysis, they serve to localize the concept as a point in an inner product vector space of low dimensionality. The resulting vector space is called the *semantic space*.

Although the method of the semantic differential had a strong impact on psychological research in the past, it is now of historical interest only. In modern psycholinguistics multidimensional scaling (MDS) techniques are frequently used to obtain a dimensional representation of semantic structures. These techniques try to reproduce an ordering of pairs of stimuli with respect to dissimilarity of meaning by the metric distances between the points of a spatial configuration of low dimensionality (cf. Fillenbaum & Rapoport, 1971). The stimuli are represented as points in a metric (see Definition 4.2.1), mostly Euclidean, vector space.

There are some fundamental problems associated with such vector space 'representations'. First, the procedures provide a spatial configuration to nearly any set of data. Despite the strong restrictions inner product or metric vector spaces impose on the data (e. g. Beals, Krantz & Tversky, 1968; Suppes, Krantz, Luce & Tversky, 1989), a quasi automatical representation results. Except for the discrepancy between the obtained distances and the empirical dissimilarities, no other structural assumptions are tested. So it is not easy, or even impossible, to decide which statements about the distance between two points are empirically meaningful, and which operations on the distances are lawful. For instance, is it meaningful to say that the distance between two points is twice as large as the distance between another pair of points? Or, how can the triangle inequality (see Definition 4.2.1) involving addition be interpreted empirically with ordinal data? Obviously, this situation is not completely satisfying. The above mentioned problems can only be solved by a measurement-theoretic foundation.

4.2.2 Feature representations

For analyzing conceptual similarity data most psycholinguists favor a feature representation. Nearly all of the methods used result in metric feature representations, which can often be depicted by graphs, especially trees.

Miller (1969) presents a psychological method to investigate verbal concepts which provides a feature representation of semantic structures by using hierarchical clustering (Johnson, 1967).

To obtain a measure of dissimilarity of meaning, Miller proposes the method of sorting. Subjects are instructed to sort English nouns (each of which is typed on a card) into piles *'on the basis of similarity of meaning'* (Miller, 1969, p. 170). The subjects are allowed as many piles as they want and they can put as many items as they want in any pile. By determining the number of subjects that do *not* put a pair of items together into one pile, a numerical dissimilarity measure δ is obtained. According to Miller (1969) the dissimilarity measure δ derived in this way satisfies the properties of a metric (see Definition 4.2.1). This is not true in general. There may occur distinct nouns, which all subjects put into one pile such that their dissimilarity vanishes. This is not the case in Miller's data, however.

A representation by a rooted tree is obtained from applying hierarchical clustering to a given dissimilarity measure $\delta\colon S \times S \longrightarrow \mathbb{R}$. A hierarchical clustering is a sequence of partitions of the set S (a partition of a set S is a family of pairwise disjoint subsets whose union is S) ranging from one with single element equivalence classes to one with a single equivalence class containing all elements. To any value $\alpha \geq 0$ of the dissimilarity measure δ a partition P_α is assigned. The equivalence classes of P_α are called *clusters*. Two elements $x, y \in S$ are grouped together in such a cluster, whenever their corresponding dissimilarity is at most α, i. e. $\delta(x, y) \leq \alpha$. This implies that any cluster in the partition P_β, with $\beta \geq \alpha$, is the union of clusters of P_α. These requirements impose restrictions on the dissimilarity measure. The resulting necessary condition is called the *ultrametric inequality* (Johnson, 1967, p. 245):

$$\delta(x, z) \leq \max\{\delta(x, y), \delta(y, z)\} \quad \text{for all} \quad x, y, z \in S. \tag{4.1}$$

The ultrametric inequality implies the triangle inequality, but the converse is not true. Together with minimality and symmetry (see Definition 4.2.1) the ultrametric inequality is sufficient for the representation of a numerical dissimilarity measure by a rooted tree (Johnson, 1967; cf. Exercise 4.3.9).

Example 4.2.1 Consider the set $S = \{\text{father}, \text{mother}, \text{son}, \text{daughter}\}$ of kinship terms with pairwise dissimilarities given by Table 4.1.

It is easily verified that the dissimilarity measure δ satisfies the ultrametric inequality. The hierarchical clustering procedure starts with the finest partition P_0, in which every cluster contains exactly one element: $P_0 = \{\{\text{father}\}, \{\text{mother}\}, \{\text{son}\}, \{\text{daughter}\}\}$. For a dissimilarity of 5, we get the partition $P_5 = \{\{\text{father}, \text{mother}\}, \{\text{son}, \text{daughter}\}\}$, and finally $P_{10} = \{\{\text{father}, \text{mother}, \text{son}, \text{daughter}\}\} = \{S\}$. Figure 4.1 presents the corresponding rooted tree. The singleton sets are assigned to the terminal

Table 4.1. Fictitious dissimilarity data on the set of kinship terms $S = \{\text{father}, \text{mother}, \text{son}, \text{daughter}\}$.

δ	father	mother	son	daughter
father	0	5	10	10
mother	5	0	10	10
son	10	10	0	5
daughter	10	10	5	0

nodes of the graph, and the nonterminal nodes indicate that two (or more) clusters merge. □

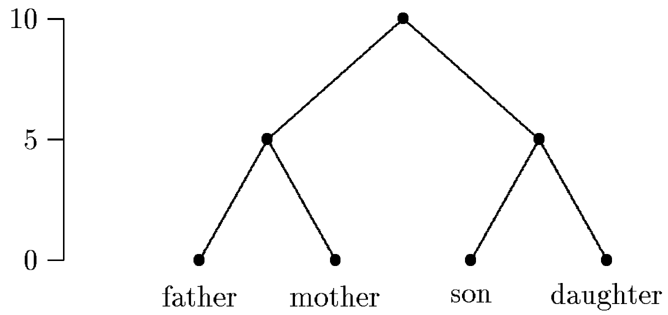


Figure 4.1. Rooted tree representation of the dissimilarity measure δ of Table 4.1.

There are also some critical remarks that apply to Miller's method of representing semantic structures. Although sufficient conditions for the representation are known and testable, they are ignored. The data are manipulated by certain algorithms resulting in various *'approximate solutions'* (see Figure 2 in Miller, 1969, p. 180). It is again unclear whether the obtained theoretical relations refer to any empirical fact. As a second problem, Miller's method does not offer any possibility to consider individual semantic structures. Only if the sorting data from a considerable number of subjects are pooled, the derived dissimilarity measure may become nontrivial. Moreover, deriving a numerical dissimilarity measure always induces a total order (see Definition 4.3.1 below) of pairs of stimuli. It is however by no means assured that for example the critical property of transitivity is satisfied for qualitative judgements on the dissimilarity of pairs of stimuli.

A representation theorem proposed by Colonius & Schulze (1981) solves the above stated problems. The authors provide a foundation of rooted tree representations by formulating qualitative conditions for a ternary relation, which results from partitioning triples of lexical items with respect to similarity of meaning: The relation holds for a triple of elements (a, b, c), if a and b are more similar to each other than a, c, and b, c, respectively. In contrast to Miller's method of sorting, Colonius et al. (1981) consider individual semantic structures. Experimental results however indicate that the constraints, which a rooted tree representation imposes on the ternary relation, are often not empirically valid (Schulze & Colonius, 1979). Thus the representation seems to be too restrictive.

Actually, the data of various experiments show that often qualitative similarity judgments even cannot be represented by a metric (cf. Tversky, 1977). An excellent survey of different representations of proximity data, and their empirical applications, is given by Suppes, Krantz, Luce & Tversky (1989), where further references can be found. For instance, Rosch (1975) showed that the prototype of a conceptual category is generally less similar to a variant than vice versa. In other terms, the similarity judgments are not symmetric. An example from Tversky (1977) may illustrate this. In an experiment nearly all subjects agreed that North Korea is similar to Red China, but only very few considered Red China to be similar to North Korea. The judgments also depend on the experimental task. Asking for dissimilarities instead of similarities does not simply reverse the ordering of pairs.

Tversky (1977) proposed a feature representation of proximity data that is neither metric nor dimensional, and takes into account the above mentioned experimental findings. Each stimulus is represented as a set of features and the proximity of two stimuli is expressed in terms of their shared and distinctive features. This representation of proximity, however, presupposes the features (and their assignment to the stimuli) to be known in advance. Obviously, the applicability of this so-called *contrast model* for assessing semantic structures is limited by this fact. The feature representation of the stimuli and the decision process (which operates on the features and generates the similarity judgments) cannot be identified simultaneously on the basis of proximity data. If the characteristics guiding the decision process are unknown, as it usually will be the case, the proximity data do not carry enough information to obtain a unique feature representation.

Because of this, the subsequently developed representation of semantic structures will not be based on any (numerical or qualitative) measure of similarity of meaning. It will be more closely related to linguistical semantics. Moreover, the feature representation that will be proposed generalizes rooted trees.

4.3 Formalizing semantic structures

This section introduces the mathematical structure of a lattice as a candidate for representing semantic structures. The following subsections characterize a lattice from different points of view. A first characterization arises from the theory of ordered sets, which is motivated by considering the ordinal properties of hyponymy. Moreover, an exposition of the theory of concept lattices (Wille, 1982) will show that a lattice structure is induced by any feature representation. From a second point of view, a lattice is defined as an algebraic structure with two binary operations satisfying certain properties. One of these operations may be interpreted as semantic generalization, which will be the basis of a method for assessing semantic structures.

Although the order-theoretic and the algebraic approach do not seem to have anything in common at first sight, they are essentially equivalent. The subsequent presentation of the theory closely follows an excellent introduction to lattices and order theory by Davey & Priestley (1990), and the standard textbook on lattice theory by Birkhoff (1967).

4.3.1 Partial orders

As we have already seen, the semantic relation of hyponymy induces an ordering on a set of words X. Such an ordering is formalized by a binary relation $R \subseteq X \times X$ on X, i. e. R is a subset of the set of all ordered pairs $X \times X$ of elements of X. A set X together with a binary relation R on X is called a *relational structure*, and is usually denoted by $\langle X, R \rangle$. Now, when do we call a relational structure $\langle X, R \rangle$ an ordering? The answer to this question is not as easy as it might seem, since there are different notions of ordering. The most common example of an ordering is probably the numerical relation *'less than or equal to'*, denoted by $\leq$. The ordinal properties of the real numbers $\mathbb{R}$ with respect to the relation $\leq$ are captured by the following definition.

Definition 4.3.1 A relational structure $\langle X, \preceq \rangle$, with $\preceq$ a binary relation, is a *total order*, if for all $a, b, c \in X$

1. $a \preceq b$ or $b \preceq a$, or both (connected)
2. if $a \preceq b$ and $b \preceq a$, then $a = b$ (antisymmetric)
3. if $a \preceq b$ and $b \preceq c$, then $a \preceq c$ (transitive)

□

The reader may easily verify that the relational structure $\langle \mathbb{R}, \leq \rangle$ is a total order. However, the semantic relation of hyponymy on a set of lexical items does not satisfy all the properties of a total order. Hyponymy may be assumed to be transitive, but in general it is not connected. If we consider the words *mammal* and *pet* for example, then neither is *mammal* hyponymous to *pet*, nor is *pet* hyponymous to *mammal*. As a consequence of this we have to generalize the notion of a total order by weakening the condition of connectedness.

Definition 4.3.2 A relational structure $\langle P, \sqsubseteq \rangle$ with a binary relation $\sqsubseteq$ on P is a *partial order*, if for all $a, b, c \in P$

1. $a \sqsubseteq a$ (reflexive)
2. if $a \sqsubseteq b$ and $b \sqsubseteq a$, then $a = b$ (antisymmetric)
3. if $a \sqsubseteq b$ and $b \sqsubseteq c$, then $a \sqsubseteq c$ (transitivity)

□

Exercise 4.3.1 Show that any total order is a partial order. □

Example 4.3.1 It is left to the reader to verify that each of the following relational structures is a partial order violating the property of connectedness.

1. $\langle \mathrm{N}, | \rangle$, with N the set of natural numbers and $|$ the relation of divisibility. The relation $m \,|\, n$ holds for $m, n \in \mathrm{N}$ if and only if there exists a $k \in \mathrm{N}$ such that $m \cdot k = n$ holds.

2. $\langle \mathrm{N} \times \mathrm{N}, \leq_2 \rangle$, with $\mathrm{N} \times \mathrm{N}$ the set of ordered pairs of natural numbers and $\leq_2$ defined by

$$(k, l) \leq_2 (m, n) \text{ iff } k \leq m \text{ and } l \leq n.$$

3. $\langle 2^X, \subseteq \rangle$, with 2^X the set of all subsets of a non-empty set X and $\subseteq$ denoting set inclusion. □

Notice that if $\langle P, \sqsubseteq \rangle$ is a partial order and for all $a, b \in P$ the relation b ¡ a holds if and only if $a \sqsubseteq b$, then $\langle P,$ ¡ $\rangle$ is a partial order too. The partial order $\langle P,$ ¡ $\rangle$ is called the *dual* of $\langle P, \sqsubseteq \rangle$. The semantic relation of superordination may thus be considered to be the dual of hyponymy and vice versa.

Given any statement about a partial order, we obtain the corresponding *dual statement* by replacing each occurrence of $\sqsubseteq$ by ¿. The duality principle asserts that for a statement about partial orders which is true in all partial orders, the corresponding dual statement is true in all partial orders too.

There exists a very useful graphical representation at least for finite partial orders $\langle P, \sqsubseteq \rangle$, i. e. the set P is finite. To describe the so-called *Hasse diagram* we need a notion of immediate successive elements.

Definition 4.3.3 Let $\langle P, \sqsubseteq \rangle$ be a partial order. For all $a, b \in P$ with $a \neq b$ we say b *covers* a and denote this by $a \lhd b$, when $a \sqsubseteq b$ and there is no element $x \in P$ distinct from a, b for which $a \sqsubseteq x \sqsubseteq b$. □

By Definition 4.3.3, a *covering relation* $\lhd$ is associated to any partial order $\langle P, \sqsubseteq \rangle$. The covering relation $\lhd$ may also be called the *transitive reduction* of $\sqsubseteq$. Moreover, if P is finite, $a \sqsubseteq b$ holds for distinct $a, b \in P$ if and only if there exists a finite sequence of covering relations $a = c_0 \lhd c_1 \lhd \ldots \lhd c_n = b$. This means that, in the finite case, the partial order $\sqsubseteq$ is completely determined by its covering relation $\lhd$. It is this fact which permits a convenient graphical illustration of finite partial orders by Hasse diagrams.

In a Hasse diagram of a (finite) partial order $\langle P, \sqsubseteq \rangle$ the elements of P are represented by points in the plane, and the covering relation is depicted by interconnecting lines. Additionally, the point associated to an element b is placed higher (i. e. nearer to the top of the paper) than that of an element a whenever $a \lhd b$ holds, and no point representing an element distinct from a and b lies on the line segment connecting the points associated to a and b, respectively.

Example 4.3.2 Figure 4.2 presents the Hasse diagrams of the partial orders (a) $\langle \{1, 2, 3, 4, 6\}, | \rangle$ (with $|$ the relation of divisibility), (b) $\langle 2^{\{a,b,c\}}, \subseteq \rangle$, and (c) $\langle \{1, 2, 3\} \times \{1, 2, 3\}, \leq_2 \rangle$. □

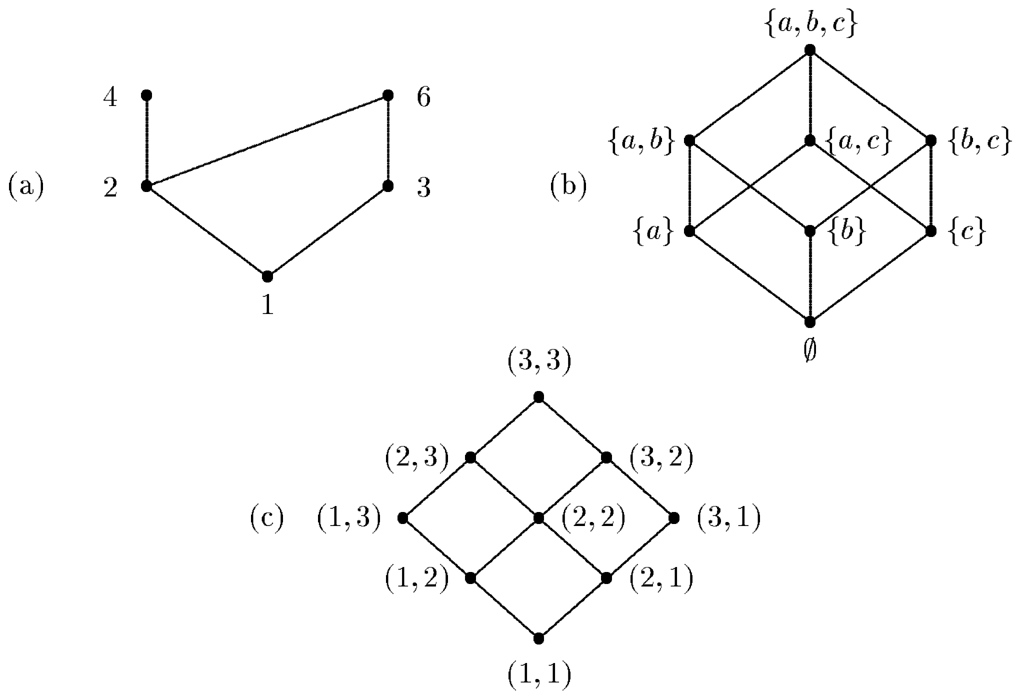


Figure 4.2. Hasse diagrams of the partial orders (a) $\langle \{1,2,3,4,6\}, | \rangle$, (b) $\langle 2^{\{a,b,c\}}, \subseteq \rangle$, and (c) $\langle \{1,2,3\}\times\{1,2,3\}, \leq_2 \rangle$ (see Example 4.3.1).

As far as we have seen, the semantic relation of hyponymy may be represented by a partial order. The occurrence of synonyms however has to be taken into consideration. Two distinct words are called *synonymous*, if they have the same meaning. Synonymy may be defined as symmetrical hyponymy (Lyons, 1968), i. e. two words are synonyms if one is hyponymous to the other, and vice versa. Given two synonyms, and assuming hyponymy to be a partial order, the property of antisymmetry then would imply their identity. Thus hyponymy is only assumed to satisfy the properties of reflexivity and transitivity. A relational structure $\langle P, \preceq \rangle$ with $\preceq$ a reflexive and transitive relation on P is called a *quasi order*. For any quasi order $\langle P, \preceq \rangle$ the binary relation $\sim$ on P defined by

$$a \sim b \text{ iff } (a \preceq b \text{ and } b \preceq a) \tag{4.2}$$

turns out to be an equivalence relation on P.

Exercise 4.3.2 Show that the relation $\sim$ defined by Equation (4.2) is an equivalence relation on P. This means that $\sim$ has to be reflexive, transitive, and symmetric (i. e. $a \sim b$ implies $b \sim a$ for all $a, b \in P$). □

In the empirical application we have in mind, the semantic relation of hyponymy will thus be represented by a quasi order. The further exposition of the theory, however, will be based on the notion of a partial order. The argument is as follows: Any quasi order induces a partial order by forming equivalence classes.

For a given quasi order $\langle P, \preceq \rangle$, the equivalence relation $\sim$ of Equation (4.2) induces a partition of the set P which consists of the equivalence classes

$$[a] = \{b \in P \mid b \sim a\}$$

for all $a \in P$. Now, the quasi order $\preceq$ induces a partial order $\sqsubseteq$ on the set of these equivalence classes by

$$[a] \sqsubseteq [b] \quad \text{iff} \quad a \preceq b.$$

EXERCISE 4.3.3 Show that the relation $\sqsubseteq$ on $\{[a] \mid a \in P\}$ is well-defined, i. e. it is independent of the elements chosen to represent the equivalence classes. □

4.3.2 Lattices

DEFINITION 4.3.4 Let $\langle P, \sqsubseteq \rangle$ be a partial order. An element $\bot \in P$ is called a *least element of* $\langle P, \sqsubseteq \rangle$, if $\bot \sqsubseteq a$ holds for all $a \in P$. An element $\top \in P$ is called a *greatest element of* $\langle P, \sqsubseteq \rangle$, if $a \sqsubseteq \top$ holds for all $a \in P$. □

EXAMPLE 4.3.3 In the partial order $\langle \{1,2,3\} \times \{1,2,3\}, \leq_2 \rangle$, depicted in Figure 4.2 c), the pair (1,1) is a least element and (3,3) is a greatest element. In $\langle \mathrm{N} \times \mathrm{N}, \leq_2 \rangle$ the pair (1,1) again is a least element, but there exists no greatest element. Even in a finite partial order, there may be no least or greatest element. Consider the partial order $\langle \{1,2,3,4,6\}, | \rangle$, with $|$ the relation of divisibility (Figure 4.2 a), in which no greatest element exists. □

The preceding examples show that a partial order may not have least or greatest elements. They are, however, unique provided they exist.

EXERCISE 4.3.4 Let $\langle P, \sqsubseteq \rangle$ be a partial order. Prove that there exists at most one least element $\bot \in P$ and at most one greatest element $\top \in P$. □

If we consider a subset Q of the set P of a partial order $\langle P, \sqsubseteq \rangle$, then the elements of P which are comparable to all elements in Q are of considerable interest.

DEFINITION 4.3.5 Let $\langle P, \sqsubseteq \rangle$ be a partial order and $Q \subseteq P$. The set

$$Q^u = \{u \in P \mid a \sqsubseteq u, \quad \text{for all} \quad a \in Q\}$$

is called the *set of the upper bounds* of Q, and an element $u \in Q^u$ is called an *upper bound* of Q. The set

$$Q^l = \{l \in P \mid l \sqsubseteq a, \quad \text{for all} \quad a \in Q\}$$

is called the *set of the lower bounds* of Q, and an element $l \in Q^l$ is called a *lower bound* of Q. □

It is particularly interesting whether the set of upper bounds Q^u of a subset Q of P has a least element, and whether the set of lower bounds Q^l has a greatest element.

Definition 4.3.6 Let $\langle P, \sqsubseteq \rangle$ be a partial order and $Q \subseteq P$. The sets Q^u and Q^l are the sets of upper and lower bounds of Q respectively. An element $s \in Q^u$ is called the *least upper bound* or *supremum* of Q, denoted by $\sup Q = s$, if $s \sqsubseteq u$ for all $u \in Q^u$. An element $i \in Q^l$ is called the *greatest lower bound* or *infimum* of Q, denoted by $\inf Q = i$, if $l \sqsubseteq i$ for all $l \in Q^l$. □

Notice that a given subset $Q \subseteq P$ may not have a least upper bound or a greatest lower bound, but $\sup Q$ and $\inf Q$ are unique provided they exist (see Exercise 4.3.4).

Definition 4.3.7 A partial order $\langle P, \sqsubseteq \rangle$ is called a *lattice*, if $\sup\{a, b\}$ and $\inf\{a, b\}$ exist for all $a, b \in P$. It is called a *complete lattice*, if $\sup Q$ and $\inf Q$ exist for all subsets $Q \subseteq P$. □

Example 4.3.4 In Figure 4.2, the partial order depicted in a) is not a lattice, whereas the partial orders in b) and c) are lattices. □

Remark 4.3.1 Notice that any finite lattice $\langle L, \sqsubseteq \rangle$, i. e. L is a finite set, is a complete lattice too. In general a lattice $\langle L, \sqsubseteq \rangle$ may not have a least or a greatest element, but if $\langle L, \sqsubseteq \rangle$ is a complete lattice, then a least and a greatest element is given by $\inf L$ and $\sup L$, respectively. □

4.3.3 Concept lattices

The theory of concept lattices (Wille, 1982, 1987; Ganter & Wille, 1989; see also Davey & Priestley, 1990) shows that any characterization of a set of objects by distinctive features corresponds to a lattice. If we assume that a collection of semantic components constitute the meaning of a lexical item, then the theory of concept lattices provides another argument for representing semantic structures by lattices. The theory however may be applied not only to semantic structures, but to any set of arbitrary objects and some features assigned to them.

Definition 4.3.8 A relational structure $\langle G, M, I \rangle$ with G and M sets and $I \subseteq G \times M$ a binary relation, is called a *(formal) context*. The elements $g \in G$ are interpreted as *objects* and the elements $m \in M$ are interpreted as *attributes*[1]. We say the object g has the attribute m, if gIm holds. □

The following example is chosen to demonstrate the universality of the theory of concept lattices.

[1]The letters G and M refer to the initial letters of the German "Gegenstände" for objects, and "Merkmale" for attributes, respectively.

Example 4.3.5 Let $G = \{R_1, R_2, R_3, R_4, R_5\}$ be the set of the binary relations

$$\begin{aligned} R_1 &= \emptyset \\ R_2 &= \{(a,a),(b,b)\} \\ R_3 &= \{(a,b),(b,a)\} \\ R_4 &= \{(a,a),(b,b),(a,b)\} \\ R_5 &= S \times S \end{aligned}$$

on the set $S = \{a, b\}$, and consider the properties of reflexivity, connectedness, transitivity, antisymmetry and symmetry as attributes. Then we get the context in Table 4.2. □

Table 4.2. Formal context of binary relations R_1 to R_5.

	reflexive	connected	transitive	antisymmetric	symmetric
R_1			×	×	×
R_2	×		×	×	×
R_3					×
R_4	×	×	×	×	
R_5	×	×	×		×

The relation $I \subseteq G \times M$ induces relations between the subsets of G and M. Consider the maps $\rho : 2^G \longrightarrow 2^M$ and $\sigma : 2^M \longrightarrow 2^G$, with 2^G and 2^M the powersets of G and M respectively. Let gI denote the set of all attributes which are assigned to the object $g \in G$, thus

$$gI = \{m \in M \mid gIm\}.$$

The set of all objects that have a particular attribute $m \in M$ is denoted by Im:

$$Im = \{g \in G \mid gIm\}.$$

Now, the maps ρ and σ are defined by

$$\rho(A) = \bigcap_{g \in A} gI \tag{4.3}$$

$$\sigma(B) = \bigcap_{m \in B} Im \tag{4.4}$$

for $A \in 2^G$ and $B \in 2^M$. By this definition, $\rho(A)$ is the set of attributes that the objects $g \in A$ have in common. $\sigma(B)$ is the set of objects to which all the attributes $m \in B$ are assigned to.

Consider the subset $A = \{R_1, R_4\}$ of the set of binary relations of Example 4.3.5. From the given context we derive $\rho(A) = \{t, a\}$. Both relations R_1 and R_4 are transitive and antisymmetric, and share none of the other properties. Conversely, for $B = \{t, a\}$ we obtain $\sigma(B) = \{R_1, R_2, R_4\}$. Notice that $\rho(\{R_1, R_2, R_4\}) = \{t, a\} = B$ holds.

The subsets $A \subseteq G$ and $B \subseteq M$, which are related to each other by the equations

$$\begin{aligned} \rho(A) &= B \\ \sigma(B) &= A \end{aligned}$$

are of particular interest.

Definition 4.3.9 Let $\langle G, M, I \rangle$ be a context. An ordered pair (A, B) with $A \subseteq G$, $B \subseteq M$ and $\rho(A) = B$, $\sigma(B) = A$ is called a *(formal) concept* of the given context. A is said to be the *extent* of the concept and B is said to be the *intent* of the concept. □

The extent A of a concept (A, B) is the set of objects, which share the attributes in B. The intent B of (A, B) is the set of attributes assigned to all objects in A. As we have already seen, the pair $(\{R_1, R_2, R_4\}, \{t, a\})$ is a concept of the context of Example 4.3.5. This is also true for the pairs $(\{R_2, R_4, R_5\}, \{r, t\})$ and $(\{R_4\}, \{r, c, t, a\})$, for instance. The pair $(\{R_1, R_4\}, \{t, a\})$ is not a concept of the context of Example 4.3.5, since $\sigma(\{t, a\}) = \{R_1, R_2, R_4\}$ is a proper superset of $\{R_1, R_4\}$.

Formally, defining ρ and σ in the above stated way results in a Galois connection between the powersets 2^G and 2^M with respect to set inclusion. This means that the properties of the following Definition 4.3.10 are satisfied.

Definition 4.3.10 Let $\langle P_1, \sqsubseteq_1 \rangle$ and $\langle P_2, \sqsubseteq_2 \rangle$ be partial orders, and let $\rho : P_1 \longrightarrow P_2$ and $\sigma : P_2 \longrightarrow P_1$ be any maps, such that for all $x, y \in P_1$ and $u, v \in P_2$

1. $x \sqsubseteq_1 y$ implies $\rho(y) \sqsubseteq_2 \rho(x)$,
2. $u \sqsubseteq_2 v$ implies $\sigma(v) \sqsubseteq_1 \sigma(u)$,
3. $x \sqsubseteq_1 \sigma \circ \rho(x)$ and $u \sqsubseteq_2 \rho \circ \sigma(u)$.

The maps ρ and σ are said to define a *Galois connection* between the partial orders $\langle P_1, \sqsubseteq_1 \rangle$ and $\langle P_2, \sqsubseteq_2 \rangle$. □

A Galois connection is a kind of order reversing relationship between two partial orders. In case of $\langle 2^G, \subseteq \rangle$ and $\langle 2^M, \subseteq \rangle$ being the respective partial orders, the compositions $\sigma \circ \rho$ and $\rho \circ \sigma$ of the maps ρ and σ can be interpreted in the following way. The set $\sigma \circ \rho(A)$ is the set of all objects in G that have in common those attributes shared by the objects $g \in A$. Conversely, $\rho \circ \sigma(B)$ is the set of attributes in M that those objects have in common, to which all the attributes $m \in B$ are assigned to.

Exercise 4.3.5 Show that the maps $\rho : 2^G \longrightarrow 2^M$ and $\sigma : 2^M \longrightarrow 2^G$ as defined by Equations (4.3) and (4.4) form a Galois connection between the powersets 2^G and 2^M with respect to set inclusion. □

If we take a closer look at the maps $\sigma \circ \rho : 2^G \longrightarrow 2^G$ and $\rho \circ \sigma : 2^M \longrightarrow 2^M$, we see that for any concept (A, B) of the context $\langle G, M, I \rangle$ the equations

$\sigma \circ \rho(A) = A$, $\rho \circ \sigma(B) = B$ follow from Definition 4.3.9. These equalities characterize the extents A and the intents B as special subsets of G and M, respectively. Definition 4.3.10 only implies $\sigma \circ \rho(X) \supseteq X$ and $\rho \circ \sigma(Y) \supseteq Y$ for arbitrary subsets $X \in 2^G$, $Y \in 2^M$. The properties of the compositions $\sigma \circ \rho$ and $\rho \circ \sigma$ are captured by the following definition of a closure operator. Definition 4.3.9 characterizes the collection of the extents, and the collection of the intents of a given context $\langle G, M, I \rangle$ as the collections of the closed subsets of G and M, respectively.

DEFINITION 4.3.11 A *closure operator* on a set S is a map $\phi : 2^S \longrightarrow 2^S$ such that for any $X, Y \in 2^S$

1. $X \subseteq \phi(X)$ (extensive)
2. $\phi(X) = (\phi \circ \phi)(X)$ (idempotent)
3. if $X \subseteq Y$, then $\phi(X) \subseteq \phi(Y)$ (isotone)

A subset X of S is said to be *closed* under ϕ (or a *fixed point* of ϕ), whenever $\phi(X) = X$ holds. □

Now, if $\mathcal{B}_{GMI}$[2] denotes the set of all concepts of a context $\langle G, M, I \rangle$, then an ordering on $\mathcal{B}_{GMI}$ arises in a natural way.

DEFINITION 4.3.12 Suppose $(A_1, B_1), (A_2, B_2) \in \mathcal{B}_{GMI}$. A binary relation $\sqsubseteq$ on $\mathcal{B}_{GMI}$ is defined by

$$(A_1, B_1) \sqsubseteq (A_2, B_2) \text{ iff } A_1 \subseteq A_2.$$ □

Notice that it suffices to consider the extents of the concepts (A_1, B_1) and (A_2, B_2), since $A_1 \subseteq A_2$ is equivalent to $B_2 \subseteq B_1$ as a result of Definitions 4.3.9 and 4.3.10. The obtained ordering relation on the set of concepts is interpreted as the semantic relation of hyponymy.

DEFINITION 4.3.13 If $(A_1, B_1) \sqsubseteq (A_2, B_2)$, then (A_1, B_1) is said to be *subordinated* to (A_2, B_2), and conversely, (A_2, B_2) is said to be *superordinated* to (A_1, B_1). □

It is evident that $\langle \mathcal{B}_{GMI}, \sqsubseteq \rangle$ is a partial order, since the relation $\sqsubseteq$ is defined on set inclusion. For $\langle \mathcal{B}_{GMI}, \sqsubseteq \rangle$ being a complete lattice, the existence of least upper bounds and greatest lower bounds for all subsets of $\mathcal{B}_{GMI}$ has to be shown. This is done by defining:

$$\text{(4.5)} \quad \inf\{(A_j, B_j) \mid (A_j, B_j) \in \mathcal{B}_{GMI}, j \in J\} = (\bigcap_{j \in J} A_j, \rho \circ \sigma(\bigcup_{j \in J} B_j)),$$

$$\text{(4.6)} \quad \sup\{(A_j, B_j) \mid (A_j, B_j) \in \mathcal{B}_{GMI}, j \in J\} = (\sigma \circ \rho(\bigcup_{j \in J} A_j), \bigcap_{j \in J} B_j).$$

According to these definitions the extent of the infimum of a subset of $\mathcal{B}_{GMI}$ is simply the intersection of the extents of the respective concepts. The intent

[2] The letter $\mathcal{B}$ refers to the German "Begriff" for concept.

of the supremum of a set of concepts results from intersecting the respective intents. This means that both the collections of all extents of $\mathcal{B}_{GMI}$, and all intents of $\mathcal{B}_{GMI}$, respectively, are $\cap$-stable: The intersection of extents (intents) of $\mathcal{B}_{GMI}$ is again an extent (intent) of $\mathcal{B}_{GMI}$. This is not true for the set-theoretic union of extents and intents of $\mathcal{B}_{GMI}$. The intent of the infimum of a set of concepts for example is the least intent of $\mathcal{B}_{GMI}$ that contains the union of the respective intents.

REMARK 4.3.2 Notice that $\langle \mathcal{B}_{GMI}, \sqsubseteq \rangle$ being a complete lattice does not put constraints on the relation I of the context $\langle G, M, I \rangle$. □

The complete lattice $\langle \mathcal{B}_{GMI}, \sqsubseteq \rangle$ can be illustrated by a Hasse diagram. Assigning the names of objects and attributes to the concepts in the following way assures that the context $\langle G, M, I \rangle$ can be reconstructed from the corresponding concept lattice: The name of an object $g \in G$ is assigned to the concept $(\sigma \circ \rho(\{g\}), \rho(\{g\}))$, which is the concept with least extent including g. The name of a attribute $m \in M$ is assigned to the concept $(\sigma(\{m\}), \rho \circ \sigma(\{m\}))$, which is the concept with least intent including m. The extent of a concept is the set of objects assigned to that particular concept or to a concept which is hyponymous to the given one. The intent of a concept is the set of all attributes assigned to that particular concept or to a concept, which is superordinated to the given one. For reconstructing the context from the corresponding concept lattice, define gIm if and only if g and m are assigned to the same concept, or the concept m is assigned to is superordinated to the concept g is assigned to.

EXAMPLE 4.3.6 The diagram in Figure 4.3 illustrates the concept lattice derived from the context of Table 4.2. □

The theory of concept lattices as introduced above, strongly supports a lattice-theoretic representation of semantic structures. The basic data are assignments of features to objects, called formal contexts. Such an assignment is however not the starting point of psycholinguistic research, but its ultimate goal. From the representation of semantic structures, which will be proposed in Section 4.4, a formal context can easily be derived.

4.3.4 Lattice algebra

In Section 4.3.2 lattices were introduced as ordered sets of a special type. Now we consider them as algebraic structures in their own right. An algebraic structure $\langle L, \circ_1, \ldots, \circ_m \rangle$ is a nonempty set L together with some n_i-ary operations $\circ_i$ defined on L, i. e. $\circ_i : L^n \longrightarrow L$ $(i = 1, \ldots, m)$. We postpone a discussion of how the assessment of semantic structures may benefit from the algebraic viewpoint until the relationship between order theoretic and algebraic characterization is established.

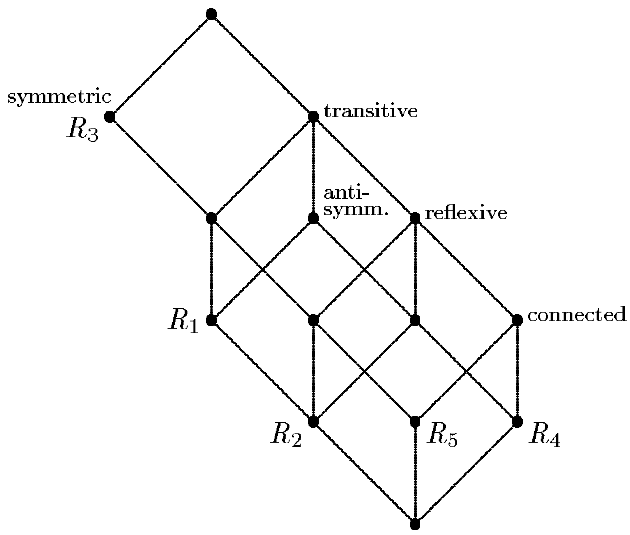


Figure 4.3. Hasse diagram of the concept lattice derived from the context of Table 4.2.

EXAMPLE 4.3.7 Obviously, an algebraic characterization of the lattice $\langle 2^S, \subseteq \rangle$ may be based on the familiar set-theoretic operations of union $\cup$ and intersection $\cap$. In particular, the reader will verify without much effort that for any $A, B \in 2^S$:

$$\begin{aligned} A \cup B &= \sup\{A, B\} \\ A \cap B &= \inf\{A, B\} \end{aligned}$$

□

For generalizing this relationship to an arbitrary lattice $\langle L, \sqsubseteq \rangle$, we define two operations $\sqcup$ and $\sqcap$ on L called *join* and *meet*, respectively. Proposition 4.3.1 deals with the details and the consequences of such a definition.

PROPOSITION 4.3.1 *Let $\langle L, \sqsubseteq \rangle$ be a lattice. The binary operations $\sqcup$ and $\sqcap$ on L defined by*

$$\begin{aligned} a \sqcup b &= \sup\{a, b\} \\ a \sqcap b &= \inf\{a, b\} \end{aligned}$$

for all $a, b \in L$, satisfy the following properties:

1. $a \sqcup (b \sqcup c) = (a \sqcup b) \sqcup c$ *and* $a \sqcap (b \sqcap c) = (a \sqcap b) \sqcap c$ *(associativity)*
2. $a \sqcup b = b \sqcup a$ *and* $a \sqcap b = b \sqcap a$ *(commutativity)*
3. $a \sqcap (a \sqcup b) = a$ *and* $a \sqcup (a \sqcap b) = a$ *(absorption)*
4. $a \sqcup a = a$ *and* $a \sqcap a = a$ *(idempotency)*

□

PROOF. The duality principle we have encountered in Section 4.3.1 implies that interchanging $\sqcup$ and $\sqcap$ in a statement about lattices results in the corresponding dual statement. It is therefore enough to consider only one of the two equations of each property.

Since commutativity, idempotency and absorption are immediate, we only prove associativity. It is enough to show that $(a \sqcup b) \sqcup c = \sup\{a, b, c\}$ because of commutativity. But this follows from

$$\begin{array}{rcl} d \in \{a,b,c\}^u & \Leftrightarrow & d \in \{a,b\}^u \text{ and } c \sqsubseteq d \\ & \Leftrightarrow & a \sqcup b \sqsubseteq d \text{ and } c \sqsubseteq d \\ v & \Leftrightarrow & d \in \{a \sqcup b, c\}^u. \end{array}$$

■

PROPOSITION 4.3.2 *Let the algebraic structure* $\langle L, \sqcup, \sqcap \rangle$ *satisfy the properties 1–3 of Proposition 4.3.1. If a binary relation* $\sqsubseteq$ *on* L *is defined by*

$$a \sqsubseteq b \quad \textit{iff} \quad a \sqcup b = b \quad \textit{iff} \quad a \sqcap b = a$$

for all $a, b \in L$*, then* $\langle L, \sqsubseteq \rangle$ *is a lattice with* $\sup\{a, b\} = a \sqcup b$ *and* $\inf\{a, b\} = a \sqcap b$. □

PROOF. It is left to the reader to prove the equivalence $a \sqcup b = b$ iff $a \sqcap b = a$ for all $a, b \in L$ (Exercise 4.3.6), and to show that absorption implies idempotency (Exercise 4.3.7).

The reflexivity of $\sqsubseteq$ follows directly from idempotency. The relation $\sqsubseteq$ is antisymmetric because $a \sqsubseteq b$ and $b \sqsubseteq a$ imply $b = a \sqcup b = b \sqcup a = a$. Since $a \sqsubseteq b$ are $b \sqsubseteq c$ are equivalent to $a \sqcup b = b$ and $b \sqcup c = c$, respectively, we have

$$\begin{array}{rcl} a \sqcup c & = & a \sqcup (b \sqcup c) \\ & = & (a \sqcup b) \sqcup c \\ & = & b \sqcup c \\ & = & c, \end{array}$$

which again is equivalent to $a \sqsubseteq c$ and therefore $\sqsubseteq$ is transitive.

It remains to show that the least upper bound of elements $a, b \in L$ equals $a \sqcup b$. The equation $\inf\{a, b\} = a \sqcap b$ then follows by duality. By absorption we get $a \sqcap (a \sqcup b) = a$ and (using commutativity) $b \sqcap (a \sqcup b) = b$, which implies that $a \sqcup b \in \{a, b\}^u$. To show that $a \sqcup b$ is the least upper bound of $\{a, b\}$ let $s \in \{a, b\}^u$. From $a \sqcup s = s$ and $b \sqcup s = s$ we obtain

$$\begin{array}{rcl} (a \sqcup b) \sqcup s & = & a \sqcup (b \sqcup s) \\ & = & a \sqcup s \\ & = & s, \end{array}$$

which is equivalent to $a \sqcup b \sqsubseteq s$. ■

EXERCISE 4.3.6 Prove the equivalence $a \sqcup b = b$ iff $a \sqcap b = a$ for all $a, b \in L$. □

EXERCISE 4.3.7 Show that absorption implies idempotency. □

The preceding two propositions show that an algebraic structure $\langle L, \sqcup, \sqcap \rangle$ satisfying

1. $a \sqcup (b \sqcup c) = (a \sqcup b) \sqcup c$ and $a \sqcap (b \sqcap c) = (a \sqcap b) \sqcap c$ (associativity)

2. $a \sqcup b = b \sqcup a$ and $a \sqcap b = b \sqcap a$ (commutativity)
3. $a \sqcap (a \sqcup b) = a$ and $a \sqcup (a \sqcap b) = a$ (absorption)

for all $a, b, c \in L$ defines, and is defined by, a lattice $\langle L, \sqsubseteq \rangle$. Because of this, both structures are considered to be equivalent, and an algebraic structure $\langle L, \sqcup, \sqcap \rangle$ satisfying associativity, commutativity and absorption will therefore be called a *lattice*.

Remark 4.3.3 As indicated by Example 4.3.7 on page 127, the relational structure $\langle 2^S, \cup, \cap \rangle$ provides a first example of an algebraically characterized lattice. Choosing the symbols $\sqcup$ and $\sqcap$ for join and meet, respectively, stresses this fact. However, the concept of a lattice is still more general than the given example might suggest. For one reason, notice that the set L is not restricted to be a family of sets, but can be any set of arbitrary objects. A second reason is that in $\langle 2^S, \cup, \cap \rangle$ for all sets $A, B, C \in 2^S$ the relation between the two operations is characterized by distributivity:

$$\begin{aligned} A \cup (B \cap C) &= (A \cup B) \cap (A \cup C) \\ A \cap (B \cup C) &= (A \cap B) \cup (A \cap C) \end{aligned}$$

Distributivity is not implied by associativity, commutativity, and absorption. Figure 4.4 presents an example of a non-distributive lattice. □

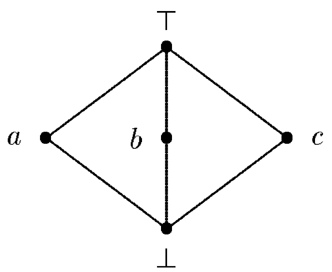


Figure 4.4. Non-distributive lattice on the set $\{\bot, a, b, c, \top\}$.

We now turn to the initially posed question concerning the intended application. The preceding sections provide strong arguments for a lattice representation of the semantic relation of hyponymy on a set of concepts L. For two concepts $a, b \in L$ the definitions of Proposition 4.3.1 then imply that $a \sqcup b$ is the least concept superordinated to a and b, and $a \sqcap b$ is the greatest concept hyponymous to a and b. For assessing a semantic structure the join-operation seems to be of particular interest. This becomes evident if we consider the relationship between semantic components or features on the one hand, and superordinated concepts on the other hand. The theory of concept lattices introduced in Section 4.3.3 formally describes this relationship. It follows from Equation (4.6) on page 125 that the features associated with the join $a \sqcup b$ of two concepts $a, b \in L$ are exactly the features shared by a and b. Any superordinated concept to a given concept can be interpreted as one of its features. Now, the assessment of a semantic structure may be based on semantic generalization: Given two concepts, a subject has to respond with a least superordinated concept. Such a situation is well-known in mental testing. In many intelligence tests (e. g. Wechsler, 1955) there is a subtest, in which a subject is presented two nouns, and is asked to give a verbal description of the features they share. For instance, if the pair *orange* – *banana* is presented a subject might answer *'you can eat them both'*, or *'fruits'*, or *'tropical fruits'*. Because we want to represent this kind of data by a closed algebraic operation, this task needs to be specified. The subjects are asked to respond with a noun (eventually modified by adjectives) that denotes a least superordinated concept. In the above given example, a subject may now answer with the noun phrase *tropical fruits*, since it is hyponymous to *fruits*, and it is superordinated to both *orange* and *banana*. Starting with a set of nouns out of a certain domain, this set successively grows with each noun phrase distinct from the previously given answers. If all pairs of the resulting set are presented to the subject, and there do not occur new noun phrases, then the data may be formalized by a binary algebraic operation. Since we want to represent this empirical operation by the lattice-theoretic join, the following Definition 4.3.14 summarizes the properties of the substructure $\langle L, \sqcup \rangle$ of a lattice $\langle L, \sqcup, \sqcap \rangle$.

DEFINITION 4.3.14 The algebraic structure $\langle L, \sqcup \rangle$ with $\sqcup\colon L \times L \longrightarrow L$ is a *join semilattice*, if for all $a, b, c \in L$:

1. $a \sqcup b = b \sqcup a$ (commutative)
2. $a \sqcup (b \sqcup c) = (a \sqcup b) \sqcup c$ (associative)
3. $a \sqcup a = a$ (idempotent)

□

The definition of a meet semilattice $\langle L, \sqcap \rangle$ is dual to Definition 4.3.14.

4.3.5 Homomorphisms and congruences

The notion of a structure preserving map — a so-called *homomorphism* — is a crucial notion in measurement theory. A homomorphism establishes the relationship between empirical observations and theoretical statements. The following Definition 4.3.15 of a *lattice homomorphism* is based on the algebraic characterization of lattices introduced in the preceding section. Exercise 4.3.16 deals with homomorphisms of partial orders.

DEFINITION 4.3.15 Let $\langle L_1, \sqcup_1, \sqcap_1 \rangle$ and $\langle L_2, \sqcup_2, \sqcap_2 \rangle$ be lattices. A map $\varphi\colon L_1 \longrightarrow L_2$ from L_1 to L_2 is called a *lattice homomorphism* if and only if for all $a, b \in L_1$

$$\begin{aligned} \varphi(a \sqcup_1 b) &= \varphi(a) \sqcup_2 \varphi(b) \quad \text{(join-preserving)} \\ &\text{and} \\ \varphi(a \sqcap_1 b) &= \varphi(a) \sqcap_2 \varphi(b) \quad \text{(meet-preserving)} \end{aligned}$$

If φ is one-one and onto, then φ is called a *lattice isomorphism*. □

Obviously, a join-semilattice homomorphism (meet-semilattice homomorphism) requires only joins (meets) to be preserved.

To prepare the handling of synonymous concepts we consider the properties of an equivalence relation induced by a given lattice homomorphism.

PROPOSITION 4.3.3 *Let* $\langle L_1, \sqcup_1, \sqcap_1 \rangle$ *and* $\langle L_2, \sqcup_2, \sqcap_2 \rangle$ *be lattices, and let* $\varphi\colon L_1 \longrightarrow L_2$ *be a lattice homomorphism from* L_1 *into* L_2*. Then the equivalence relation* $\sim$ *defined on* L_1 *by*

$$a \sim b \quad \textit{iff} \quad \varphi(a) = \varphi(b)$$

for all $a, b \in L_1$ *is compatible with join* $\sqcup_1$ *and meet* $\sqcap_1$ *on* L_1*, such that*

$$a \sim b \qquad \textit{and} \qquad c \sim d$$
$$\textit{implies}$$
$$a \sqcup_1 c \sim b \sqcup_1 d \qquad \textit{and} \qquad a \sqcap_1 c \sim b \sqcap_1 d$$

for all $a, b, c, d \in L_1$. □

PROOF. It is immediate that $\sim$ is an equivalence relation on L_1. If we assume $a \sim b$ and $c \sim d$ which is equivalent to $\varphi(a) = \varphi(b)$ and $\varphi(c) = \varphi(d)$, then we obtain

$$\varphi(a \sqcup_1 c) = \varphi(a) \sqcup_2 \varphi(c) = \varphi(b) \sqcup_2 \varphi(d) = \varphi(b \sqcup_1 d),$$

since φ preserves join. By definition, this equation is equivalent to $a \sqcup_1 c \sim b \sqcup_1 d$. Dually, the equivalence relation $\sim$ is compatible with meet. ■

This result gives rise to the following definition.

DEFINITION 4.3.16 An equivalence relation $\sim$ on a lattice $\langle L, \sqcup, \sqcap \rangle$ which is compatible with both join and meet (according to Proposition 4.3.3) is called a *congruence* on the lattice $\langle L, \sqcup, \sqcap \rangle$. □

In Proposition 4.3.3 congruences are derived from lattice homomorphisms, but Definition 4.3.16 defines them without referring to homomorphisms. Moreover, we shall show that to any congruence on a lattice there is associated a lattice homomorphism.

Let $\langle L, \sqcup, \sqcap \rangle$ be a lattice. Given an equivalence relation $\sim$ on L, we consider the set

$$L/{\sim} = \{[a] \mid a \in L\}$$

of equivalence classes $[a] = \{b \in P \mid b \sim a\}$. The natural definitions[3]

$$\begin{aligned} [a] \sqcup [b] &:= [a \sqcup b] \\ [a] \sqcap [b] &:= [a \sqcap b] \end{aligned}$$

[3]We use the same symbols for join and meet on L and $L/{\sim}$, respectively. It will always be clear from the context to which of the two sets we refer.

for all $a, b \in L$ provide well-defined operations $\sqcup$ and $\sqcap$ on $L/{\sim}$ if they are independent of the elements chosen to represent the equivalence classes. This requires that

$$[a_1] = [a_2] \quad \text{and} \quad [b_1] = [b_2]$$
$$\text{implies}$$
$$[a_1 \sqcup b_1] = [a_2 \sqcup b_2] \quad \text{and} \quad [a_1 \sqcap b_1] = [a_2 \sqcap b_2]$$

for all $a_1, a_2, b_1, b_2 \in L$, which is precisely the compatibility condition of Proposition 4.3.3. It follows that $\sqcup$ and $\sqcap$ are well defined on $L/{\sim}$ if and only if $\sim$ is a congruence.

DEFINITION 4.3.17 If $\sim$ is a congruence on a lattice $\langle L, \sqcup, \sqcap \rangle$, we call $\langle L/{\sim}, \sqcup, \sqcap \rangle$ the *quotient lattice of* $\langle L, \sqcup, \sqcap \rangle$ *modulo* $\sim$. □

The following Proposition 4.3.4 now provides the announced result.

PROPOSITION 4.3.4 *Let* $\sim$ *be a congruence on the lattice* $\langle L, \sqcup, \sqcap \rangle$. *Then* $\langle L/{\sim}, \sqcup, \sqcap \rangle$ *is a lattice and the natural quotient map* $\pi : L \longrightarrow L/{\sim}$, *defined by* $\pi(a) := [a]$, *is a lattice homomorphism.* □

EXERCISE 4.3.8 Prove Proposition 4.3.4. □

4.3.6 Supplementary Problems

EXERCISE 4.3.9 Let $\langle S, d \rangle$ be a metric space. For each $\alpha \geq 0$, define a relation $\approx_\alpha$ on S by

$$x \approx_\alpha y \;\text{ iff }\; d(x, y) \leq \alpha.$$

Show that $\approx_\alpha$ is an equivalence relation for each $\alpha \geq 0$ if and only if d satisfies the ultrametric inequality. □

EXERCISE 4.3.10 Consider the set $A = \{a, b, c, d\}$ and the relation $R = \{(a, a), (b, b), (c, c), (d, d), (a, c), (a, d), (b, c), (b, d), (c, d)\}$ on A.

1. Show that $\langle A, R \rangle$ is a partial order.
2. Determine the covering relation $\lhd$ (see Definition 4.3.3) on A.
3. Draw the Hasse diagram of $\langle A, R \rangle$.
4. Do there exist a least element and a greatest elements in $\langle A, R \rangle$?
5. Determine the least upper bounds and the greatest lower bounds for all subsets of A for which they exist. □

EXERCISE 4.3.11 Draw the Hasse diagram of the partial order $\langle \{2, 3, 5, 7, 6, 35, 210\}, | \rangle$ with $|$ the relation of divisibility, and compare it to the rooted tree in Figure 4.1. □

EXERCISE 4.3.12 Draw the Hasse diagrams of the partial orders $\langle \mathcal{P}(\{1, 2, 3, 4\}), \cup, \cap \rangle$ and $\langle \{1, 2, 3, 4, 6, 9, 12, 18, 36\}, | \rangle$. Compare the last diagram to that of the partial order $\langle \{1, 2, 3\} \times \{1, 2, 3\}, \leq_2 \rangle$ (see Figure 4.2 c). □

Exercise 4.3.13 Show that in any lattice $\langle L, \sqcup, \sqcap \rangle$ $x \sqcup y = z$ implies $x \sqcap z = x$, and $x \sqcap y = z$ implies $x \sqcup z = x$ for all $x, y, z \in L$. □

Exercise 4.3.14 Prove that in any lattice $\langle L, \sqcup, \sqcap \rangle$ the following implications hold for all $x, y \in L$: $x \sqcup y = x$ implies $x \sqcap y = y$ and $x \sqcap y = x$ implies $x \sqcup y = y$. □

Exercise 4.3.15 Show that in any lattice $\langle L, \sqcup, \sqcap \rangle$, the equality $x \sqcup y = x \sqcap y$ implies $x = y$ for all $x, y \in L$. □

Exercise 4.3.16 Let $\langle P_1, \sqsubseteq_1 \rangle$ and $\langle P_2, \sqsubseteq_2 \rangle$ be partial orders. A map $\psi \colon P_1 \longrightarrow P_2$ from P_1 to P_2 is called a *homomorphism of partial orders* if and only if for all $a, b \in P_1$

$$\text{if} \quad a \sqsubseteq_1 b, \quad \text{then} \quad \psi(a) \sqsubseteq_2 \psi(b)$$

If ψ is onto and

$$a \sqsubseteq_1 b \quad \text{iff} \quad \psi(a) \sqsubseteq_2 \psi(b)$$

holds for all $a, b \in P_1$, then ψ is called an *isomorphism of partial orders.*

A homomorphism of partial orders, which is one-one and onto, is not necessarily an isomorphism of partial orders. Show this by providing an example. □

4.4 A method for assessing semantic structures

4.4.1 Theory

In this section, we present a representation theorem that applies to the data resulting from a method for assessing semantic structures for a set of nouns B_0, which we introduced in Section 4.3.4. The details of the experimental procedure will be given in a moment. The data resulting from such a procedure may be formalized by an empirical structure $\langle B, \circ \rangle$, where $B \supseteq B_0$ is the set of noun phrases a subject responds with, and $\circ$ denotes his or her assignments of least superordinates to pairs.

As already mentioned, we cannot expect $\langle B, \circ \rangle$ to be isomorphic to a join semilattice. There may occur synonyms, i. e. distinct nouns with the same meaning associated to them. Within the empirical structure $\langle B, \circ \rangle$ it is difficult to distinguish an occurrence of synonyms from a violation of the axioms of a join semilattice (Definition 4.3.14). Consider the case $a \circ b = c$ and $b \circ a = d$ for $a, b, c, d \in B$. If c and d are not synonymous, then commutativity (see Definition 4.3.14) is violated. If c and d are synonyms, then $a \circ b$ will not be equal to $b \circ a$, but will be equivalent to $b \circ a$ with respect to synonymy. The equality sign in commutativity $a \circ b = b \circ a$ has to be substituted by an equivalence relation $\sim$ on B. Weaker versions of the remaining properties of a join semilattice are obtained in the same way. But how do we arrive at an equivalence relation $\sim$ on B representing synonymy? According to Lyons (1968), synonymy can be defined as symmetrical superordination (see Section 4.3.1). This suggests a complete pair comparison experiment resulting in an empirical relation $\succeq$ of superordination on B. The following definition introduces an equivalence relation for a more general situation.

DEFINITION 4.4.1 Let S be a set and let $R \subseteq S \times S$ be a binary relation on S. A binary relation $\sim$ on S is defined in the following way. For $x, y \in S$ there is $x \sim y$ if and only if

$$\begin{aligned} xRw \quad &\text{iff} \quad yRw, \\ wRx \quad &\text{iff} \quad wRy \end{aligned}$$

for all $w \in S$. □

EXERCISE 4.4.1 Show that the relation $\sim$ of Definition 4.4.1 is an equivalence relation on the set S. □

REMARK 4.4.1 Notice that the relation R in Exercise 4.4.1 is completely arbitrary. Defining synonymy according to Definition 4.4.1 always results in an equivalence relation. If synonymy is based on Equation (4.2) (see page 120), then its properties depend on the relation R. The following exercise deals with the details. □

EXERCISE 4.4.2 Let R be a binary relation on the set S. Let the relation $\sim$ on S be defined according to Definition 4.4.1, and let the relation $\approx$ on S be defined by

$$x \approx y \;\text{ iff }\; (xRy \;\text{ and }\; yRx)$$

for all $x, y \in S$ (see Equation (4.2)). Show that the relations $\sim$ and $\approx$ on S coincide, if R is a quasi-order, i. e. reflexive and transitive. □

DEFINITION 4.4.2 Let the relation $\sim$ be defined on $\langle B, \succeq \rangle$ according to Definition 4.4.1. Then two elements $a, b \in B$ are called *contextual synonyms*, whenever $a \sim b$ holds. □

The term *contextual synonymy* stresses the fact that this definition of synonymy explicitly refers to the context of the experimental task, the set B. This parallels the position of theoretical linguistics that synonymy always depends on the context (Lyons, 1968). Since the relation of contextual synonymy $\sim$ is defined on $\langle B, \succeq \rangle$, its compatibility to the operation $\circ$ has to be shown. Congruence is the compatibility condition we refer to. The subsequently stated representation theorem directly follows from the results of Section 4.3.5.

THEOREM 4.4.1 *If $\langle B, \circ, \sim \rangle$ with $\circ$ a binary operation and $\sim$ an equivalence relation on B satisfies the following conditions for all $a, b, c, d \in B$*

1. $a \circ (b \circ c) \sim (a \circ b) \circ c$ *(weak associativity)*

2. $a \circ b \sim b \circ a$ *(weak commutativity)*

3. $a \circ a \sim a$ *(weak idempotency)*

4. if $a \sim b$ *and* $c \sim d$*, then* $a \circ c \sim b \circ d$ *(congruence)*

then there exists a homomorphism $\varphi : B \longrightarrow L$ *onto a join semilattice* $\langle\, L, \sqcup \,\rangle$*, such that*

$$\begin{array}{ccc} \varphi(a \circ b) & = & \varphi(a) \sqcup \varphi(b) \\ & \textit{and} & \\ a \sim b & \textit{iff} & \varphi(a) = \varphi(b) \end{array}$$

for all $a, b \in B$*.* □

With the theoretical preparations of Section 4.3.5 at hand, the proof of Theorem 4.4.1 poses no problems.

Proof. Defining $L := B/\sim$, and $[a] \sqcup [b] := [a \circ b]$, the required homomorphism is given by the canonical quotient map $\varphi(a) = [a]$ (see Proposition 4.3.4). ■

Remark 4.4.2 As a consequence of Proposition 4.3.3, the conditions of Theorem 4.4.1 are not only sufficient, but also necessary, i. e. they follow from the representation. □

Theorem 4.4.1 lists the empirically testable conditions of a join semilattice representation of the relational structure $\langle\, B, \circ, \sim \,\rangle$. Because of the equivalence of the algebraic and the order-theoretic characterization (see Section 4.3.4), we may also get a join semilattice representation of the relational structure $\langle\, B, \succeq \,\rangle$. We do not explicitly state a representation theorem for this situation, because the exposition of the mathematical background exactly tells us how to obtain such a representation. Consider the relation $\sqsupseteq$ on the set of equivalence classes $\mathcal{B} := B/\sim$, which is induced by $\succeq$. For $\langle\, \mathcal{B}, \sqsupseteq \,\rangle$ the properties of a partial order (see Definition 4.3.2) have to be satisfied. Moreover, Proposition 4.3.1 implies that least upper bounds have to exist for all pairs of equivalence classes of $\mathcal{B}$. If $\langle\, \mathcal{B}, \sqsupseteq \,\rangle$ meets all these requirements, then the natural quotient map $\varphi : B \longrightarrow \mathcal{B}$, $\varphi(a) = [a]$ is again the required homomorphism.

We now turn to the problem of extracting the semantic features of the elements of the considered domain B_0 from the join semilattice representation. The following assumes that at least one of the empirical relational structures $\langle\, B, \circ, \sim \,\rangle$, or $\langle\, B, \succeq \,\rangle$, can be represented by a join semilattice $\langle\, \mathcal{B}, \sqsupseteq \,\rangle$[4]. As already mentioned, the concepts superordinated to that associated to a particular element of B_0 may be interpreted as its semantic features. We thus consider the set $\overline{\mathcal{B}} = \{[b] \in \mathcal{B} \mid [b] \sqsupseteq [a], \text{ for some } a \in B_0\}$, and derive a formal context $\langle\, B_0, \overline{\mathcal{B}}, I \,\rangle$ by defining

$$aI[b] \text{ iff } [b] \sqsupseteq [a]$$

for all $a \in B_0$, $[b] \in \overline{\mathcal{B}}$. In the terminology of formal concept analysis, B_0 is the set of objects, and $\overline{\mathcal{B}}$ is the set of attributes. The relation $I \subseteq B_0 \times \overline{\mathcal{B}}$ describes

[4]The order-theoretic characterization of a join semilattice leads to simpler expressions.

the assignment of the semantic features to the nouns of the domain B_0. From this formal context, a concept lattice may be constructed (see Section 4.3.3).

REMARK 4.4.3 In the formal context $\langle B_0, \overline{\mathcal{B}}, I \rangle$, the feature corresponding to the greatest element of $\overline{\mathcal{B}}$ may be omitted, because it is assigned to all elements of B_0. Doing this does not change the concept lattice. □

4.4.2 Experimental paradigm

Let B_0 denote a set of nouns out of a certain domain. To assess an individual's semantic structure of B_0, two experiments have to be conducted.

Experiment 1 starts with presenting a randomly selected pair of elements of B_0. The subject is instructed to respond with a noun (eventually modified by adjectives) denoting a concept which exactly contains the two given concepts. If the subject's answer is a_0, we proceed by presenting two randomly selected elements from the set $B_1 = B_0 \cup \{a_0\}$ at the next trial. Continuing in this way, for $k \geq 1$ the set B_k is obtained by $B_k = B_{k-1} \cup \{a_{k-1}\}$. At each trial k, besides the two concepts in question, all elements of B_k are displayed. The experiment is finished regularly after trial l, if each pair of elements of B_l has been presented to the subject, and a_l is already an element of B_l. The results of Experiment 1 can then be formalized by a binary operation on B_l. We do not continue with data collection after trial l too, if the number of elements of B_{l+1} exceeds 25. A total number of $26^2 = 676$ presentations would annoy even a willing subject. In any case, the set resulting from Experiment 1 is denoted by B.

Experiment 2 is a complete pair comparison experiment on the set B. For all pairs out of B, the subject is asked to decide whether the first concept contains the second concept. This results in an empirical relational structure $\langle B, \succeq \rangle$.

4.4.3 An empirical application

Heller (1988) provides a first empirical application of the method proposed above. The study aims at evaluating the experimental procedure, and not at assessing a semantic structure for a large domain. Consequently, the set B_0 should be a small set of nouns with quite obvious semantic features, like $B_0 = \{$man, woman, boy, girl$\}$[5]. The subsequent presentation of the data will show that the results are absolutely nontrivial. Neither all subjects satisfy the above stated representation conditions, nor are they violated by all subjects. The resulting semantic structures show remarkably individual differences.

The experiment was run on a Personal Computer. The subjects were asked to type in their answers on the keyboard, and the context was displayed on

[5]The reported experiments were conducted in German. The translations are as close as possible to the originally used concepts.

the lower half of the screen. The following Tables 4.3, 4.4, and 4.5 summarize the results of both experiments for all subjects.

Table 4.3 contains the number of elements n_c of the set B, and the number of equivalence classes n_e of contextual synonyms. For five out of eight subjects there are no contextual synonyms (i. e. $n_c = n_e$), whereas in the data of Subject 6 many synonyms occur.

Table 4.3. The number of concepts n_c and the number of equivalence classes n_e of contextual synonyms.

Subject	1	2	3	4	5	6	7	8
n_c	10	8	6	9	16	26	9	16
n_e	9	8	6	9	14	18	9	16

Table 4.4 shows the results of testing the conditions for a representation of the relational structure $\langle B, \circ, \sim \rangle$ by a join semilattice (see Theorem 4.4.1). We do not obtain a closed operation $\circ$ on B for Subject 6, because the number of the answers exceeded the limits. This is because the subject always responded with a noun phrase different from the presented ones. The axioms of Theorem 4.4.1 cannot be tested in this case.

Table 4.4. The results of testing the axioms for the empirical structure $\langle B, \circ, \sim \rangle$. The symbols $+$, $-$, $\times$ denote that the corresponding condition is satisfied, not satisfied, or not tested.

Subject	1	2	3	4	5	6	7	8
weak idempotency	$+$	$-$	$+$	$+$	$-$	$\times$	$+$	$-$
weak commutativity	$+$	$-$	$+$	$+$	$-$	$\times$	$+$	$-$
weak associativity	$+$	$-$	$+$	$+$	$-$	$\times$	$+$	$-$
congruence	$+$	$+$	$+$	$+$	$-$	$\times$	$+$	$+$

The data of four subjects can be represented by a join semilattice. This means that semantic generalization from two given concepts, as demanded in the experimental task, is independent from the order of presentation (commutativity). Satisfying associativity implies that the least superordinates of n-tuples of concepts are unique. In the data of the remaining four subjects, there are no systematic violations of the axioms, except for Subject 8 and idempotency, and violations are not limited to a particular axiom. This means that subjects are able to generalize from given concepts in a way consistent with a join semilattice representation.

Table 4.5 contains the results of testing the conditions for the relational structure $\langle \mathcal{B}, \sqsupseteq \rangle$ being a join semilattice. The existence of the least upper bounds for all pairs of elements of $\mathcal{B} = B/\sim$ is tested only if the remaining axioms are satisfied.

Table 4.5. Results of testing the axioms for the empirical structure $\langle \mathcal{B}, \sqsupseteq \rangle$. The symbols $+$, $-$, $\times$ denote that the corresponding condition is satisfied, not satisfied, or not tested.

Subject	1	2	3	4	5	6	7	8
reflexivity	$+$	$+$	$+$	$+$	$+$	$+$	$-$	$+$
transitivity	$+$	$+$	$+$	$+$	$-$	$+$	$-$	$-$
antisymmetry	$+$	$+$	$+$	$+$	$-$	$+$	$-$	$-$
existence of least upper bounds	$+$	$+$	$+$	$+$	$\times$	$+$	$\times$	$\times$

A representation by a join semilattice is possible for the data of five subjects. Three out of these five subjects also satisfy the axioms formulated for Experiment 1. In these cases, the joins and the the least upper bounds coincide (see Proposition 4.3.1). Again, the remaining three subjects do not violate the axioms of reflexivity, transitivity, and antisymmetry systematically. Violating antisymmetry (Subjects 5, 7, and 8) implies that synonymy does not refer to the whole context, the set B. This means that there are $a, b \in B$ with $a \succeq b$ and $b \succeq a$, but not $a \sim b$ (see Definition 4.4.1). Because Subject 6 extensively uses synonyms in Experiment 1, the respective data could not be formalized by an algebraic operation. However, there is a join semilattice representation for the results of Experiment 2. Subject 2 violates the representation conditions for Experiment 1, but also satisfies the axioms in Experiment 2.

Let us now have a look at the individual semantic structures. The following Hasse diagrams illustrate the semantic structures of Subjects 1, 4, and 3, resulting from Experiment 1. The semantic structure of Subject 7 is isomorphic to that of the Subjects 1 and 4.

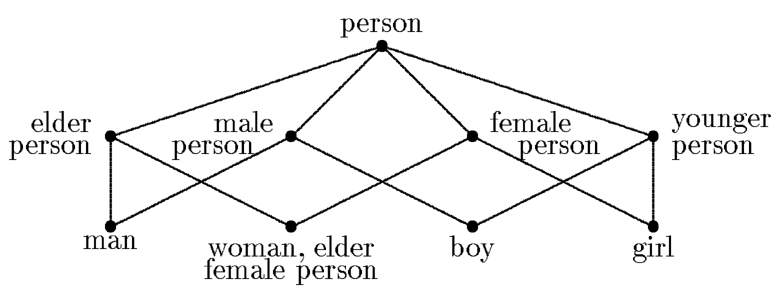


Figure 4.5. Semantic structure of Subject 1.

It is worth mentioning that only the Hasse diagram of Subject 3 is a rooted tree. The semantic structures of the remaining subjects cannot be represented by a rooted tree. Table 4.6 provides an example for a formal context, which is derived from the semantic structure of Subject 4. The formal contexts for the semantic structures of Subjects 1 and 7 are essentially the same as Table 4.6. For Subject 3, only the last column is present.

The Hasse diagram in Figure 4.8 illustrates the results of Subject 2 in

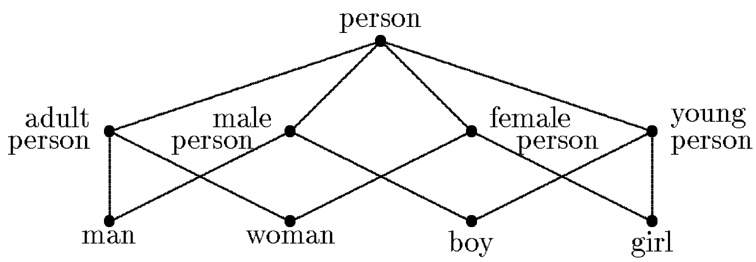


Figure 4.6. Semantic structure of Subject 4.

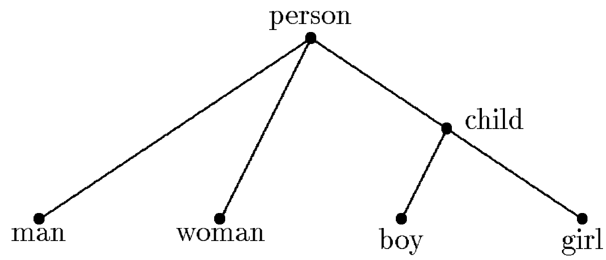


Figure 4.7. Semantic structure of Subject 3.

Experiment 2. Figure 4.9 and Table 4.7 present the results of Subject 6. Experiment 1 could not be finished regularly in this case because of the steadily growing number of distinct answers. However, the Hasse diagram in Figure 4.9 impressively demonstrates that the subject's behavior has not been chaotic. On the contrary, Subject 6 consistently generalized from the concepts presented in Experiment 1 with probably two exceptions. The answers *young man* and *young woman* result from generalizing the concepts *boy* and *girl*, respectively. In Experiment 2, however, they are incompatible to *boy* and *girl*, but are hyponymous to *man* and *woman*.

Figure 4.10 presents an example of a concept lattice derived from the join semilattice representation of semantic structures. The Hasse diagram corresponds to the semantic structures of Subjects 2 and 6.

Consider the data of Experiment 1 (except for Subject 6), which reveal that all but one answer (the noun *child* in the semantic structure of Subject 3) contains the noun *person*, which is the greatest element in the semantic structures. The concepts hyponymous to it are denoted by a combination of

Table 4.6. Formal context derived from the semantic structure of Subject 4.

	sex		age	
	male	female	adult	young
man	×		×	
woman		×	×	
boy	×			×
girl		×		×

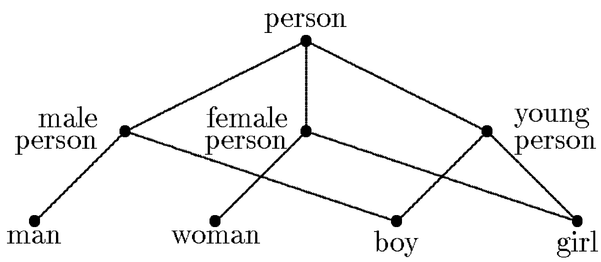


Figure 4.8. Semantic structure of Subject 2.

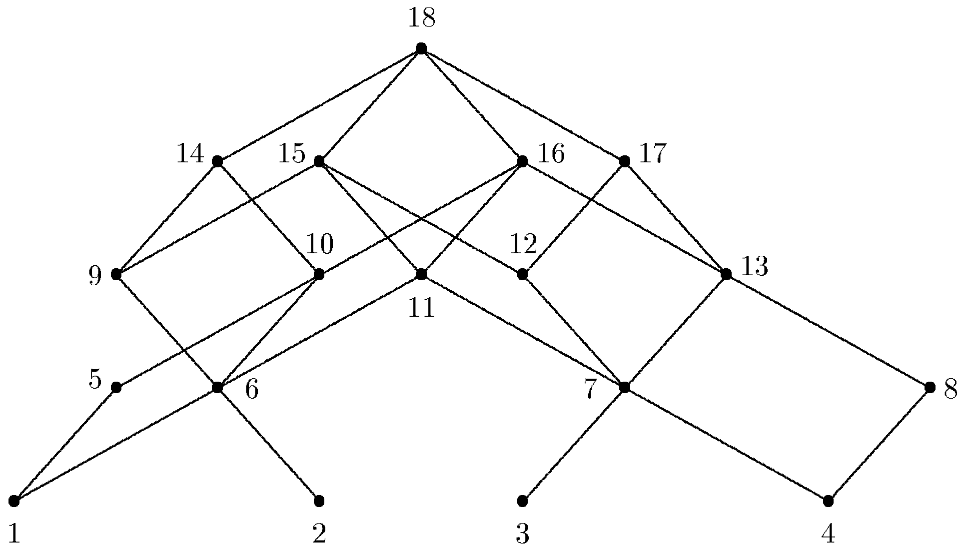


Figure 4.9. Semantic structure of Subject 6. The labels refer to the numbers of the equivalence classes of contextual synonyms in Table 4.7.

one or more adjectives and the noun *person*, like *young person*. This may be because the noun *person* denotes a basic level concept (Rosch, 1978) in the conceptual hierarchies of the semantic structures. A basic level concept has the most inclusive extent with objects having a lot of attributes in common. Subordinated concepts do not have significantly more attributes in common, and superordinated concepts only very few. It seems that, at least within the

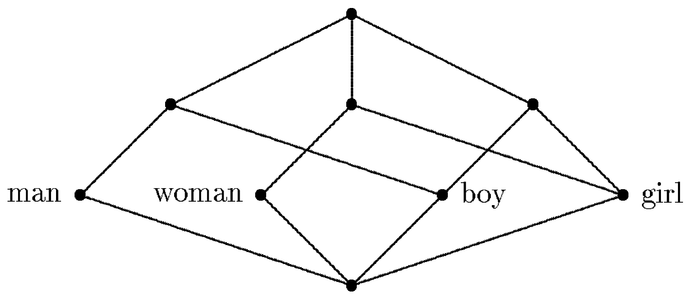


Figure 4.10. Concept lattice corresponding to the semantic structures of Subjects 2 and 6.

Table 4.7. Equivalence classes of contextual synonyms from the semantic structure of Subject 6. The numbers refer to the labels in Figure 4.9.

Number	Equivalence class of contextual synonyms
1	young man
2	boy
3	girl
4	young woman, young female human being
5	man
6	male young person, young male person, young male human being
7	young female person
8	woman
9	young male being, young male living being
10	male person
11	young person, young human being
12	young female living being
13	female person, female human living being, female human being
14	male being
15	young living being
16	person
17	female being, female living being
18	living being

present experimental paradigm, the subjects always start from the basic level concept and express additional common attributes by adding adjectives. Another remarkable result shows up in the data of the Subjects 2, 3, and 6. The superordinate to *man* and *woman* is not *adult* or any synonym, but *person*. This again may result from the predominance of the basic level concept *person* and its intimate relation to the prototypically exemplars denoted by *man* and *woman* (cf. Rosch, 1978).

4.5 General discussion

The preceding sections proposed a method for assessing semantic structures, and presented a lattice-theoretic representation of semantic features.

The method differs from other psychological methods serving this purpose in at least three respects. First, it is designed to assess individual semantic structures, and does not depend on pooling the data of different subjects. Second, the method is not based on global judgments of similarity of meaning, but on semantic generalization. Subjects are asked to respond with a noun phrase expressing the shared features of two given nouns. The method thus directly aims at a feature representation of the meaning of nouns, and does not run into the difficulties, the models based on proximity data are confronted with (see Section 4.2). Third, the theory is formulated according to the principles of measurement theory. This means that the assumptions, the feature representation relies on, are explicitly stated as empirically testable conditions. The results of the reported experiments show that these conditions are not too restrictive.

The lattice-theoretic representation of semantic structures is motivated twice. It is established by considering the ordinal properties of the semantic relation of hyponymy (Sections 4.3.1 and 4.3.2), and it is induced by any assignment of semantic features (Section 4.3.3). Kintsch (1972), however, pointed out that the first argument may be problematic. If we consider the ordinal properties of hyponymy on the set of all nouns, then the existence of least upper bounds is by no means guaranteed. The example in Figure 4.11 illustrates this objection. No least upper bound exists for *dog* and *cat*, since there is no noun denoting the conceptual extent, which in the example consists of *dog* and *cat*.

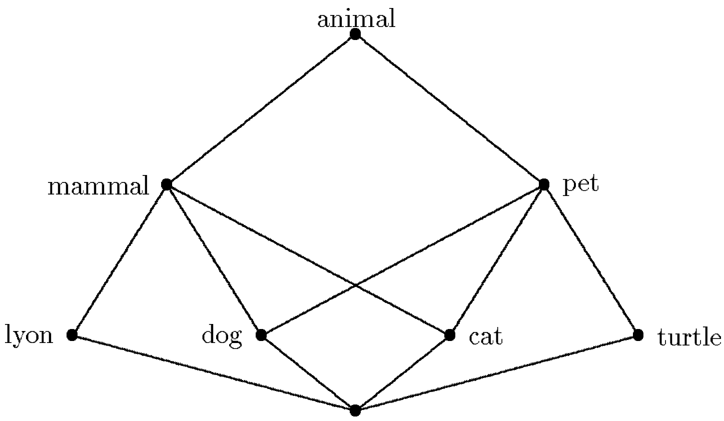


Figure 4.11. Example from Kintsch (1972).

However, this criticism does not apply to the proposed method. On the one hand, the method is not confined to the set of nouns, but to a set of noun phrases, like *mammalian pet*. On the other hand, the measurement-theoretic foundation of the lattice representation assures that its empirical assumptions may be tested in any individual case. If these assumptions are satisfied, then a join semilattice representation exists. The data presented in the preceding section show that a lattice-theoretic representation of semantic structures is adequate for empirical reasons.

It is nevertheless desirable to have a method for assessing semantic structures, which does not rely on verbal reports, and thus avoids the above mentioned problems. Heller (1991) has recently proposed a method that is based on yes-no judgments. Subjects are asked questions like:

Do *orange* and *grapefruit* have any features in common, which are not shared by *apple*?

Founded on theoretical results of the theory of knowledge spaces (Koppen & Doignon, 1990), Heller provides a lattice-theoretic representation of the data resulting from such an experimental paradigm. The semantic structures of a number of domains have already been investigated by this method. Current experiments are concerned with the relationship of the respective results to the feature representations obtained by similarity judgments. This research is still in progress.

References

Beals, R., Krantz, D. H., & Tversky, A. (1968). Foundations of multidimensional scaling. *Psychological Review*, **75**, 127–142.

Birkhoff, G. (1967). *Lattice theory* (3rd ed.). Providence: American Mathematical Society.

Colonius, H., & Schulze, H.-H. (1981). Tree structures for proximity data. *British Journal of Mathematical and Statistical Psychology*, **34**, 167–180.

Davey, B. A., & Priestley, H. A. (1990). *Introduction to lattices and order.* Cambridge: Cambridge University Press.

Fillenbaum, S., & Rapoport, A. (1971). *Structures in the subjective lexicon.* New York: Academic Press.

Ganter, B., & Wille, R. (1989). Conceptual scaling. In F. S. Roberts (Ed.), *Applications of combinatorics and graph theory to the biological and social sciences* (pp. 139–167). New York: Springer.

Heller, J. (1988). Experimentelle Untersuchung der Bildung von Oberbegriffen [Experimental investigation of the formation of superordinated concepts]. *Zeitschrift für Experimentelle und Angewandte Psychologie*, **35**, 74–87.

Heller, J. (1991). *Experimentelle und theoretische Untersuchung zur Begriffsbildung* [Experimental and theoretical investigation of concept formation]. Unpublished doctoral dissertation, Universität Regensburg, Regensburg, Germany.

Johnson, S. C. (1967). Hierarchical clustering schemes. *Psychometrika*, **32**, 241–254.

Kintsch, W. (1972). Notes on the structure of semantic memory. In E. Tulving & W. Donaldson (Eds.), *Organization of memory* (pp. 249–308). New York: Academic Press.

Koppen, M., & Doignon, J.-P. (1990). Building a knowledge space by querying an expert. *Journal of Mathematical Psychology*, **34**, 311–331.

Krantz, D. H., Luce, R. D., Suppes, P., & Tversky, A. (1971). *Foundations of measurement* (Vol. I). New York: Academic Press.

Lyons, J. (1968). *Introduction to theoretical linguistics.* Cambridge: Cambridge University Press.

Miller, G. A. (1969). A psychological method to investigate verbal concepts. *Journal of Mathematical Psychology*, **6**, 169–191.

Osgood, C. E., Suci, G. J., & Tannenbaum, P. H. (1957). *The measurement of meaning.* Urbana: University of Illinois Press.

Rosch, E. (1975). Cognitive reference points. *Cognitive Psychology*, **7**, 532–547.

Rosch, E. (1978). Principles of categorization. In E. Rosch & B. B. Lloyd (Eds.), *Cognition and categorization* (pp. 27–48). Hillsdale: Erlbaum.

Schulze, H.-H., & Colonius, H. (1979). Eine neue Methode zur Erforschung des subjektiven Lexikons [A new method to investigate the subjective lexicon]. In L. H. Eckensberger (Ed.), *Bericht über den 31. Kongreß der Deutschen Gesellschaft für Psychologie in Mannheim 1978* (Vol. 1, pp. 85–88). Göttingen: Hogrefe.

Suppes, P., Krantz, D. H., Luce, R. D., & Tversky, A. (1989). *Foundations of measurement* (Vol. II). New York: Academic Press.

Tversky, A. (1977). Features of similarity. *Psychological Review*, **89**, 327–352.

Wechsler, D. (1955). *Manual for the Wechsler Intelligence Scale.* New York: Psychological Corporation.

Wille, R. (1987). Bedeutungen von Begriffsverbänden [Meanings of concept lattices]. In B. Ganter, R. Wille, & K. E. Wolff (Eds.), *Beiträge zur Begriffsanalye* (pp. 161–211). Mannheim: Bibliographisches Institut.

Wille, R. (1982). Restructuring lattice theory: An approach based on hierarchies of concepts. In I. Rival (Ed.), *Ordered sets* (pp. 445–470). Dordrecht: Reidel.

5 Operations on cognitive structures — their modeling on the basis of graph theory

Erdmute Sommerfeld
Friedrich-Schiller-Universität Jena
Institut für Psychologie, Leutragraben 1, D-07743 Jena, Germany

Fred Sobik
Universität Potsdam
Institut für Computerintegrierte Systeme GmbH
Karl-Liebknecht-Str., D-14476 Golm, Germany

5.1 Knowledge representation — the problem of formation and transformation

It is a central problem in cognitive psychology to identify basic components in processes of formation and transformation of internal representations. Operations on internal representations are essential in different fields of cognitive psychology, e. g. in text comprehension and text processing, problem solving, cognitive development, diagnostics, and differential psychology. Thus Kintsch and van Dijk (1978) and van Dijk and Kintsch (1983) identified cognitive operations for the generation of knowledge representations that are typical for processes in text comprehension and text processing, such as operations for generating hierarchies of propositions and for the formation of macro propositions. Parts of the approach of Kintsch and van Dijk could be specified by experimental results of Beyer (1986). Also the analyses of Schnotz, Ballstaedt, and Mandl (1981), Garrod and Sanford (1981), and Ehrlich (1982) are directed to the identification of general cognitive principles in this field. Schnotz (1988) investigated text comprehension as formation of mental models (Johnson-Laird, 1983). The papers of Rickheit (1991) and Strohner (1991) are based on the idea of mental model formation too. Thereby the question for the basic structure of mental models and for operations for its formation as well as the problem of an adequate formalization are of great importance.

Also in problem solving operations for formation and transformation of internal representations are a wide field of investigation. Sydow (1970) analyzed methods for detecting subjective problem states and its change in the solution process. In different experiments Krause (1970) determined cognitive conditions for the relationship between internal representations and cognitive operations applicable in an effective way. In Klix (1971, 1984) essential components in thinking processes like formation and combination of features as well as processes of abstraction on concepts and operations are described, and their influence on the problem solving process is analyzed. Processes of generalization and differentiation are also discussed and investigated in Dörner (1974, 1976) and Anderson (1976, 1983, 1985, 1988). In Krause (1982) and Krause et al. (1986) task dependent structuring of internal representations is analyzed. Thereby in different experiments an effective task dependent structuring of new information as well as an effective task dependent restructuring of knowledge could be shown (Krause, Sommerfeld, 1988).

Operations for formation and transformation of internal representations are also important in cognitive development (cf. Keil, 1986; Schmidt & Sydow, 1981; Mandler, 1983; Hagendorf, 1985) as well as in differential psychology and diagnostics (cf. Berg & Schaarschmidt, 1984; Berg, 1991). From this point of view one of the main problems in mathematical modeling of cognitive processes is to provide an adequate description of internal representations and operations on internal representations that are fundamental components in different cognitive processes (cf. Spada, 1976; Kluwe & Spada, 1981; Johnson-Laird, 1983; Klimesch, 1988; Mandl & Spada, 1989).

Obviously, such internal representations are of central importance, which represent not only information about elements and their features, but information which is determined by certain relations between these elements. Examples for such relations are grammatical relations between the words of a text or spatial relations between the parts of a picture. This kind of information is called structured or *structural information* (cf. Klix, 1971, 1980).

Within the information processing approach we want to investigate the problem of transfer and processing of structural information. There are external as well as internal representations of structural information. Internal representations are *cognitive structures.* External representations are for example texts or pictures.

Formation and transformation of cognitive structures are connected with processes changing as well the represented structural information as the representing structure. Therefore we have to formalize both *structure* and structural *information.*

5.2 Graphs and structural information in knowledge psychology

In cognitive psychology there is a number of approaches using graph theory to describe knowledge structures and search processes within structures.

A wide field of application of graph theory in cognitive psychology is the representation of knowledge structures. In connection with the question for the organization of knowledge structures the concept "semantic network" plays a central role. In the type/token model Collins and Quillian (1969, 1972) represent super-sub-concept relations (see also Collins & Loftus, 1975; Aebli, 1981). Rumelhart, Lindsay, and Norman (1972) developed a model for the representation of situational relations. Thereby the actions are of essential importance. Anderson and Bower (1973) and Anderson (1983) give model approaches for the representation of hierarchical structured declarative and procedural knowledge. In opposite to this approach Klimesch (1988) analyzes a nonhierarchical connectivity model with structured vertices which represent structured components of knowledge. Klix (1989) distinguishes between object-related and event-related knowledge. Thereby the vertices of a graph represent concepts and the edges represent different relations. The experimental evidence of such representations has been shown.

The central concept of the theory of Falmagne and Doignon (1988) and Doignon (this volume) is that the knowledge state of a subject (with regard to a specified field of information) can be described by a particular subset of questions or problems (in that field) that the subject is capable of solving. The family of all knowledge states (described by the vertices of a representing graph) forms the knowledge space. Connections between knowledge states (described by the edges of the representing graph) indicate inclusion relations with respect to the capability to solve certain problems. Falmagne and Doignon not only use graphs for the representation of knowledge structures but they describe a theory for an efficient assessment of knowledge. Further investigations to formalize knowledge structures can be found in Heller (this volume).

Often problem solving processes are investigated as searching processes through a problem space represented by a graph. The vertices represent (external or internal) problem states and the edges represent transformations of a given state into another one (cf. Duncker, 1935; Mesarovic, 1965; Klix & Goede, 1968; Krause, 1970; Sydow, 1970; Krause et al., 1986).

To make exact statements about formation and transformation of cognitive structures it is necessary to exactly describe operations on cognitive structures. In Klix and Krause (1969) the concept "structure" in psychology is related to the concept "graph" and "adjacency matrix", and in this connection, operations on such structures are related to matrix operations. For certain problem classes Sydow (1980) and Sydow and Petzold (1981) demonstrated possibilities for modeling problems and problem solving processes on the basis of graphs and graph products. Nenniger (1980) applied graph theoretical principles for the formalization of special structure transformations in educational psychology. Strube (1985) modeled biographical knowledge on the basis of comparison and transformation of structures, formalized by special operations on graphs.

There are several approaches for the formalization of the concept of information. Shannon's classical information theory (Shannon & Weaver, 1949) is not useful for describing and measuring structural information. A concept for formalization of information on the basis of structural aspects was given by MacKay (1950). In relation to this concept Leeuwenberg (1968) introduced the "structural information load" as a measure of the information of a perceptive pattern. It is determined by the number of operations necessary to recognize this pattern. Obviously, the "structural information load" is a measure of complexity. Leeuwenberg (1968) and Buffart and Leeuwenberg (1983) developed their "structural information theory" especially for pattern perception processes. Some aspects of this approach are used as a basis for our investigations.

It is necessary to investigate more general aspects of structural information in formation and transformation of internal representations. Both qualitative and quantitative differences between external given and internal represented information have to be exactly described. Furthermore the analysis and formalization of different interpretations of one and the same external representation of a structural information is of importance. For example this plays a role for the investigation of processes of text understanding and problem solving.

Our investigations to modelize operations on cognitive structures continue the idea of modeling special important classes of cognitive operations on the basis of graph theoretical principles (Sommerfeld & Sobik, 1986). Thereby we have to formalize transformations of external given pieces of information to an internal representation of this information as well as transformations of an internal representation into another one.

In this connection it is essential to have an appropriate theory which gives the possibility of characterizing that part of given information which a subject has transformed and represented internally. Moreover it is important whether the information content is changed or not and in the case of change whether it is enlarged or reduced. The internal represented information and its information content are fundamental components in analyzing thinking processes. For example, in diagnostics by means of learning tests it is of importance to know what information the subject has gained.

5.3 Structural information — representation and interpretation

If we want to investigate processes of mapping and processing of structural information we have to take into consideration that information processing is an active process which is determined by the subject. Thus one and the same external representation of a structural information, for example a text or a picture, can be interpreted in a different manner and therefore internally represented in a different manner too. This can be influenced by the context or by certain preknowledge but also by personality factors or motivational components.

Is it possible to formalize the interpretation of texts or pictures with the aim of a formalization of qualitative and and quantitative differences of structural information?

To make a contribution for answering this question it is necessary to specify the concept of structural information.

For this purpose we have to formalize both a structure representing a piece of structural information and different interpretations in the process of the formation of an internal representation on the basis of a piece of external represented structural information.

In the following we want to discuss a formal approach. Based on the formalization of a structure representing structural information a formalization for an interpretation system is introduced. On the basis of this approach it is possible to describe qualitative and quantitative differences between external given and internal represented structural information. Let us begin with the formal description of a structure.

Structural information can be represented by a structure. A structure can be described by means of a relational algebra. Thereby the elements of the structure correspond to the elements of the carrier set of the relational algebra and the relations between the elements of the structure correspond to the relations of the relational algebra. If the set of relations contains only one- and two-figure relations, we can describe such a structure by a graph.

Definition 5.3.1 $G = (V, E, f, g, W_V, W_E)$ is a (finite, labeled, directed) *graph* iff V is a finite nonempty set and $E \subseteq V \times V$, and W_V and W_E are power sets of nonempty sets W_V^* and W_E^*, and $f : V \to W_V$ and $g : V \times V \to W_E$ are functions of V into W_V and $V \times V$ into W_E, respectively. Thereby we have $g((u, v)) = \emptyset$ if $(u, v) \notin E$.

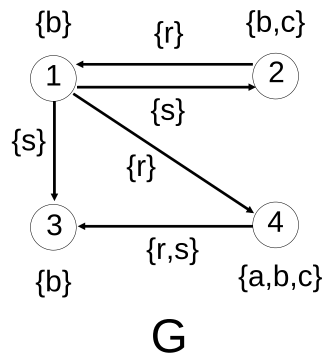
The graph $G' = (V', E', f', g', W_V, W_E)$ is a *subgraph* of G iff $V' \subseteq V$ and $E' \subseteq E \cap (V' \times V')$, and the functions f' and g' are restrictions of f on V' and g on E', respectively, with

$$g'(e) = \begin{cases} g(e), & \text{if } e \in E' \\ \emptyset, & \text{otherwise} \end{cases} \quad \text{for any } e \in V' \times V'.$$

G' is an *induced subgraph* of G iff $E' = E \cap (V' \times V')$. Thereby $G =_{Def.} G\langle V' \rangle = G\langle E^* \rangle$ is induced by the vertex set V' or by an edge set $E^* \subseteq E'$ with the property that for any $u \in V'$ there exists a vertex $v \in V'$ with $(u, v) \in E^*$ or $(v, u) \in E^*$.

The order $|G| =_{Def.} |V|$ of a graph G is the number of its vertices. □

Remark 5.3.1 V is the vertex set and E is the edge set of G. The vertices represent the basic elements of a cognitive structure and the edges represent the relations between these basic elements. W_V and W_E are sets of possible labels of vertices and edges. W_V and W_E contain at least all image points of the functions f and g, respectively. Thereby we assume that these labels represent sets of elementary properties. By means of these labels different types of elements and different relations can be distinguished. Subgraphs and induced subgraphs can represent certain parts of a cognitive structure. Such substructures are important in processes of selecting relevant information, for instance if not the complete given piece of information is necessary for solving the problem under consideration.



G = ({1,2,3,4}, {(1,2),(1,3),(1,4),(2,1),(4,3)},f, g)

with: f(1) = {b}, f(2) = {b,c}, f(3) = {b}, f(4) = {a,b,c}

g((1,2)) = {s}, g((1,3)) = {s}, g((1,4)) = {r}, g((2,1)) = {r}, g((4,3)) = {r,s}.

and $M_V^* \supseteq$ {a,b,c}

and $M_E^* \supseteq$ {r,s}

Figure 5.1. A graph G and its formal description.

Usually in classical graph theory investigations are restricted to unlabeled graphs, i. e. there are no functions f and g and no label sets W_V and W_E and an unlabeled graph is determined only by its vertex set V and its edge set E (cf. Harary, 1969). A possibility to describe an unlabeled graph by means of Definition 5.3.1 is to use the label sets $W_V = \{$'is a vertex'$\}$ and $W_E = \{$'is an edge'$\}$. □

Example 5.3.1 In Figure 5.1 you can see a graph and its formal description. Figure 5.2 shows a graph and its transformation to a more abstract representation. Both graphs can be used as representation of the same structural information. □

In the following we fix the sets W_V and W_E of vertex and edge labels and write for a graph $G = (V, E, f, g, W_V, W_E)$ only the short form $G = (V, E, f, g)$.

Exercise 5.3.1 Give the formal description of the graph G in Figure 5.3. □

Exercise 5.3.2
a) Give the vertex sets and edge sets of the graphs $G_1 = (V_1, E_1, f_1, g_1)$, $G_2 = (V_2, E_2, f_2, g_2)$ and $G_3 = (V_3, E_3, f_3, g_3)$ in Figure 5.4.
b) Is one of these graphs an induced subgraph of another one? Give the inducing vertex set V' and an inducing edge set E^* such that $G = G\langle V' \rangle = G\langle E^* \rangle$ with $i, j \in \{1, 2, 3\}$. □

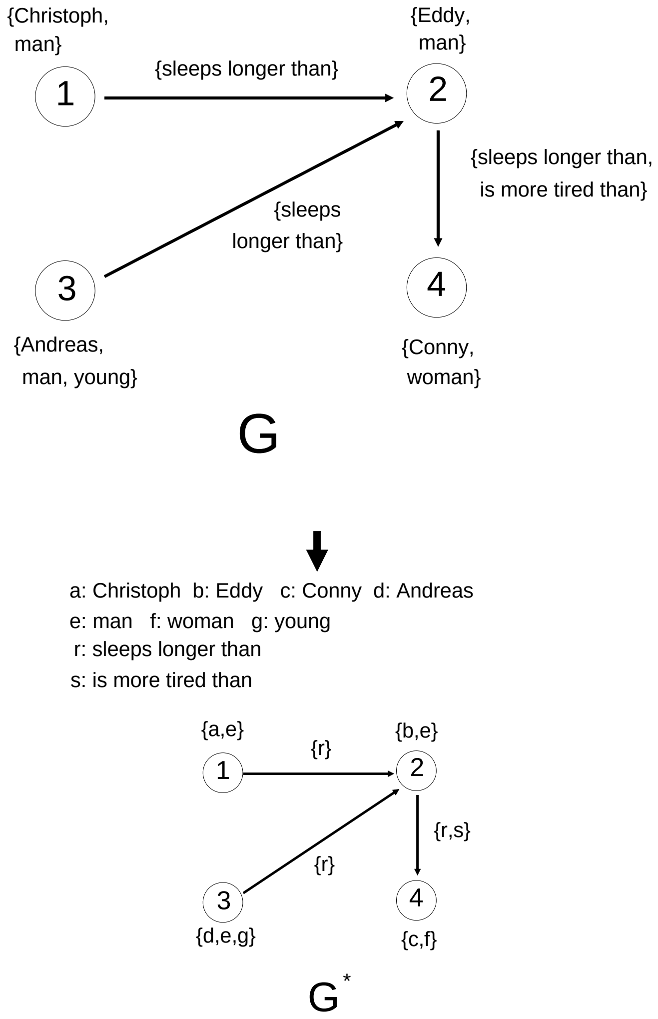


Figure 5.2. Two graphs which can represent equivalent structural information.

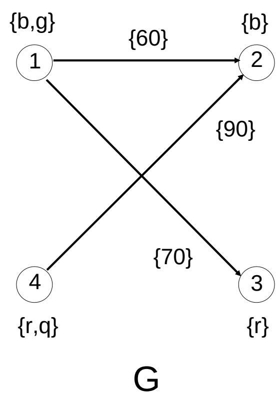


Figure 5.3. Graph G for Exercise 5.3.1.

After the formal description of a structure representing a structural information it is necessary to specify the concept of information represented by

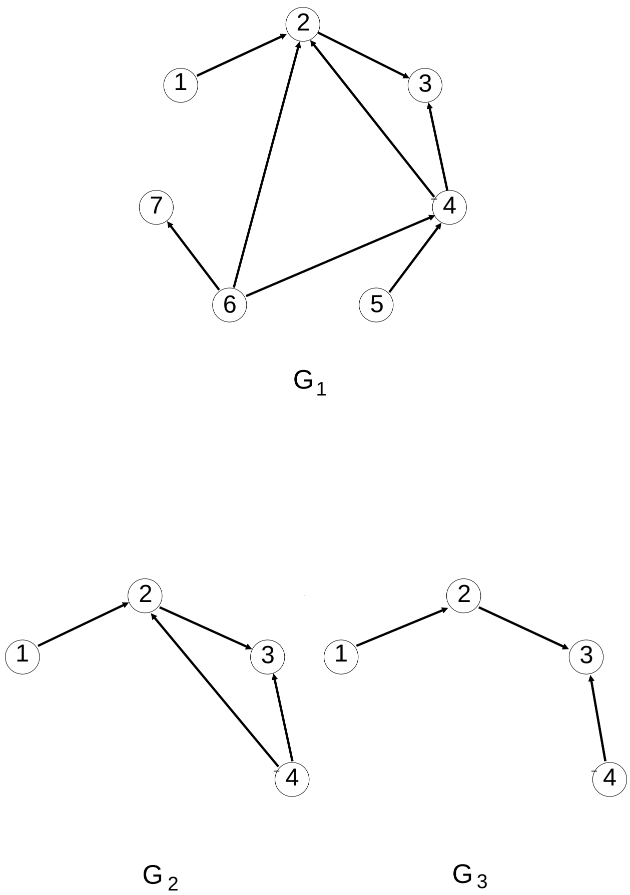


Figure 5.4. Graphs for Exercise 5.3.2.

means of a certain structure.

The structural information content of a structure is the knowledge about the existence of certain relations between elements of this structure, based on existing connections and labelings. In opposite to Shannon's information measure and Leeuwenberg's structural information load the structural information content is not a number, it is given by a set of interpretations. There is a partial order between different pieces of structural information given by the relation of set inclusion.

On the basis of this relation it is possible to compare certain pieces of structural information, but we cannot compare arbitrary pieces of structural information. The structural information content depends on the kind of interpretation of the given structure. Thus at first we have to give a formal description of this kind of interpretation: the definition of an interpretation system. On the basis of this definition we can define the structural information content.

Definition 5.3.2 Let $\mathcal{G}$ be a set of graphs. The tuple $Int = (J, C, s, t)$ is called an *interpretation system* of $\mathcal{G}$, if for any graph $G = (V, E, f, g) \in \mathcal{G}$ the following conditions hold

— J is a set of possible interpretations,
— $C \subseteq J$ is a set of contradictory interpretations,
— s is a function from G into the set V^* of all finite sequences of vertices of G, i. e., $s(G) \subseteq V^*$,
— t is a function from $s(G)$ into J, i. e., for any $w \in s(G)$ we have $t(w) \in J$.

For any graph $G \in \mathcal{G}$ we define the *structural information content* of G with respect to the interpretation system $Int = (J, C, s, t)$ to be

$$I(G, Int) =_{Def.} \{t(w) | w \in s(G), t(w) \notin C\}.$$

The structural information content of the set $\mathcal{G}$ is defined to be

$$I(\mathcal{G}, Int) =_{Def.} \bigcup_{G \in \mathcal{G}} I(G, Int).$$

The structural information content $I(G, Int)$ of a graph G is called greater than the structural information content $I(H, Int)$ of a graph H iff $I(G, Int)$ is a proper subset of $I(H, Int)$, i. e.

$$I(G, Int) < I(H, Int) \leftrightarrow I(G, Int) \subset I(H, Int). \qquad \square$$

REMARK 5.3.2 The set J contains all possible interpretations of the structures under consideration, for example all correct sentences of a certain language. The set C of contradictory interpretations contains all semantically impossible interpretations, for example syntactically correct sentences saying nonsense. By means of the *selection function* s certain sequences of vertices (representing elementary objects) are selected on the basis of structural properties of the representing structure. The *interpretation function* t gives for all selected sequences of vertices an interpretation. □

EXAMPLE 5.3.2 Here we want to select from a given graph $G = (V, E, f, g)$ all sequences of two vertices $u, v \in V$ which are connected by an edge $(u, v) \in E$, i. e., $s(G) = \{(u, v) | (u, v) \in E\}$.

The interpretation of such a pair (u, v) of vertices is given by a sentence from a certain simple language which states that an element with the property x is in relation z to an element with the property y, where x and y are vertex labels and z is an edge label, i. e., $J = \{$"An element with the property x is in relation z to an element with the property y." $| x, y \in W_V, z \in W_E\}$. A possible interpretation function is $t((u, v)) =$ "An element with the property $f(u)$ is in the relation $g((u, v))$ to an element with the property $f(v)$.". An example is a sentence like "An element with the property 'child' is in the relation 'younger than' to an element with the property 'adult'".

The set C of contradictory interpretations could contain for example such sentences like "An element with the property 'mosquito' is in the relation 'greater than' to an element with the property 'elephant'". □

Now let us consider some selection functions s for a given set $\mathcal{G}$ of graphs. The first function we already used in the example given above. We select all ordered pairs of vertices which are connected by an edge. For any graph $G = (V, E, f, g) \in \mathcal{G}$ we define

$$s_e(G) = \{(u, v) | (u, v) \in E\}.$$

By the function s_e all ordered pairs of vertices which are not connected in the graph are not contained in the selection. Sometimes it will be useful also to take into account explicitly these vertices. This can be done by the following selection function

$$s_v(G) = \{(u, v) | u, v \in V\}.$$

The following examples of selection functions are based on graph theoretical properties. At first we can choose all paths in G:

$$\begin{aligned} s_p(G) \quad = \quad & \{(v_1, ..., v_m) | v_1, ..., v_m \in V,\ (v_1, ..., v_m) \text{ is a path in } G, \text{ i.e.,} \\ & (v_i, v_{i+1}) \in E \text{ for all } i = 1, ..., m-1\}. \end{aligned}$$

We can restrict our selection to the maximal paths in G, i.e., to all pathes $(v_1, ..., v_m)$ such that there exists no $v_{m+1} \in V$, that $(v_1, ..., v_m, v_{m+1})$ is a path in G:

$$\begin{aligned} s_{pmax}(G) \quad = \quad & \{(v_1, ..., v_m) | (v_1, ..., v_m) \text{ is a path in } G, \\ & \text{there exists no } v \in V \text{ with } (v, v) \in E\}. \end{aligned}$$

Another possibility is to select all circles in G, i.e., all pathes $(v_1, ..., v_m)$ such that there exists an edge (v_m, v_1) in G:

$$\begin{aligned} s_c(G) \quad = \quad & \{(v_1, ..., v_m) | (v_1, ..., v_m) \text{ is a path in } G, \\ & (v_m, v_1) \in E\}. \end{aligned}$$

And now we consider two examples of interpretation functions for the selection function s_e. The first one results in the reduction of the sentences given by the interpretation function in Example 5.3.2 to its essential components: the properties of the elements and the relations between them. This gives the set

$$J_e = \{(x, y, z) | x, y \in W_V, z \in W_E\}$$

of possible interpretations and for any graph $G = (V, E, f, g)$ and any pair $(u, v) \in s_e(G)$ we define

$$t_e((u, v)) = (f(u), f(v), g((u, v))).$$

The second interpretation function includes more relational context of the elements:

$$t_m((u, v)) = (f(u), f(v), \begin{pmatrix} g((u, u)) & g((u, v)) \\ g((v, u)) & g((v, v)) \end{pmatrix})$$

Exercise 5.3.3 Give a possible complete interpretation system for the interpretation function t_m. □

Example 5.3.3 Often not all labels which are contained in the representing graph G are of interest. Here we want to give an example for a special

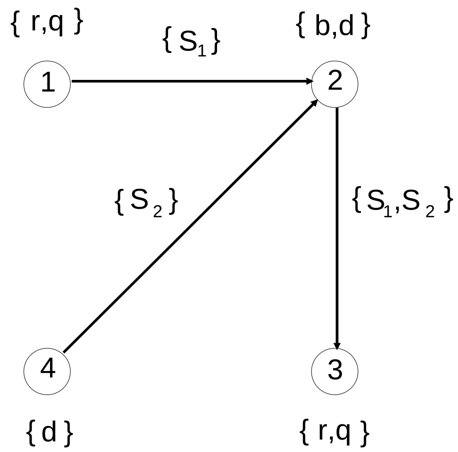


Figure 5.5. The graph for Exercise 5.3.4.

interpretation system in which only one relation is of importance. This relation is represented by edges with the label $r \in W_E$. As the selection function we choose again s_e. Using the set J_e of possible interpretations we have the interpretation system

$$Int_{\{r\}} = (J_e, C, s_e, t_{\{r\}})$$

with

$$t_{\{r\}}((u,v)) = (f(u), f(v), g((u,v)) \cap \{r\})$$
$$\text{for all edges } (u,v) \text{ contained in } s_e(G). \qquad \square$$

EXERCISE 5.3.4
a) The structural information content $I(G, Int_e)$ with $Int_e = (J_e, C, s_e, t_e)$ and $C = \emptyset$ of the graph in Figure 5.5 is to be determined.
b) The structural information content $I(G, Int_{\{S_1\}})$ with $C = \emptyset$ of the graph in Figure 5.5 is to be determined. $\square$

EXERCISE 5.3.5 The structural information content $I(\mathcal{G}, Int_e)$ with $C = \emptyset$ of the graph set $\mathcal{G}$ in Figure 5.6 is to be determined. $\square$

EXAMPLE 5.3.4 Now we want to give an example for the application of the formal approach. In Figure 5.7 we give a text and a related representing graph. Thereby the names of the persons are not included in the vertex labels.

We are able to formalize different possibilities for the interpretation of this text by application of different interpretation systems to the representing graph. Here we choose two selection functions and two interpretation functions. At first we consider the selection function s_e selecting the set of all connected ordered pairs of vertices. The second selection function s_{e*} selects a certain subset of this set.

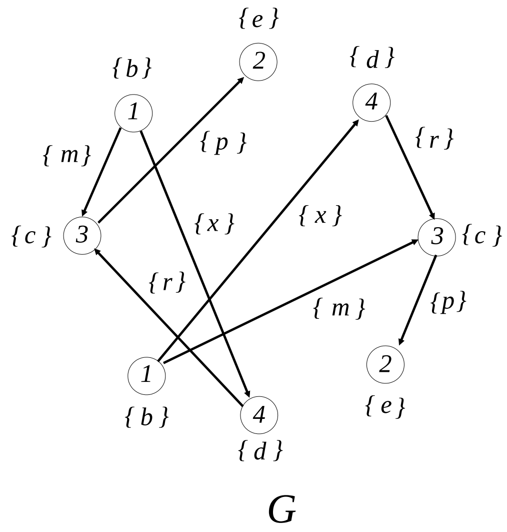


Figure 5.6. Set $\mathcal{G}$ of graphs for Exercise 5.3.5.

The considered interpretation functions t_e and t_m take into account different levels of relational context. These selection and interpretation functions are presented in paragraph 5.3. They are parts of the interpretation systems presented in paragraph 5.3. Also the determination of the structural information content with respect to these interpretation systems is demonstrated in this Figure. □

Application of different selection functions and interpretation functions to the graph G in Figure 5.7.

selection functions:

$s_e(G) = \{(u,v) \mid (u,v) \in E\}$ (set of all connected pairs of vertices)
$s_e^*(G) = \{(u,v) \mid (u,v) \in E, f(u) = \{\text{physician}\}\}$ (set of all connected pairs of vertices, for which the first vertex is labeled by 'physician')

in the given example:
$s_e(G) = \{(H,E),(H,P),(H,T),(E,T),(P,H)\}$
$s_e^*(G) = \{(H,E),(H,P),(H,T)\}$

interpretation functions:

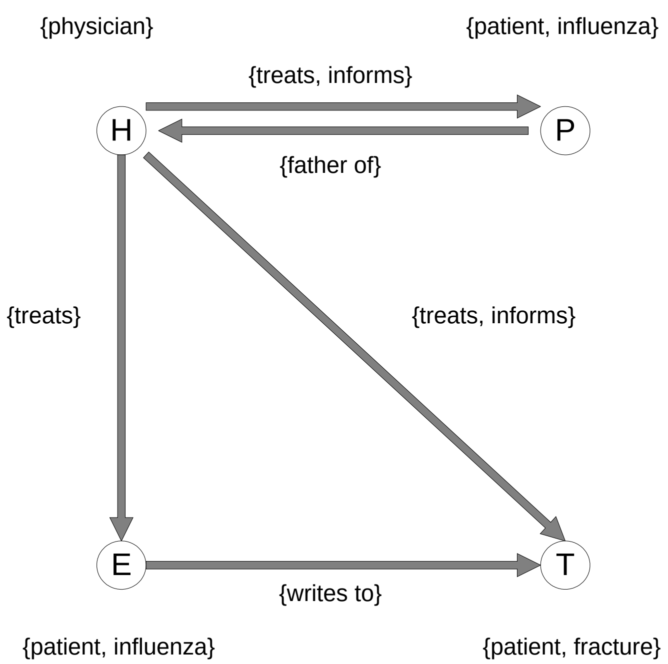
$t_e((u,v)) = (f(u), f(v); g((u,v)))$
$t_m((u,v)) = (f(u), (f(v); \begin{pmatrix} g((u,u)) & g((u,v)) \\ g((v,u)) & g((v,v)) \end{pmatrix})$

in the given example:
$t_e((H,P))$ =({physician},{patient, influenza}; {treats, informs})
$t_m((H,P))$ =({physician},{patient,influenza}; $\begin{pmatrix} \emptyset & \{\text{treats,informs}\} \\ \{\text{father of}\} & \emptyset \end{pmatrix}$)

Determination of structural information content with respect to dif-

Given information:

Harry is a physician. He treats Paul, Theo and Ernest.
Paul has influenza. He is the father of Harry.
Theo has a fracture. Harry informs Theo and Paul.
Ernest has influenza. He writes to Theo.



G

Figure 5.7. Example for a text and a representing graph as basis for different selection functions and interpretation functions (paragraph 5.3) and for the determination of structural information content with respect to different interpretation systems (paragraph 5.3).

ferent interpretation systems.

interpretation systems:

$Int1 = (J_e, \emptyset, s_e, t_e)$
$Int2 = (J_e, \emptyset, s_e^*, t_e)$
$Int3 = (J_e, \emptyset, s_e^*, t_m)$

structural information content of G with respect to these interpretation systems:

$$
\begin{aligned}
I(G, Int1) = \{&(\{\text{physician}\}, \{\text{patient}, \text{influenza}\}; \{\text{treats}\}),\\
&(\{\text{physician}\}, \{\text{patient}, \text{influenza}\}; \{\text{treats}, \text{informs}\}),\\
&(\{\text{physician}, \text{patient}, \text{fracture}\}; \{\text{treats}, \text{informs}\}),\\
&(\{\text{patient}, \text{influenza}\}, \{\text{physician}\}; \{\text{fatherof}\}),\\
&(\{\text{patient}, \text{influenza}\}, \{\text{patient}, \text{fracture}\}; \{\text{wrightsto}\})\}\\
I(G, Int2) = \{&(\{\text{physician}\}, \{\text{patient}, \text{influenza}\}; \{\text{treats}\}),\\
&(\{\text{physician}\}, \{\text{patient}, \text{influenza}\}; \{\text{treats}, \text{informs}\}),\\
&(\{\text{physician}\}, \{\text{patient}, \text{fracture}\}; \{\text{treats}, \text{informs}\})\}
\end{aligned}
$$

$$
\begin{aligned}
&I(G, Int3) =\\
&\quad \{(\{\text{physician}\}, \{\text{patient}, \text{influenza}\}; \begin{pmatrix} \emptyset & \{\text{treats}\} \\ \emptyset & \emptyset \end{pmatrix}),\\
&\quad (\{\text{physician}\}, \{\text{patient}, \text{influenza}\}; \begin{pmatrix} \emptyset & \{\text{treats,informs}\} \\ \{\text{father of}\} & \emptyset \end{pmatrix}),\\
&\quad (\{\text{physician}\}, \{\text{patient}, \text{fracture}\}; \begin{pmatrix} \emptyset & \{\text{treats,informs}\} \\ \emptyset & \emptyset \end{pmatrix})\}
\end{aligned}
$$

$$I(G, Int1) \supset I(G, Int2) \Longleftrightarrow I(G, Int1) > I(G, Int2)$$
(comparable pieces of information)

$$I(G, Int1) \begin{array}{c}\not\supset\\ \not\subset\\ \neq\end{array} I(G, Int3) \Longleftrightarrow I(G, Int1) \begin{array}{c}\not>\\ \not<\\ \neq\end{array} I(G, Int3)$$
(incomparable pieces of information)

$$I(G, Int2) \begin{array}{c}\not\supset\\ \not\subset\\ \neq\end{array} I(G, Int3) \Longleftrightarrow I(G, Int2) \begin{array}{c}\not>\\ \not<\\ \neq\end{array} I(G, Int3)$$
(incomparable pieces of information)

Remark 5.3.3 By means of Example 5.3.4 the concept of comparable and incomparable pieces of information is demonstrated. In the case of comparable pieces of information quantitative differences can be evaluated on the basis of the cardinality of sets of interpretations. □

With this formal approach we have a basis for formalizing qualitative and quantitative differences of structural information content in processes of formation and transformation of cognitive structures. Now we want to systematize and formally describe operations for the formation and transformation of internal representations, based on external structural information.

5.4 Systematization and formalization of cognitive structure transformations

A fundamental situation of psychological relevance is the following one. The cognitive system gets an input information, for instance a text or a picture, which also can contain information about a specified cognitive problem to be solved. For processing this information, for example to understand a text or to solve a given problem, the cognitive system has to generate an internal representation of the information that is externally given, i. e. it has to generate a representing *cognitive structure*. This situation is characterized in Figure 5.8.

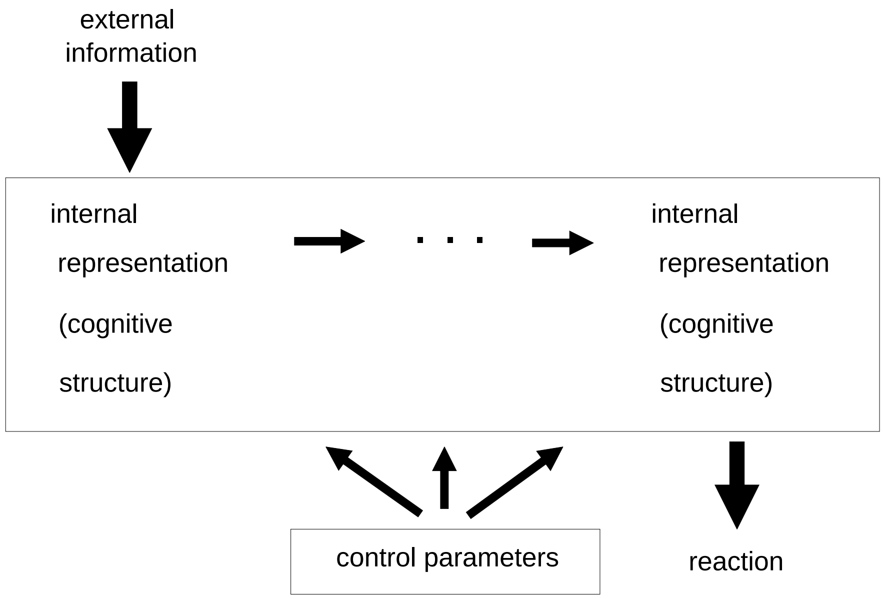


Figure 5.8. Formation and transformation of cognitive structures.

Cognitive processes of formation and transformation of internal representations are connected with processes of selection, structuring, and inference of information, as well as with processes of integration of external given and inferred information into knowledge structures. This is related to processes of knowledge activation, knowledge acquisition, and restructuring of knowledge. In Klix (1971, 1983, 1989), Lompscher (1972), Dörner (1976), and Mehlhorn and Mehlhorn (1985) overviews about cognitive operations in these processes are given. Such operations are for example selection, inference, elaboration, integration and restructuring of information.

It is our aim to formalize such basic cognitive operations to generate the basis for a systematization of hypotheses about formation and transformation of internal representations for different cognitive tasks. Therefore a systematization as well as a formalization of cognitive structure transformations is necessary. On the one hand such a systematization must include *operations of psychological relevance*, on the other hand it must be *complete* under certain restrictions.

To reach these two aims we have taken into consideration all changes of information characterized by means of reduction or enlargement of structural information content in combination with all changes of the representing structure characterized by means of deleting or adding of substructures. Thus against the background of the two properties "structure" and "information" of structural information, we have to analyze two aspects of the transformation of a structure (which represents certain structural information): the influence on the *structure*, and the influence on the *information*. This gives us the opportunity for discussing relationships between structure and information.

Beside such a systematization of *cognitive structure transformations* also a systematization of *graph transformations* has been carried out for the purpose of formalization. Formalization of cognitive structure transformations is a precondition for an exact description of cognitive processes as well as a basis for modeling and simulation, and by this also for the prediction of behavior. Furthermore formalization is a basis for differentiations of cognitive operations and for an analysis of cognitive aspects in detail.

Processes of formation and transformation of internal representations can be accompanied by a reduction or an enlargement of information. Typical cognitive transformations with a reduction of information are processes of selection of information. Selection of information means that only a part of the given external information will be represented internally. Inferred information and its internal representation in combination with the given information characterizes a cognitive structure transformation which results in an enlargement of information.

Both types of changes of structural information content can be connected with an enlargement or a reduction of the representing structure. Thus we have to analyze transformations which enlarge or reduce the structural information content in combination with possible enlargements or reductions of the structure which represents this information. Furthermore these cases must be combined with transformations that do not change the structural information content or the structure, respectively. Thus we have a definite set of combinations between variations of structure and information content. From this point of view now we analyze cognitive structure transformations and graph transformations for the purpose of their formalization.

At first it is necessary to ask for elementary graph transformations which can be the basis for the formalization of cognitive structure transformations. For this purpose we have investigated systematically the change of all parameters characterizing a structure represented by a graph $G = (V, E, f, g)$.

5.4.1 Elementary graph transformations

On the basis of enlargement or reduction of the sets $V, E, f(v), g(e)$ $(v \in V, e \in E)$ of a graph $G = (V, E, f, g)$ we get the following set of elementary graph transformations.

Definition 5.4.1 Let $G = (V, E, f, g)$ and $G' = (V', E', f', g')$ be graphs. G' is obtained from G by application of the elementary graph transformation of

— **addition** φ_v^+ **of a vertex** v $(v \notin V)$ (with an empty label) iff

$$V' = V \cup \{v\}, E' = E, g' = g,$$

$$f'(u) = \begin{cases} \emptyset, \text{ if } u = v, \\ f(u), \text{ otherwise} \end{cases} \text{for any } u \in V',$$

— **addition** φ_e^+ **of an edge** e $(e \notin E)$ (with an empty label) iff

$$V' = V, E' = E \cup \{e\}, f' = f,$$

$$g'(h) = \begin{cases} \emptyset, \text{ if } h = e, \\ g(h), \text{ otherwise} \end{cases} \text{for any } h \in E',$$

— **addition** $\varphi_{v,MV}^+$ **of a vertex label** $MV \in W_V$ to a vertex $v \in V$ iff

$$V' = V, E' = E, g' = g,$$

$$f'(u) = \begin{cases} f(v) \cup MV, \text{ if } u = v, \\ f(u), \text{ otherwise} \end{cases} \text{for any } u \in V',$$

— **addition** $\varphi_{e,ME}^+$ **of an edge label** $ME \in W_E$ to an edge $e \in E$ iff

$$V' = V, E' = E, f' = f,$$

$$g'(h) = \begin{cases} g(e) \cup ME, \text{ if } h = e, \\ g(h), \text{ otherwise} \end{cases} \text{for any } h \in E',$$

— **deletion** φ_v^- **of an (isolated) vertex** $v \in V$ (with an empty label) iff

$$f(v) = \emptyset \text{ and}$$

$$\text{there exists no vertex } u \in V \text{ with } (u, v) \in E \text{ or } (v, u) \in E,$$

$$V' = V \setminus \{v\}, E' = E, g' = g, f'(u) = f(u) \text{ for any } u \in V',$$

— **deletion** φ_e^- **of an edge** e $(e \in E)$ (with an empty label) iff

$$g(e) = \emptyset \text{ and}$$

$$V' = V, E' = E \setminus \{e\}, f' = f, g'(h) = g(h) \text{ for any } h \in E',$$

— **deletion** $\varphi_{v,MV}^-$ **of a vertex label** $MV \in W_V$ from a vertex $v \in V$ iff

$$V' = V, E' = E, g' = g,$$

$$f'(u) = \begin{cases} f(v) \setminus MV, & \text{if } u = v, \\ f(u), & \text{otherwise} \end{cases} \quad \text{for any } u \in V',$$

— **deletion** $\varphi^{-}_{e,ME}$ **of an edge label** $ME \in W_E$ from an edge $e \in E$ iff

$$V' = V, E' = E, f' = f,$$

$$g'(h) = \begin{cases} g(e) \setminus ME, & \text{if } h = e, \\ g(h), & \text{otherwise} \end{cases} \quad \text{for any } h \in E'.$$

□

On the basis of combinations of these elementary graph transformations we get more complex transformations. For instance an addition of a labeled vertex or edge can be obtained by successive application of the elementary transformations φ^{+}_{v} and $\varphi^{+}_{v,MV}$ or φ^{+}_{e} and $\varphi^{+}_{e,ME}$, respectively.

Beside such changes of graph structures also operations for combining several structures into one unit are of importance and therefore they must be formalized. There exist different operations for the combination of graphs (cf. for example Harary, 1969). As an important operation for the connection of cognitive structures here we characterize the operation of graph union.

Definition 5.4.2 Let $G_1 = (V_1, E_1, f_1, g_1)$ and $G_2 = (V_2, E_2, f_2, g_2)$ be graphs.

Then we define the union $G_{\cup}$ of G_1 and G_2 to be the graph

$$G_{\cup} = G_1 \cup G_2 = (V_1 \cup V_2, E_1 \cup E_2, f_{\cup}, g_{\cup})$$

with

$$f_{\cup}(v) = \begin{cases} f_1(v), & \text{if } v \in V_1, v \notin V_2 \\ f_2(v), & \text{if } v \notin V_1, v \in V_2 \\ f_1(v) \cup f_2(v), & \text{otherwise} \end{cases}$$

and

$$g_{\cup}(v) = \begin{cases} g_1(e), & \text{if } e \in E_1, e \notin E_2 \\ g_2(e), & \text{if } e \notin E_1, e \in E_2 \\ g_1(e) \cup g_2(e), & \text{otherwise} \end{cases}$$

$G_1 \cup G_2$ is the operation of *union* of the graphs G_1 and G_2, that means it is the transformation of the graph set $\{G_1, G_2\}$ into the united structure $G_1 \cup G_2$. For the transformation of G_1 into $G_1 \cup G_2$ we write $\varphi_{\cup,G_2}(G_1) =_{Def.} G_1 \cup G_2$.

□

Remark 5.4.1 A graph union can be obtained by successive application of the elementary graph transformations φ^{+}_{v} of vertex addition, $\varphi^{+}_{v,MV}$ of vertex label addition, φ^{+}_{e} of edge addition, and $\varphi^{+}_{e,ME}$ of edge label addition. Because the property of symmetry of the operation of set union holds $\varphi_{\cup,G_1}(G_2) = \varphi_{\cup,G_2}(G_1)$. □

Example 5.4.1 In Figure 5.9 you can see two examples of graph unions.

□

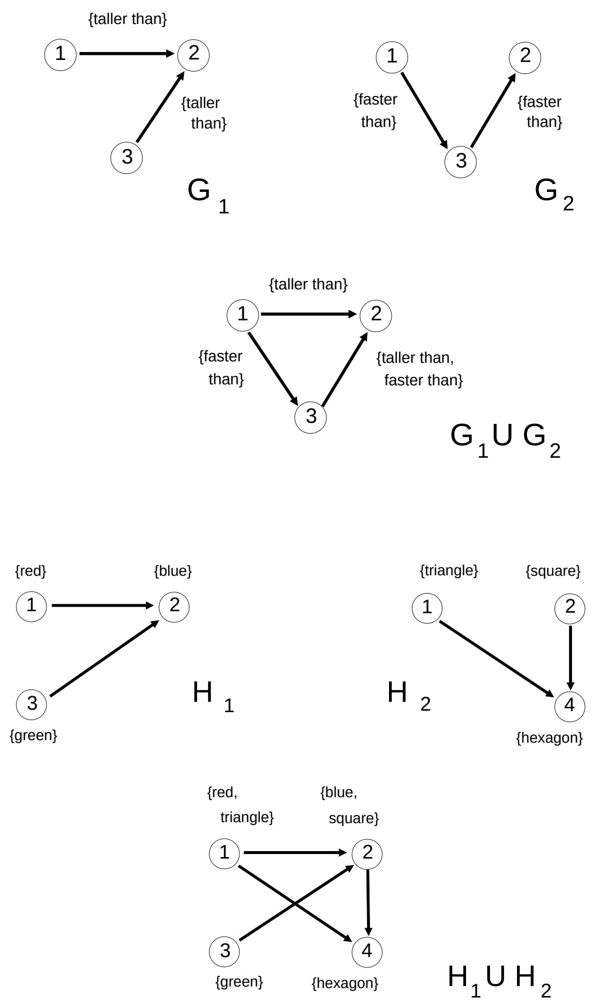


Figure 5.9. Examples for the transformation of graph union.

EXERCISE 5.4.1 For the union $G_1 \cup G_2$ of the graphs of Figure 5.10 give:
a) the drawing,
b) the formal description. □

Combinations of these elementary graph transformations give a basis for the formalization of cognitive structure transformations which enlarge, reduce or do not change the structural information content of a given structural information.

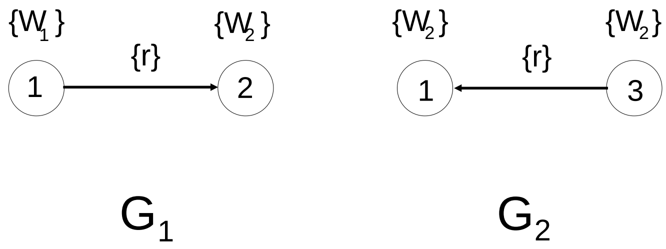


Figure 5.10. Graphs G_1 and G_2 for Exercise 5.4.1.

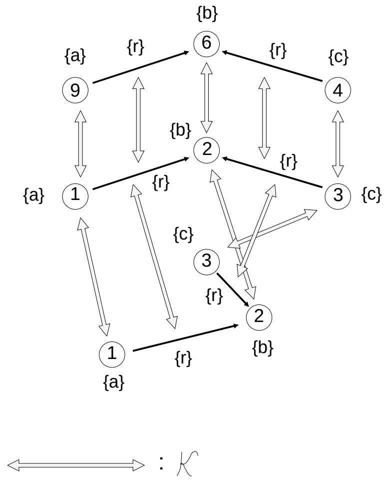


Figure 5.11. Example for isomorphic mappings.

Thereby we restrict our considerations to interpretation systems of the form $Int_e = (J_e, C, s_e, t_e)$.

5.4.2 Formalization of cognitive structure transformations without change of structural information content

Transformations that *do not effect any change in the structure* and therefore also do not effect any change of the information content, that means transformations that preserve both the structure and the information content, can be described by an isomorphic mapping of two graphs.

Definition 5.4.3 Let $G = (V, E, f, g)$ and $H = (V', E', f', g')$ be graphs. G and H are *isomorphic* ($G \cong H$) iff there exists an one-to-one mapping κ from $V \cup E$ onto $V' \cup E'$ with $\kappa(v) \in V'$ for any $v \in V$, $\kappa(e) \in E'$ for any $e \in E$, $\kappa((u, v)) = (\kappa(u), \kappa(v))$ for any $u, v \in V$ and $(u, v) \in E$, $f(v) = f'(\kappa(v))$ for any $v \in V$ and $g(e) = g'(\kappa(e))$ for any $e \in E$.

The isomorphism class of a graph G is the class of all possible graphs which are isomorphic to G.

The transformation $\varphi_{so}(G) =_{Def.} H$ is called *isomorphic mapping.* □

Example 5.4.2 In Figure 5.11 three isomorphic graphs and the corresponding isomorphic mapping κ is shown. □

Exercise 5.4.2 For the graphs G, H_1, H_2, H_3 in Figure 5.12 give the correct answer to the following questions:
a) $G \cong H_1$?
b) $G \cong H_2$?
c) $G \cong H_3$? □

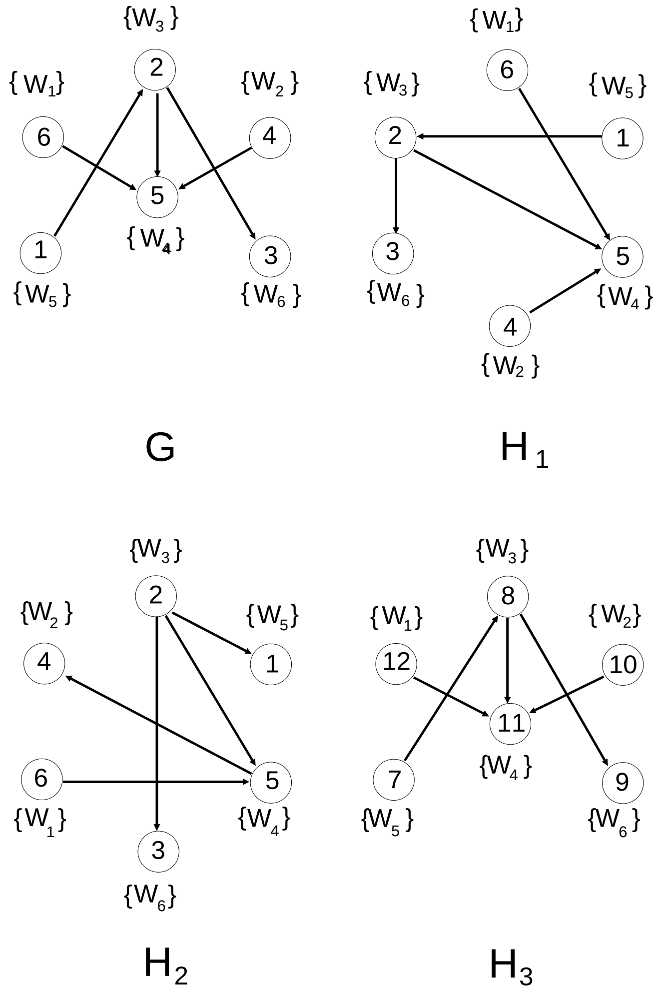


Figure 5.12. Graphs G, H_1, H_2 and H_3 for Exercise 5.4.2.

Lemma 5.4.1 *Let $G = (V, E, f, g)$ be a graph. Then we have $I(G, Int_e) = I(\varphi_{so}(G), Int_e)$.* □

Transformations which result in *adding or deleting of structures* without changing the structural information content consist in adding or deleting redundancy. They can be formalized on the basis of adding or deleting of graphs which are isomorphic to parts of the original graph. If the following conditions are fulfilled it is possible to describe such cognitive transformations by the elementary graph transformations φ_v^+, φ_e^+, $\varphi_{v,MV}^+$, $\varphi_{e,ME}^+$ and φ_v^-, φ_e^-, $\varphi_{v,MV}^-$, $\varphi_{e,ME}^-$ and by their combinations.

Lemma 5.4.2 *Let $G = (V, E, f, g)$ be a graph. Then we have, if $G' = (V', E', f', g')$ denotes the graph obtained by application of the respective elementary graph transformation to G, the following results:*

1. $I(G, Int_e) = I(\varphi_v^+(G), Int_e)$.

2. $I(G, Int_e) = I(\varphi_e^+(G), Int_e)$ *with* $e = (u, v)$ *iff* $(f'(u), f'(v), g'((u, v)) \in C$ *or there exists an edge* $(w, z) \in E$ *with* $(f(w), f(z), g((w, z))) = (f'(u), f'(v), g'((u, v)))$.
3. $I(G, Int_e) = I(\varphi_{v,MV}^+(G), Int_e)$ *iff* $MV \subseteq f(v)$ *or* *for any* $(u, v) \in E'$ *with* $(f'(u), f'(v), g'((u, v))) \notin C$ *exists an edge* $(w, z) \in E$ *with* $(f(w), f(z), g((w, z))) = (f(u), f(v) \cup MV, g((u, v))) = (f'(u), f'(v), g'((u, v)))$ *and* *for any* $(v, u) \in E'$ *with* $(f'(v), f'(u), g'((v, u))) \notin C$ *exists an edge* $(w, z) \in E$ *with* $(f(w), f(z), g((w, z))) = (f(v) \cup MV, f(u), g((v, u))) = (f'(v), f'(u), g'((v, u)))$ *and* *for any* $(u, v) \in E$ *with* $(f(u), f(v), g((u, v))) \notin C$ *exists an edge* $(w, z) \in E'$ *with* $(f'(w), f'(z), g'((w, z))) = (f(u), f(v), g((u, v)))$ *and* *for any* $(v, u) \in E$ *with* $(f(v), f(u), g((v, u))) \notin C$ *exists an edge* $(w, z) \in E'$ *with* $(f'(w), f'(z), g'((w, z))) = (f(v), f(u), g((v, u)))$.
4. $I(G, Int_e) = I(\varphi_{e,ME}^+(G), Int_e)$ *with* $e = (u, v)$ *iff* $ME \subseteq g(e)$ *or* $[[(f'(u), f'(v), g'((u, v))) \in C$ *or exists an edge* $(w, z) \in E$ *with* $(w, z) \neq (u, v)$ *and* $(f(w), f(z), g((w, z))) = (f(u), f(v), g((u, v)) \cup ME) = (f'(u), f'(v), g'((u, v)))]$ *and* $[(f(u), f(v), g((u, v))) \in C$ *or exists an edge* $(w, z) \in E$ *with* $(w, z) \neq (u, v)$ *and* $(f(w), f(z), g((w, z))) = (f(u), f(v), g((u, v)))]]$.
5. $I(G, Int_e) = I(\varphi_v^-(G), Int_e)$.
6. $I(G, Int_e) = I(\varphi_e^-(G), Int_e)$ *with* $e = (u, v)$ *iff* $(f(u), f(v), g((u, v)) \in C$ *or there exists an edge* $(w, z) \neq (u, v)$ *with* $(w, z) \in E$ *and* $(f(w), f(z), g((w, z))) = (f(u), f(v), g((u, v)))$.
7. $I(G, Int_e) = I(\varphi_{v,MV}^-(G), Int_e)$ *iff* $MV \cap f(v) = \emptyset$ *or* *for any* $(u, v) \in E'$ *with* $(f'(u), f'(v), g'((u, v))) \notin C$ *exists an edge* $(w, z) \in E$ *with* $(f(w), f(z), g((w, z))) = (f(u), f(v) \setminus MV, g((u, v))) = (f'(u), f'(v), g'((u, v)))$ *and* *for any* $(v, u) \in E'$ *with* $(f'(v), f'(u), g'((v, u))) \notin C$ *exists an edge* $(w, z) \in E$ *with* $(f(w), f(z), g((w, z))) = (f(v) \setminus MV, f(u), g((v, u))) = (f'(v), f'(u), g'((v, u)))$ *and* *for any* $(u, v) \in E$ *with* $(f(u), f(v), g((u, v))) \notin C$ *exists an edge* $(w, z) \in E'$ *with* $(f'(w), f'(z), g'((w, z))) = (f(u), f(v), g((u, v)))$ *and* *for any* $(v, u) \in E$ *with* $(f(v), f(u), g((v, u))) \notin C$ *exists an edge* $(w, z) \in E'$ *with* $(f'(w), f'(z), g'((w, z))) = (f(v), f(u), g((v, u)))$.
8. $I(G, Int_e) = I(\varphi_{e,ME}^-(G), Int_e)$ *with* $e = (u, v)$ *iff* $ME \cap g(e) = \emptyset$ *or* $[[(f'(u), f'(v), g'((u, v))) \in C$ *or exists an edge* $(w, z) \in E$ *with* $(w, z) \neq (u, v)$ *and* $(f(w), f(z), g((w, z))) = (f(u), f(v), g((u, v)) \setminus ME) = (f'(u), f'(v), g'((u, v)))]$ *and* $[(f(u), f(v), g((u, v))) \in C$ *or exists an edge* $(w, z) \in E$ *with* $(w, z) \neq$

(u, v) and $(f(w), f(z), g((w, z))) = (f(u), f(v), g((u, v)))]].$ □

REMARK 5.4.2 By these elementary transformations and their combinations (which change the structure but do not change the structural information content) we enlarge or reduce the redundancy of the representing structure. Isolated vertices, i. e., vertices v (also with nonempty label) for which do not exist an edge (u, v) or an edge (v, u) are redundant, i. e., adding or deleting of such isolated vertices does not change the structural information content $I(G, Int_e)$ (1. and 5.).

With respect to the other elementary transformations it holds that for any noncontradictory substructure of the original graph there must exist a substructure of the generated graph G' with the same structural information content. Vice versa for any generated noncontradictory substructure of G' there must exist a substructure of the original graph G with the same structural information content.

For the cases 2., 4., 6., 8. we can state that if a noncontradictory substructure of the graph is changed by a transformation there must exist an unchanged substructure of the original graph with an equal structural information content and for any generated noncontradictory substructure there must also exist a substructure of the original graph with an equal structural information content. □

Another case consists in a connection of different substructures without changing the structural information content. Thereby different possibilities for the connection of substructures exist. For example identical elements of a structure can be connected by adding an edge representing this identity (coherence graph, Kintsch, 1974). This can be described by means of a specific combination of the graph transformations φ_e^+ and $\varphi_{e,ME}^+$.

Cognitive operations which combine different structures to a unit (without changing the structural information content) can be formally described on the basis of certain graph unions.

There are some special cases in which also the transformation of graph union does not change the structural information content.

LEMMA 5.4.3 *Let $G_1 = (V_1, E_1, f_1, g_1)$ and $G_2 = (V_2, E_2, f_2, g_2)$ be graphs. If $G_1 \cong G_2$ and $V_1 \cap V_2 = \emptyset$ then*

$$I(G_1, Int_e) = I(G_2, Int_e) = I(G_1 \cup G_2, Int_e).$$

If G_2 is an induced subgraph of G_1 then

$$I(G_1, Int_e) = I(G_1 \cup G_2, Int_e).$$ □

5.4.3 Formalization of cognitive structure transformations with enlargement of structural information content

Cognitive structure transformations which result in an enlargement of structural information content characterize a relevant class of cognitive operations involved in generating or elaborating information (Posner, 1976; Rickheit & Strohner, 1985; Krause, Sommerfeld, & Zießler, 1991).

Transformations with *enlargement of structure* and with enlargement of information content can be based on adding parts of knowledge structures from long term memory (LTM) or on adding knowledge inferred on the basis of rules stored in LTM. Adding of parts of knowledge structures can be based for instance on concept structures, on situational structures, or structures that characterize actions stored in LTM. Let us consider a simple example. If the external information "tree" is given, in general, we activate in our LTM the knowledge that a tree has branches and limbs, and add these properties to the knowledge in our working memory.

Adding of knowledge inferred on the basis of rules stored in the LTM is denoted as inference. Again a simple example. If the external information is given by the propositions 'a is greater than b' and 'b is greater than c' we can use the knowledge that 'greater' is a transitive relation and can apply the rule of transitive inference. By this rule we obtain the proposition 'a is greater than c'. This inferred information can be added to the working memory.

This is the case for other types of inferences too. Thus analogical inferences are a special case of cognitive processes with an enlargement of information. The procedure of projection carried out in these processes (Klix, 1990) is based on a mapping of structural properties of one domain of knowledge into another one. The carrier structure of a given piece of information is connected with new properties. If this new information is added to the knowledge about the original information we have an enlargement of structural information content in connection with an enlargement of the representing structure.

If the given and the added information are combined into a unit we get an integrated cognitive structure: The external information (for instance a set of propositions) is combined into a unit with the information gained (for example a set of inferred propositions). Thus integration of information is a cognitive operation that results in an enlargement of information content and in a combination of different cognitive structures into a unit.

Beside this important type of cognitive structure transformations characterized by an enlargement of both information content and structure we must ask for possibilities of cognitive structure transformations with an enlargement of structural information content but with a *reduction of structure*. A possibility for transformations with an enlargement of structural information content but with a reduction of structure is the deletion of contradictions.

Transformations with an enlargement of structural information content *without changing the structure* are impossible.

All these types of cognitive structure transformations can be formally described by graph transformations with enlargement of structural information content.

Graph transformations resulting in an enlargement of structural information content on the basis of enlargement of the original graph are based on adding certain vertices, edges or labels to the original graph which are not in contradiction to parts of the original graph.

Graph transformations resulting in an enlargement of structural information content on the basis of a reduction of the original graph are based on deleting certain vertices, edges or labels from the original graph, for example that are in contradiction to other parts of the original graph.

Preconditions for such graph transformations can be obtained on the basis of the elementary graph transformations described in 5.4.1. Analyses of these transformations can be carried out in an analogous way to Lemma 5.4.2. Here we want to give only a short characterization of this class of elementary graph transformations. For a more detailed formal approach cf. Sommerfeld (1991).

By such elementary transformations and their combinations (which enlarge or reduce the structure) we enlarge the structural information content of the representing structure. Isolated vertices, i. e., vertices v (also with nonempty label) for which do not exist an edge (u, v) or an edge (v, u) are redundant, i. e., adding or deleting of such isolated vertices does not enlarge the structural information content $I(G, Int_e)$.

In general on the one hand there must exist a noncontradictory substructure of the generated graph G' such that there does not exist a substructure of the original graph G with the same structural information content. On the other hand it is necessary that for any noncontradictory substructure of the original graph G there must exist a substructure of the generated graph G' with the same structural information content.

On the basis of such elementary graph transformations and their combinations all types of cognitive structure transformations with enlargement of structural information content discussed above can be formally described.

Here we want to define a combination which is very important for formalizing and modeling inference processes. The process of inferring all propositions — based on certain properties of a given set of propositions — can be described by means of the process of hull formation of graphs.

This graph transformation can be obtained by successive application of the elementary graph transformations φ_e^+ of edge addition and $\varphi_{e,ME}^+$ of edge label addition. Such a transformation can be realized step by step by means of a generating transformation and a combining transformation. Here we want to consider the transformation of transitive hull formation which is based on the generating transformation of transitive supplement formation and the combining transformation of graph union.

Definition 5.4.4 Let $G = (V, E, f, g)$ be a graph with $g(e) = \{r\}$ for all $e \in E$ and let $G^{(0)} = (V, E^{(0)}, f, g^{(0)}) = G$.

For $i = 1, 2, \ldots$ we define $E_t^{(i)} \subseteq V \times V$ with:
For all $v_i, v_j, v_k \in V$ holds: $(v_i, v_k) \in E_t^{(i)}$ iff $(v_i, v_j), (v_j, v_k) \in E^{(i-1)}$ and $(v_i, v_k) \notin E^{(i-1)}$. Then we define $G_t^{(i)} = (V_t^{(i)}, E_t^{(i)}, f_t^{(i)}, g_t^{(i)})$ whereas $V_t^{(i)}$ is the vertex set of the subgraph induced by $E_t^{(i)}$ in $G^* = (V, V \times V, f^*, g^*)$,

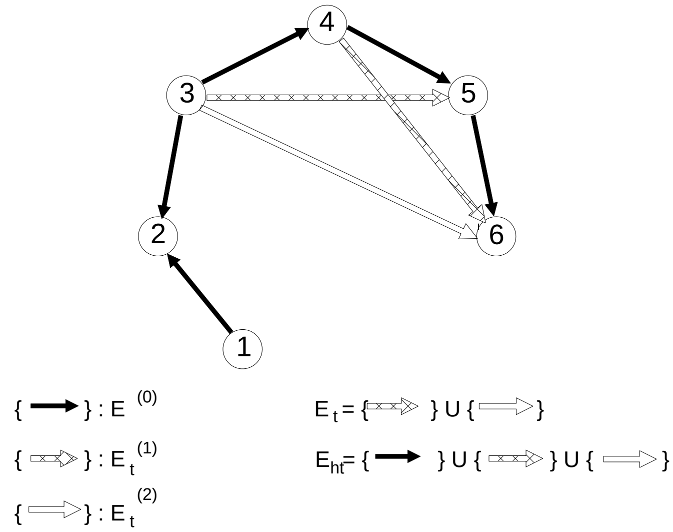


Figure 5.13. Example for the formation of transitive supplement and transitive hull.

and $f_t^{(i)}(v) = f(v)$ for all $v \in V_t^{(i)}$ and $g_t^{(i)}(e) = \{r\}$ for all $e \in E_t^{(i)}$ and $G^{(i)} = G^{(i-1)} \cup G_t^{(i)}$.

If $E_t^{(i)} = \emptyset$, then the graph $G_t =_{Def.} G_t^{(1)} \cup \ldots \cup G_t^{(i-1)} = (V, E_t, f, g_t)$, with $E_t = E_t^{(1)} \cup \ldots \cup E_t^{(i-1)}$ is the *transitive supplement* of G, and the transformation $\varphi_t(G) =_{Def.} G_t$ is called *transitive supplement formation.*

The graph $G_{ht} =_{Def.} G \cup G_t = (V, E_{ht}, f, g_{ht})$ with $E_{ht} = E \cup E_t$ is the *transitive hull* of G and the transformation $\varphi_{ht}(G) =_{Def.} G_{ht}$ is called *transitive hull formation.* □

Remark 5.4.3 The transitive supplement formation can be used to describe the information which can be obtained by application of the rule of transitive inference to the original graph. The united given and inferred structural information can be characterized by means of the transitive hull. □

Example 5.4.3 In Figure 5.13 an example of transitive supplement formation and transitive hull formation is presented. □

Exercise 5.4.3 Determine for the graph $G = (V, E, f, g)$ in Figure 5.14:
a) $G_t^{(1)}, G_t^{(2)}, \ldots$
b) the transitive supplement G_t
c) the transitive hull G_{ht}. □

Lemma 5.4.4 *Let $G = (V, E, f, g)$ be a graph with $g(e) = \{r\}$ for all $e \in E$, let $\varphi_{ht}(G) = G_{ht}$ be its transitive hull, and let $\varphi_t(G) = G_t = (V, E_t, f, g_t)$ be its transitive supplement. Then we have $I(G, Int_e) \subseteq I(G_{ht}, Int_e)$ and $[I(G, Int_e) \subset I(G_{ht}, Int_e)$ iff there exists an edge $e = (u, v) \in E_t$ with $(f(u), f(v), g(e)) = (f(u), f(v), \{r\}) \notin C]$.* □

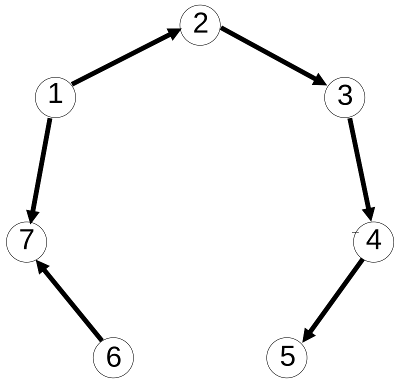


Figure 5.14. Graph for Exercise 5.4.3.

Remark 5.4.4 In general by application of transitive hull formation we have an enlargement of structural information content. There are only a few noninteresting cases without an enlargement: $G = G_{ht}$ or all edges of the transitive supplement G_t are contradictory. □

By processes of supplement formation inferences can be formally described. Processes of inference in connection with a combination of given and inferred information to an integrated cognitive structure can be formalized on the basis of hull formation processes.

5.4.4 Formalization of cognitive structure transformations with reduction of structural information content

Transformations *reducing structure* and information content play an important role in task dependent selection of information because often it is sufficient to select only a part of (externally given or internally stored or inferred) information for solving a given problem.

As an example we want to consider a problem investigated in Klix (1983) and Offenhaus (1984).

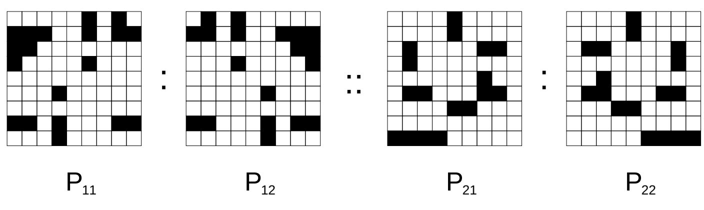


Figure 5.15. Pattern combination of the experiment by Offenhaus (1984).

Two ordered pairs (P_{11}, P_{12}) and (P_{21}, P_{22}) of geometrical patterns (consisting of squares) with substructures (formed by these squares) are used in an analogy task. In the geometrical pattern P_{11} a reflection at a certain symmetry axis is carried out such that the result of this reflection is represented by the geometrical pattern P_{12}. Subjects have to decide whether or not in the second pair (P_{21}, P_{22}) a reflection of P_{21} was carried out at the same axis as in the first pair resulting in P_{22}. To answer such a question for the correctness of an analogy in general it is not necessary to use the complete geometrical pattern. But it is sufficient to select specific subpatterns with the (important) property to be not invariant with respect to the specific reflection carried out in the first pair of patterns.

Such transformations result in a cognitive structure which represents the input information in a reduced or condensed form. Possibilities for a general or temporal reduction of structural information content are for example forgetting or nonactivating a piece of information from LTM or compressing certain features or relations to a class feature. These transformations characterize a wide field of operations of psychological relevance denoted for instance as selection, elimination, substitution, symbolization or idealization of information (cf. Klix, 1989, 1990; Mehlhorn & Mehlhorn, 1985) in combination with connection or disconnection, addition or deletion of substructures.

Beside this we have to ask for cognitive structure transformations with a reduction of structural information content but with an *enlargement of structure.* This is possible by adding contradictions to the representing structure.

A reduction of structural information content *without change of structure* is impossible.

All these types of structure transformations can be formally described by graph transformations with reduction of structural information content.

Graph transformations resulting in a reduction of structural information content on the basis of a reduction of the original graph are based on deleting certain vertices, edges or labels from the original graph, that are not in contradiction to parts of the original graph.

Graph transformations resulting in a reduction of structural information content on the basis of enlargement of the original graph are based on adding specific vertices, edges or labels to the original graph which are for example in contradiction to other parts of the original graph.

Preconditions for such graph transformations can be obtained on the basis of the elementary graph transformations described in 5.4.1. Analyses of these transformations can be carried out in an analogous way to Lemma 5.4.2. Here we want to give only a short characterization of this class of elementary graph transformations. For a more detailed formal approach cf. Sommerfeld (1991).

By such elementary transformations and their combinations (which enlarge or reduce the structure) we reduce the structural information content of the representing structure. Isolated vertices, i. e., vertices v for which do not exist an edge (u, v) or an edge (v, u) (also with nonempty label) are redundant, i. e., adding or deleting of such isolated vertices does not reduce the structural information content $I(G, Int_e)$.

In general on the one hand there must exist a noncontradictory substructure of the original graph G such that there does not exist a substructure of the generated graph G' with the same structural information content. On the other hand it is necessary that for any noncontradictory substructure of the generated graph G' there must exist a substructure of the original graph G with the same structural information content.

On the basis of such elementary graph transformations and their combinations all types of cognitive structure transformations with reduction of structural information content discussed above can be formally described.

Here we want to define combinations which are very important for formalizing and modeling processes of selection of information. Such transformations can be realized step by step by disconnecting parts of the original graph and by deleting vertices, edges or labels and are called graph coarsenings.

A simple transformation of this type is a coarsening by edge deletion. This graph transformation can be obtained by successive application of the elementary graph transformations $\varphi^-_{e,ME}$ of edge label deletion and φ^-_e of edge deletion.

Definition 5.4.5 Let $G = (V, E, f, g)$ be a graph. For a (labeled) edge $e \in E$ of G we define the graph $G^c_e =_{Def.} (V, E \setminus \{e\}, f, g')$ whereas g' is the restriction of g to $E \setminus \{e\}$.

The transformation $\varphi^c_e(G) =_{Def.} G^c_e$ is called *graph coarsening by edge deletion.* □

Lemma 5.4.5 *Let $G = (V, E, f, g)$ be a graph and $e = (u, v) \in E$. Then we have $I(G, Int_e) \supseteq I(\varphi^c_e(G), Int_e)$ and $[I(G, Int_e) \supset I(\varphi^c_e(G), Int_e)$ iff $(f(u), f(v), g(e)) \notin C$ and there exists no edge $e' = (u', v') \in E \setminus e$ with $(f(u'), f(v'), g(e')) = (f(u), f(v), g(e))]$.* □

Remark 5.4.5 In general by application of graph coarsenings by edge deletion we have a reduction of structural information content. There are only a few noninteresting cases without a reduction of structural information content: if the edge e which is deleted from G is contradictory or there exists another edge with the same structural information content. □

A certain class of graph coarsenings is based on partitions of the vertex set. These graph coarsenings can be obtained by successive application of graph coarsening by edge deletion.

Definition 5.4.6 Let $G = (V, E, f, g)$ be a graph. Let $P_V(G) = \{V_1, \ldots, V_m\}$ with $V_i \subseteq V$ $(i = 1, \ldots, m)$ be a partition of the vertex set V of G. Let $\{G\langle V_1 \rangle, \ldots, G\langle V_m \rangle\}$ be the set of subgraphs induced by the partition classes of $P_V(G)$. We define the graph $G_{PV} =_{Def.} G\langle V_1 \rangle \cup \ldots \cup G\langle V_m \rangle$.

The transformation $\varphi_{PV}(G) =_{Def.} G_{PV}$ is called *graph coarsening by vertex set partition.* □

Remark 5.4.6 This coarsening is obtained by the union of the subgraphs induced by the partition classes. It is generated from G by deleting all edges between these subgraphs. In this reduced graph there are no relations between elements from different partition classes. □

Lemma 5.4.6 *Let $G = (V, E, f, g)$ be a graph. Let $P_V(G) = \{V_1, \ldots, V_1\}$ with $V \subseteq V_i$ $(i = 1, \ldots, m)$ be a partition of the vertex set V of G. Then we have $I(G, Int_e) \supseteq I(\varphi_{PV}(G), Int_e)$ and $[I(G, Int_e) \supset I(\varphi_{PV}(G), Int_e)$ iff there exists an edge $e = (u, v) \in E$ with $u \in V_i$ and $v \in V_j$ for any $i, j \in \{1, \ldots, m\}$ and $V_i \neq V_j$ and $(f(u), f(v), g(e)) \notin C$ and there exists no edge $e' = (w, z) \in E$ with $w, z \in V_k$ for $k \in \{1, \ldots, m\}$ and $(f(w), f(z), g(e')) = (f(u), f(v), g(e))]$.* □

Remark 5.4.7 In general by application of graph coarsenings by vertex set partition we have a reduction of structural information content. There are only a few noninteresting cases without a reduction of structural information content: for any edge between the partition classes holds that it is contradictory or there exists an edge within a partition class with the same structural information content. □

5.4.5 Formalization of cognitive structure transformations with enlargement and reduction of structural information content

There exist a lot of transformations which result on the one hand in a reduction of certain information content but on the other hand in an enlargement of another specific information content.

An important case of a structure transformation which is a combination of several transformations with reduction and enlargement of information content generates a *hierarchical structure*. A hierarchical structure consists of different levels of abstraction, each level represents only a part of the whole piece of information. At an upper level only information about classes of substructures is stored. The information about each of these substructures may be obtained from a lower level. More abstract information about classes and relations between classes can be gained through inference processes. The different levels are generated by selection processes. In a lot of experimental investigations the formation of hierarchical structures has been shown (cf. for instance Pliske & Smith, 1979; Adelson, 1981; Maki, 1982; Krause, 1982; Krause et al. 1986; Pohl, 1990).

These types of cognitive structure transformations can be formally described by combinations of graph transformations with enlargement and reduction of structural information content.

Combinations of different graph transformations often can be characterized by connecting several reduced and enlarged graphs.

Here we want to select some essential graph transformations for the formal description of hierarchical structuring.

One possibility is the coarsening by identification of vertices. By application of this transformation several vertices are contracted to one vertex. This graph transformation is a combination of all types of elementary graph transformations discussed above. It can be used for the formalization of processes of compression of several features to one class feature (Klix, 1990).

Definition 5.4.7 Let $G = (V, E, f, g)$ be a graph. For the vertex set $V_0 = \{v_1, \ldots, v_k\} \subseteq V$ of G we define the graph $G =_{Def.} (V', E', f', g')$ with

$$
\begin{aligned}
V' &= V^* \cup \{V_0\} \text{ and } V^* =_{Def.} V \setminus V_0, \\
E' &= E \cap V^* \times V^* \\
&\quad \cup \{(\{V_0\}, w) | w \in V^* \text{ and exists } u \in V_0 \text{ with } (u, w) \in E\} \\
&\quad \cup \{(w, \{V_0\}) | w \in V^* \text{ and exists } u \in V_0 \text{ with } (w, u) \in E\}, \\
f'(w) &= \begin{cases} f(v_1) \cup \ldots \cup f(v_k), & \text{if } w = \{V_0\} \\ f(w), & \text{otherwise} \end{cases} \\
g'(e) &= \begin{cases} g((v_1, w)) \cup \ldots \cup g((v_k, w)), & \text{if } e = (\{V_0\}, w) \\ g((w, v_1)) \cup \ldots \cup g((w, v_k)), & \text{if } e = (w, \{V_0\}) \\ g(e), & \text{otherwise} \end{cases}
\end{aligned}
$$

The transformation $\varphi_{V0} =_{Def.} G_{V0}$ is called *graph coarsening by vertex identification.* □

Remark 5.4.8 In general on the one hand information about relations between elements of the vertex set V_0 is lost. On the other hand information about this set is obtained. □

The operation of graph coarsening by vertex identification is the basis for an essential operation in hierarchical structuring: graph coarsening by condensation. This transformation is obtained by identification of all vertices in each subgraph induced by a partition class to one vertex.

Definition 5.4.8 Let $G = (V, E, f, g)$ be a graph. Let $P_V(G) = \{V_1, \ldots, V_m\}$ with $V_i \subseteq V$ $(i = 1, \ldots, m)$ be a partition of the vertex set V of G.

The graph $G_{cond} =_{Def.} \varphi_{V1}(\varphi_{V2}(\ldots(\varphi_{Vm}(G))\ldots))$ is the *graph condensation* of G with respect to the vertex set partition $P_V(G)$.

The transformation $\varphi_{cond}(G) =_{Def.} G_{cond}$ is called *graph coarsening by condensation.* □

Remark 5.4.9 In general on the one hand information on relations between elements which are members of the same partition class is lost. On the other hand information on partition classes is obtained. □

Example 5.4.4 In Figure 5.16 graph coarsenings $\varphi_{PV}(G)$ and $\varphi_{cond}(G)$ of the graph G are shown. □

Exercise 5.4.4 For the graph G shown in Figure 5.18 and the partition $P_V(G)$ of the vertex set of G give

a) $\varphi_{PV}(G)$

b) $\varphi_{cond}(G)$. □

$G = (V, E, f, g, W_V, W_E)$
with:
$V = \{1,2,3,4,5,6,7\}$,
$E = \{(1,4),(2,1),(2,3),(2,4),(2,5),(2,6),(2,7),(3,1),(3,5),$
$(4,7),(5,6),(6,4)\}$,
$f(1) = f(4) = f(7) = \{\text{pupil, 8 years old}\}$,
$f(3) = f(5) = f(6) = \{\text{pupil, 12 years old}\}$,
$f(2) = \{\text{teacher}\}$,
$g((2,1)) = g((2,3)) = g((2,4)) = g((2,5)) = g((2,6)) =$
$g((2,7)) = g((3,1)) = g((5,1)) = g((5,4)) = g((5,7)) =$
$g((6,4)) = \{\text{helps}\} = \{\Rightarrow\}$
$g((1,4)) = g((3,5)) = g((4,7)) = g((5,6)) = \{\text{smaller}\} =$
$= \{\longrightarrow\}$
$W_V^* = \{\text{pupil, teacher, 8 years old, 12 years old}\}$,

$W_E^* = \{\text{helps, smaller}\}$.

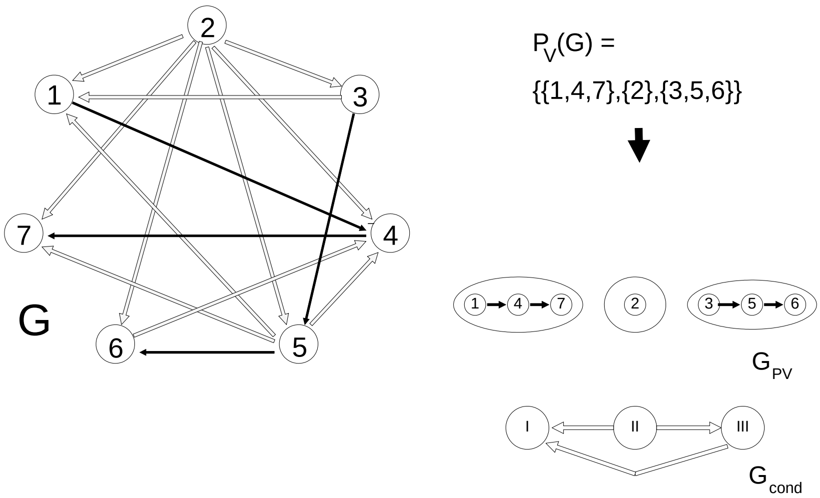


Figure 5.16. Example for graphs coarsenings by vertex set partition and by condensation.

The transformations φ_{cond} and φ_{PV} give a basis to formalize the formation of hierarchical structures.

We can describe each level of a hierarchical structure by means of a reduced graph, for instance an upper level by a graph condensation (cf. Definition 5.4.8), a lower level by another graph coarsening based on a partition of the vertex set (cf. Definition 5.4.6). A possible transformation to connect these levels is the join formation.

Definition 5.4.9 Let $G = (V, E, f, g)$ and $G' = (V', E', f', g')$ be graphs with $V \cap V' = \emptyset$. Let $\{w^*\}$ be an edge label. We define the graph $G_{join} =_{Def.}$

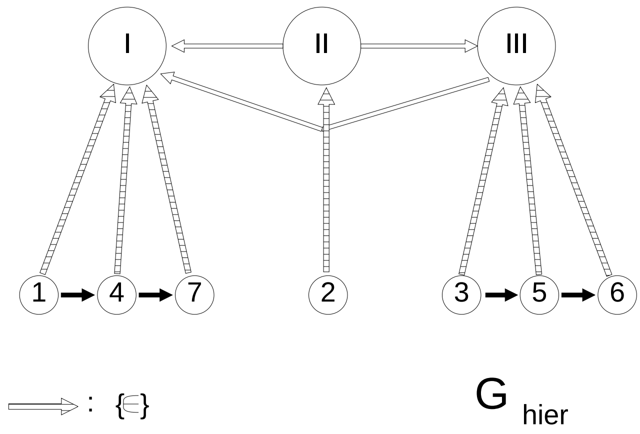


Figure 5.17. Example for a hierarchical structure G_{hier}.

(V^*, E^*, f^*, g^*) with

$$\begin{aligned} V^* &= V \cup V', \\ E^* &= E \cup E' \cup \{(v, v')|v \in V, v' \in V'\}, \\ f^*(u) &= \begin{cases} f(u), & \text{if } u \in V \\ f'(u), & \text{if } u \in V' \end{cases} \\ g^*(e) &= \begin{cases} g(e), & \text{if } e \in E \\ g'(e), & \text{if } e \in E' \\ \{w^*\}, & \text{otherwise.} \end{cases} \end{aligned}$$

G_{join} is called graph join of G and G' with respect to the label w^*. The transformation $\varphi_{join}(G, G', w^*) =_{Def.} G_{join}$ is called *graph transformation by join formation.* □

EXAMPLE 5.4.5 In Figure 5.19 the join $G_{join} = \varphi_{join}(G_1, G_2, w^*)$ of the graphs G_1 and G_2 is shown. □

EXERCISE 5.4.5 Give the join $\varphi_{join}(G,G_2, w^*)$ of the graphs G_1 and G_2 in Figure 5.20. □

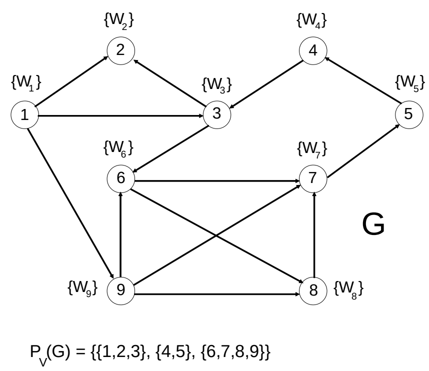


Figure 5.18. Graph G for Exercise 5.4.4.

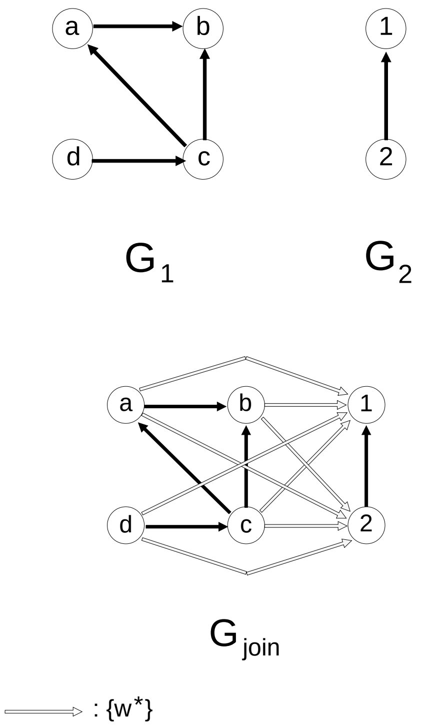


Figure 5.19. Example for a graph join.

For example a hierarchical structure of two levels can be generated on the basis of the joins between the induced subgraphs of order one of the upper level with the related induced subgraphs of the lower level. In this case the complete hierarchical structure can be obtained by the union of these joins with the graph representing the upper level. This can be extended to the formation of hierarchical structures with more than two levels by successive applications of these operations.

Example 5.4.6 In Figure 5.17 a hierarchical structure G_{hier} based on G_{PV}

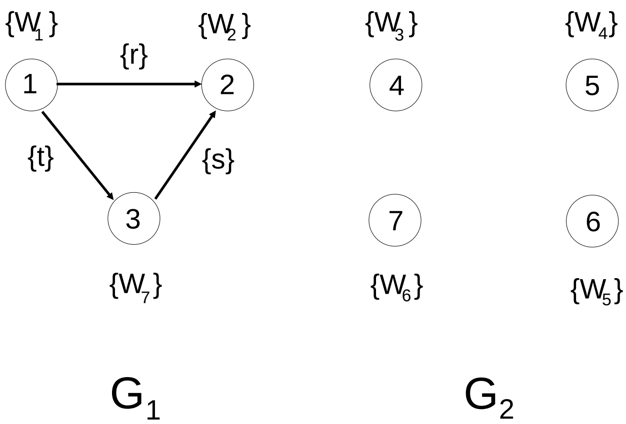


Figure 5.20. Graphs G_1 and G_2 for Exercise 5.4.5.

and G_{cond} of Figure 5.16 is shown. □

To give a contribution for an adequate formal description of cognitive structure transformations as well as for modeling and simulation of task-dependent cognitive processes, systematic formal and experimental investigations for concrete problems are necessary.

5.4.6 Application to psychological problems

The application of the approach consists in model-theoretical analyses of different problems connected with comparisons of theoretical and experimental results.

We have applied the approach to different fields in cognitive psychology. An overview about formal descriptions of different cognitive structure transformations on the basis of graph transformations is given in Sommerfeld (1991).

A suitable paradigm to analyze processes of cognitive structuring in formation and transformation of mental representations are ordering problems. It is well known that orderings of sets of elements are of importance in many domains of knowledge processing (Bower, 1971; Trabasso & Riley, 1975; Moyer & Bayer, 1976; Crowder, 1976; Banks, 1977; Groner, 1978; Pliske & Smith, 1979; Krause, 1982; Anderson, 1983; Wagener & Wender, 1985; Krause et al., 1986; Pohl, 1990).

The process of solving an ordering problem is characterized by typical cognitive structure transformations such as restructuring processes as well as deleting and adding processes within the cognitive structure that represents a certain piece of ordering information.

In relation to the definition of an one-relational ordering problem (Groner, 1978; Krause et al., 1986) a general (more-relational) ordering problem is determined by a given set V of elements, a set R of ordering relations on V, a given set $\mathcal{G}_{giv}$ of representations of propositions which give information about relationships between elements and relations, and a set Q of questions which have to be answered, i. e. a set of pairs of elements for which the question has to be answered whether the considered two elements are in a specified relation $r \in R$ or not. The information inferred on the basis of $\mathcal{G}_{giv}$ which gives the possibility of a correct answer on all these questions is a solution of the considered ordering problem.

Definition 5.4.10 Let $V = \{v_1, \ldots, v_p\}$ be a nonempty set and let $R = \{r_1, \ldots, r_q\}$ be a set of ordering (i. e. transitive, irreflexive, asymmetric) relations on V. Let $\mathcal{G}_{giv} = \{G_1, \ldots, G_n\}$ be a set of graphs such that for any $i \in \{1, ..., n\}$ and any $r \in R$ holds $G_i = (V_i, E_i, f_i, g_i)$, $V_i \subseteq V$, $f_i(v) = \{v\}$ for any $v \in V_i$, $g_i(e) \in \mathcal{P}(R)$ for any $e \in E_i$, and if $(u, v) \in E_i$ with $\{r\} \subseteq g_i((u, v))$, then $(u, v) \in r$ (i. e. u is in relation r to v).

Let $J^*_{1R} = \{(\{x\}, \{y\}, R^*)|x, y \in V, R^* \in \mathcal{P}(R)\}$ and
$J_{1R} = \{(\{x\}, \{y\}, \{r\})|x, y \in V, r \in R\}$ be sets of possible interpretations, and

let $Q = Q_T \cup Q_F \cup Q_U$ such that for any $r \in R$
$Q_T \subseteq J_{1R}$ and if $(\{x\}, \{y\}, \{r\}) \in Q_T$, then $(x, y) \in r$,
$Q_F \subseteq J_{1R}$ and if $(\{x\}, \{y\}, \{r\}) \in Q_F$, then $(y, x) \in r$,
$Q_U \subseteq J_{1R}$ and if $(\{x\}, \{y\}, \{r\}) \in Q_U$, then $(x, y) \notin r$ and $(y, x) \notin r$.
The tuple $(V, R, \mathcal{G}_{giv}, J_{1R}, Q_T, Q_F, Q_U)$ is called an *ordering problem.*

A set $S \subseteq J_{1R}$ is a *solution* of the ordering problem $(V, R, \mathcal{G}_{giv}, J_{1R}, Q_T, Q_F, Q_U)$ iff for any $(\{x\}, \{y\}, \{r\}) \in S$ holds $(\{y\}, \{x\}, \{r\}) \notin S$ and for any $x \in V$ and any $r \in R$ we have $(\{x\}, \{x\}, \{r\}) \notin S$ and $Q \subseteq S$, and if $(\{x\}, \{y\}, \{r\}) \in Q_F$ then $(\{y\}, \{x\}, \{r\}) \in S$, and for any $(\{x\}, \{y\}, \{r\}) \in Q_U$ we have $(\{x\}, \{y\}, \{r\}) \notin S$ and $(\{y\}, \{x\}, \{r\}) \notin S$.

An ordering problem $(V, R, \mathcal{G}_{giv}, J_{1R}, Q_T, Q_F, Q_U)$ is called *solvable* iff there exists a solution S of $(V, R, \mathcal{G}_{giv}, J_{1R}, Q_T, Q_F, Q_U)$. A solution S is represented by a graph $G_S = (V_S, E_S, f_S, g_S)$ (describing a cognitive structure) on the basis of an interpretation system of the type $Int_{Or} = (J^*_{1R}, C, s, t)$ iff
$S = \{(\{x\}, \{y\}, \{r\}) |$ exists $(\{x\}, \{y\}, R^*) \in I(G_S, Int_{Or})$ with $r \in R^*\}$.

An ordering problem $(V, R, \mathcal{G}_{giv}, J_{1R}, Q_T, Q_F, Q_U)$ with $R = \{r_1, \ldots, r_q\}$ is called a q–relational ordering problem.

An ordering relation r is called linear iff for any $x, y \in V$ with $x \neq y$ we have $(x, y) \in r$ or $(y, x) \in r$. The ordering problem $(V, R, \mathcal{G}_{giv}, J_{1R}, Q_T, Q_F, Q_U)$ is called linear iff all relations $r \in R$ are linear orderings.

The ordering problem $(V, R, \mathcal{G}_{giv}, J_{1R}, Q_T, Q_F, Q_U)$ is called complete iff for any $r \in R$ and any $x, y \in V$ with $x \neq y$ holds $(\{x\}, \{y\}, \{r\}) \in Q$ or $(\{y\}, \{x\}, \{r\}) \in Q$. □

REMARK 5.4.10 We have a set of given structural information units, e. g. objects, events or patterns that are in a certain ordering relation r. Based on relational connections between different information units the set of structural information units can be brought into a linear order and classified as a basis for a hierarchically ordered structure.

The general task in solving an ordering problem is to comprehend and to remember given information about ordering relations between elements and in consequence to answer questions on given and inferable information (Bower, 1971; Banks, 1977; Pliske & Smith, 1979; Krause, 1982; Pohl, 1990).

The given information consisted of a set of structural information units of the type "element - ordering relation - element $(v_i r v_k)$", ordered by chance and presented to the subjects successively at a monitor. In general only information about immediately neighboring elements is given. The subjects have to answer questions $v_l r v_m$? on neighboring elements as well as on non-neighboring elements.

That means the task in solving an ordering problem is to determine a definite set $S \subseteq V \times V \times R$ of propositions inferable from $\mathcal{G}_{giv}$, with the property that S is consistent (i. e. S represents the ordering relations under consideration), and that for any question $(\{x\}, \{y\}, \{r\}) \in Q_T$ ('x in relation r to y?') there exists an answer $(\{x\}, \{y\}, \{r\}) \in S$ ('x in relation r to y!') and for any question $(\{x\}, \{y\}, \{r\}) \in Q_F$ there exists an answer $(\{y\}, \{x\}, \{r\}) \in S$

(‘y in relation r to x!’, i. e. ‘x not in relation r to y!’ because r is an ordering relation and thus it is asymmetric) and for any question $(\{x\}, \{y\}, \{r\}) \in Q_U$ there exists no answer in S.

That means the subjects must be able to answer all these questions correctly. □

Here we want to demonstrate the application of our approach by means of an example of a linear one-relational ordering problem (cf. Pliske & Smith, 1979).

Example 5.4.7 We have the set of the following given propositions:

Prop. 1: Maria is more intelligent than Eve.

⋮

Prop. i: Anne is more intelligent than Paul.

⋮

Prop. n: Bert is more intelligent than Benny.

In this ordering problem subjects have be able to answer all questions about the relation ‘is more intelligent than’ between persons under consideration.

Thus we have $V = \{$Maria, Eve, ..., Anne, Paul,..., Bert, Benny$\}$,
$R = \{r\}$ with $r =$ ‘is more intelligent than’. We describe proposition 1 by the following graph:
$G_1 = (V_1, E_1, f_1, g_1)$ with $V_1 = \{u_1, v_1\}$, $E_1 = \{(u_1, v_1)\}$, $f_1(u_1) = \{$Maria$\}$, $f_1(v_1) = \{$Eve$\}$, and $g_1((u_1, v_1)) = \{$is more intelligent than$\}$. The other propositions can be represented in an analogous way, such that we have the set $\mathcal{G}_{giv} = \{G_1, \ldots, G_n\}$ of representing graphs.

Furthermore we have $Q = \{(\{x\}, \{y\}, \{r\}) | x, y \in V, x \neq y\}$, Q_U is empty and Q_T and Q_F are determined by the given ordering relation ‘is more intelligent than’. The ordering problem $(V, R, \mathcal{G}_{giv}, J_{1R}, Q_T, Q_F, Q_U)$ is complete.

Now it is our task to formalize different cognitive structure transformations for the solution of this ordering problem. At first we have to make a specification of the interpretation system Int_{Or}. Obviously it is sufficient to use the interpretation system $Int_{Or1} = (J_{1R}, \emptyset, s_e, t_e)$. Because the representing graph G only contains the edge label $\{r\}$ in this case the interpretation system Int_{Or1} is equivalent to the interpretation system $Int_{\{r\}} = (J_e, C, s_e, t_{\{r\}})$ with $C = \emptyset$.

On the basis of this interpretation system and the transformations characterized now we are able to analyze and formalize systematically different types of cognitive transformations which can be realized by a subject to generate an internal representation of external information about such an ordering problem. Subjects can apply different combinations of elementary transformations (to the representation of external given information) which reduce, enlarge or do not change the structural information content.

Obviously for this ordering problem there exists no solution S which does not contain the structural information represented by $\mathcal{G}_{giv}$. Thus it follows that a solution cannot be obtained by a transformation reducing the structural information content of $\mathcal{G}_{giv}$. Also $\mathcal{G}_{giv}$ represents no solution of the considered ordering problem, i. e. a solution S cannot be obtained only by transformations without change of structural information content.

To solve the ordering problem transitive inference processes are necessary. That means transformations which results in an enlargement of structural information content must be carried out.

Here we consider only transformations which give the possibility to solve the ordering problem. Thus it is sufficient to investigate a special class of combinations of elementary transformations with enlargement of structural information content described above.

The inference of information which is necessary for the solution (based on $\mathcal{G}_{giv}$ and the transitive property of the relation r) can be described by means of the process of transitive supplement formation of $G = G_1 \cup \ldots \cup G_n$, that means by $\varphi_t(G) = G_t$. In this way for example we can obtain the proposition 'Anne is more intelligent than Benny'.

The united given and inferred information is described by means of the transitive hull $\varphi_{ht}(G) = G \cup G_t = G_S = (V, E_S, f_S, g_S)$ which represents a solution S of the considered ordering problem on the basis of the interpretation system Int_{Or1}.

In the given ordering problem all girls are more intelligent than the boys. That means it is possible to use information about these classes and to solve this ordering problem on the basis of a hierarchical structure. It is easier to answer the questions on the basis of a hierarchical structure based on this classification of the set of persons than only on the given set of structural information units.

For this purpose we start with the graph G_S and the partition $P_V(G_S) = \{V_1, V_2\}$ of the vertex set V where V_1 is the set of all boys and V_2 is the set of all girls. Using these partition classes 'boys' and 'girls' at first we apply to G_S the transformation of graph coarsening by condensation and obtain $\varphi_{cond}(G_S) = \varphi_{V_2}(\varphi_{V_1}(G_S))$. This structure is used as the upper level in a two level hierarchical structure.

The structure of the lower level can be obtained by application of graph coarsening by vertex set partition on the basis of the same partition classes, i. e. we have $\varphi_{PV}(G_S) = G_S\langle V_1 \rangle \cup G_S\langle V_2 \rangle$.

The complete hierarchical structure is generated on the basis of the joins

$$\begin{aligned} \varphi_{join}(\varphi_{cond}(G_S\langle V_1 \rangle, \varphi_{PV}(G_S\langle V_1 \rangle), \{\in\}) &= G_{join1} \qquad \text{and} \\ \varphi_{join}(\varphi_{cond}(G_S\langle V_2 \rangle), \varphi_{PV}(G_S\langle V_2 \rangle), \{\in\}) &= G_{join2} \end{aligned}$$

between the induced subgraphs $\varphi_{cond}(G_S\langle V_1 \rangle)$ and $\varphi_{cond}(G_S\langle V_2 \rangle)$ of order one of the upper level with the related induced subgraphs $\varphi_{PV}(G_S\langle V_1 \rangle)$ and $\varphi_{PV}(G_S\langle V_2 \rangle)$ of the lower level, and by the union of these joins with the graph $\varphi_{cond}(G_S) = \varphi_{V_2}(\varphi_{V_1}(G_S))$ representing the upper level. By these transformations we get the graph $G_{S^*} = \varphi_{cond}(G_S) \cup G_{join1} \cup G_{join2}$.

The graph G_{S^*} does not represent a solution of the ordering problem $(V, R, \mathcal{G}_{giv}, J_{1R}, Q_T, Q_F, Q_U)$ on the basis of the interpretation system Int_{Or1},

i. e. the hierarchical organized information contained in G_{S^*} cannot be extracted by means of this interpretation system, but it is easy to choose another interpretation system which can use this hierarchical structured information.

For this purpose we have to make another specification of the interpretation system Int_{Or}. We use the interpretation system $Int_{Or2} = (J_{1R}, \emptyset, s_h, t_h)$ with $s_h(G^*) = \Big\{(u,v)|u,v \in V^*,$ exists $x,y \in V^*$ with $(u,x) \in E^*$, $(v,y) \in E^*$, $g^*((u,x)) = g^*((v,y)) = \{\in\}$, $(x,y) \in E^*$, $g^*((x,y)) = \{r\}\Big\} \cup \Big\{(u,v)|u,v \in V^*$, $(u,v) \in E^*$, $g^*((u,v)) = \{r\}\Big\}$ and $t_h((u,v)) = (f^*(u), f^*(v), \{r\})$ for any graph $G^* = (V^*, E^*, f^*, g^*)$ and for any $(u,v) \in s_h(G^*)$.

On the basis of this interpretation system Int_{Or2} using hierarchical structured information G_{S^*} represents a solution of the given ordering problem $(V, R, \mathcal{G}_{giv}, J_{1R}, Q_T, Q_F, Q_U)$. □

Remark 5.4.11 By means of this example we have shown how the solution of an ordering problem can be formalized on the basis of graph transformations. Experimental results have shown that such a formalization is adequate for cognitive processes in solving certain classes of problems. This is a basis for modeling of task-dependent cognitive processes. □

Exercise 5.4.6 There is a short story. A little hare was walking through the forest. He lost his way and asked several animals for information. This information was more or less helpful for him. The following propositions give the relationships between the answers of different animals.

The fox was more helpful than the cat.
The goose was more helpful than the craw.
The rat was more helpful than the mouse.
The sparrow was more helpful than the bee.
The mouse was more helpful than the stork.
The cat was more helpful than the rat.
The elephant was more helpful than the horse.
The craw was more helpful than the sparrow.
The stork was more helpful than the goose.
The bee was more helpful than the fly.
The horse was more helpful than the fox.

The following problem is to solve:
All questions about the relation ‘was more helpful than’ between the animals under consideration have to be answered.
1. Give two different graphs and corresponding interpretation systems such that each of these graphs represent a solution of the ordering problem described above.
2. Which cognitive structure transformations give the possibility to solve this ordering problem? Give their formal description on the basis of graph transformations characterized in this paper. □

The experimental evidence of a representative subset of the systematized cognitive structure transformations, i. e. for different types of integrative, hierarchical and selective cognitive structurings, could be produced on several experiments of solving linear ordering problems. Thereby cognitive structurings of subjects were be determined on the basis of the reaction time using the symbol distance effect (Krause et al., 1986). To generalize our approach beside linear ordering problems we also investigated partial ordering problems (Sommerfeld, 1991).

In the experiments it could be shown that relevant transformations are covered by the systematization of our approach and that these transformations can be described adequately by certain graph transformations.

Another field of application of this approach has been text comprehension and text processing. We formalized operations which have been investigated in Kintsch and van Dijk (1978) and Beyer (1985). For an overview about formalization of operations for the generation of hierarchies of propositions and the formation of macropropositions we refer to Sommerfeld (1991).

Systematization and formalization of cognitive structure transformations give a basis for a systematization and specification of hypotheses for the solution of a concrete problem under consideration. But beside the aspects of completeness and an adequate formal description of cognitive structure transformations there is the question for parameters controlling these processes, and based on this there is the question for evaluating cognitive processes with respect to their efficiency in formation and transformation of internal representations.

Based on this approach strategies can be classified and evaluated by means of pieces of structural information (relevant for solving the problem under consideration) which is used by a (theoretical or real) subject in processes of formation of internal representations. Differences between theoretical and experimental data have been determined (Sommerfeld, 1991).

5.5 Summary

The investigations are based on a central problem of cognitive psychology, to identify cognitive structures and operations on cognitive structures which are fundamental components in formation and transformation of internal representations. From this point of view one of the main problems in mathematical psychology is to provide an adequate description of internal representations and operations on these representations.

We consider the structural aspect of given and represented information. For the formalization of structures and structure transformations we use graph theory because it is very useful for describing structures, relations between structures and operations on structures.

The concept of information represented by means of a certain structure has been specified. The structural information which can be extracted from a given structure depends on the representation of this structure and on the kind of its interpretation. Therefore we introduce the concept "interpretation system" and the concept "structural information content". By means of these concepts we are able to investigate the relationships between transformations of representing structures and the change of their structural information content.

Based on the formalization of a structure representing a structural information a formal approach for an interpretation system has been developed. On the basis of this formal approach it is possible to analyze the part of an external given piece of information which is internal represented by a subject. Thereby it is possible to describe whether more or less context is taken into account and whether certain structural properties are emphasized or suppressed. Furthermore the relation between the structural information content of the internal represented information and the structural information content of the external given information can be investigated more formally.

Thus we have given prerequisites for the formal description of qualitative and quantitative differences between external given and internal represented information. Theoretical and experimental investigations are carried out on the formalization of cognitive structure transformations on the basis of graph transformations. This approach of formalization of cognitive structure transformations is a basis for an adequate formal description and evaluation of processes of formation and transformation of cognitive structures.

References

Adelson, B. (1981). Problem solving and the development of abstract categories in programming languages. *Memory and Cognition*, **9**(4), 422–433.

Aebli, H. (1981). *Denken: Das Ordnen des Tuns: Bd. II. Denkprozesse.* [Thinking: The ordering of doing. Vol. II. Processes of thinking.] Stuttgart: Klett-Cotta.

Anderson, J. R., & Bower, G. H. (1973). *Human associative memory.* Washington: Winston.

Anderson, J. R. (1976). *Language, memory and thought.* Hillsdale, NJ: Erlbaum.

Anderson, J. R. (1983). *The architecture of cognition.* Cambridge, MA: Harvard University Press.

Anderson, J. R. (1985). *Cognitive psychology and its implications* (2nd. ed.). New York: Freeman.

Anderson, J. R. (1988). *Kognitive Psychologie.* [Cognitive psychology.] Heidelberg: Spektrum der Wissenschaft.

Banks, W. P. (1977). Encoding and processing of symbolic information in comparative judgements. In G. H. Bower (Ed.), *The psychology of learning and motivation* (Vol. 11, pp. 101–160). New York: Academic Press.

Berg, M. (1991). Differentielle Gültigkeit bei Inferenzanforderungen. In W. Krause, E. Sommerfeld, & M. Zießler (Eds.), *Inferenz- und Interpretationsprozesse* (pp. 173–179). Jena: Friedrich-Schiller-Universität. Differential validity in inferential tasks. In ... Processes of inference and interpretation.]

Berg, M., & Schaarschmidt, U. (1984). Überlegungen zu neuen Wegen in der Intelligenzdiagnostik. [Approaches to new directions in diagnostics of intelligence.] *Wissenschaftliche Zeitschrift der Humboldt-Universität Berlin. Mathematisch-Naturwissenschaftliche Reihe*, **33**(6), 565–573.

Beyer, R. (1986). Psychologische Probleme der Textverarbeitung unter besonderer Berücksichtigung des Modells von Kintsch und van Dijk (1978). [Psychological problems of text processing with respect to the model of Kintsch and van Dijk(1978).] *Zeitschrift für Psychologie*, **8**(Suppl.).

Bower, G. H. (1971). Adaptation-level coding of stimuli and serial position effects. In M. H. Appley (Ed.), *Adaptation-level theory* (pp. 175–201). New York: Academic Press.

Buffart, H., & Leeuwenberg, E. (1983). Structural information theory. In H. Geißler, H. Buffart, E. Leeuwenberg, & V. Sarris (Eds.), *Modern issues in perception.* Amsterdam: North Holland. Berlin: DVW.

Collins, A. M., & Quillian, M. R. (1969). Retrieval time from semantic memory. *Journal of Verbal Learning and Verbal Behavior*, **8**, 241–248.

Collins, A. M., & Quillian, M. R. (1972). How to make a language user. In E. Tulving & W. Donaldson (Eds.), *Organization of memory.* New York: Academic Press.

Collins, A. M., & Loftus, E. F. (1975). A spreading activation theory of semantic processing. *Psychological Review*, **82**, 407–428.

Crowder, R. G. (1976). *Principles of learning and memory.* Hillsdale, NJ: Erlbaum.

Dörner, D. (1974). *Die kognitive Organisation beim Problemlösen.* Bern: Huber. Cognitive organization in problem solving.]

Dörner, D. (1976). *Problemlösen als Informationsverarbeitung.* [Problem solving as information processing.] Stuttgart: Kohlhammer.

Duncker, K. (1935). *Zur Psychologie des produktiven Denkens.* [Psychology of productive thinking.] Berlin.

Ehrlich, M.-F. (1982). An experimental study of the relationship between comprehension and memorization of a text. In J.-F. Le Ny & W. Kintsch (Eds.), *Language and comprehension* (Advances in Psychology, Vol. 9). Amsterdam: North Holland.

Falmagne, J.-C., & Doignon, J.-P. (1988). A class of stochastic procedures for the assessment of knowledge. *British Journal Mathematical Psychology*, **41**, 1–22.

Garrod, S.-C., & Sanford, A.-J. (1981). Bridging inferences and the extended domain of reference. In J. Long & A. Badley (Eds.), *Attention and Performance.* Hillsdale, NJ: Erlbaum.

Groner, R. (1978). *Hypothesen im Denkprozeß.* [Hypotheses in the process of thinking.] Bern: Huber.

Hagendorf, H. (1985). Zur Struktur ereignisbezogenen Wissens bei Kindern im Alter von 4–6 Jahren. [Structure of event-related knowledge of children in the age of 4–6 years.] *Zeitschrift für Psychologie*, **7**(Suppl.).

Harary, F. (1969). *Graph Theory.* Reading, MA: Addison-Wesley.

Johnson-Laird, P. N. (1983). *Mental models: Towards a cognitive science of language, inference, and consciousness.* Cambridge: University Press.

Keil, F. C. (1986). On the structure dependent nature of stages of cognitive development. In I. Levin (Ed.), *Stage and structure: Reopening the debate* (pp. 144–163). Norwood, NJ: Ablex.

Kintsch, W. (1974). *The representation of meaning in memory.* Hillsdale, NJ:

Erlbaum.

Kintsch, W., & van Dijk, T. A. (1978). Towards a model of text comprehension and production. *Psychological Review*, **85**, 363–394.

Kintsch, W., & Greeno, J. G. (1985). Understanding and solving word arithmetic problems. *Psychological Review*, **92**, 109–129.

Klimesch, W. (1988). *Struktur und Aktivierung des Gedächtnisses*. [Structure and activation of memory.] Bern: Huber.

Klix, F. (1971). *Information und Verhalten*. [Information and behavior.] Berlin: DVW.

Klix, F. (1980). Informationsbegriff und Informationstheorie in der Psychologie. Methodologische Betrachtungen über Grenzen und Möglichkeiten. [The concept of information and the information theory in psychology. Methodological considerations about frontiers and possibilities.] *Deutsche Zeitschrift für Philosophie*, **4**, 405–419.

Klix, F. (1983). Begabungsforschung — ein neuer Weg in der kognitiven Intelligenzdiagnostik. [Giftedness research — a new direction in the cognitive intelligence diagnostics.] *Zeitschrift für Psychologie*, **191**, 360–387.

Klix, F. (1984). Denken und Gedächtnis — über Wechselwirkungen kognitiver Kompartments bei der Erzeugung geistiger Leistungen. [Thinking and memory — about interrelations of cognitive compartments in generation of mental performance.] *Zeitschrift für Psychologie*, **192**, 213–244.

Klix, F. (1989). Kognitive Psychologie: Woher, wohin, wozu? [Cognitive psychology: Coming from, going to, why?] *Psychologie für die Praxis*, (Ergänzungsheft).

Klix, F. (1990). Wissensrepräsentation und geistige Leistungsfähigkeit im Lichte neuer Forschungsergebnisse der kognitiven Psychologie. [Knowledge representation and mental ability with respect to new results in cognitive psychology.] *Zeitschrift für Psychologie*, **198**, 165–187.

Klix, F., & Goede, K. (1968). Struktur- und Komponentenanalyse von Problemlösungsprozessen. [Analysis of structure and components of problem solving processes.] *Zeitschrift für Psychologie*, **174**.

Klix, F., & Krause, B. (1969). Zur Definition des Begriffs "Struktur", seinen Eigenschaften und Darstellungsmöglichkeiten in der Experimentalpsychologie. [Toward the definition of the term "structure", its properties and possibilities of representation in experimental psychology.] *Zeitschrift für Psychologie*, **176**, 22–54.

Kluwe, R. H., & Spada, H. (1981). Wissen und seine Veränderung: Einige psychologische Beschreibungsansätze. [Knowledge and knowledge transformation: Some psychological approaches of description.] In F. Foppa & R. Groner (Eds.), *Kognitive Strukturen und ihre Entwicklung*. Bern: Huber.

Krause, W. (1970). Untersuchungen zur Komponentenanalyse in einfachen Problemlösungsprozessen. [Investigations of the analysis of components of simple problem solving processes.] *Zeitschrift für Psychologie*, **177**, 3/4.

Krause, W. (1982). Eye fixation and three-term series problems, or: Is there evidence for task-dependent information units. In R. Groner & P. Fraisse (Eds.), *Cognition and eye movements* (pp. 122–138). Berlin: DVW.

Krause, W. (1991). Klassenzuweisung und Vergleich statt Inferenz? Ein neuer Ansatz zur Messung geistiger Leistungen. [Classifying and comparison instead of inference? A new approach of measuring mental performance.] In W. Krause, E. Sommerfeld, & M. Zießler (Eds.), *Inferenz- und Interpretationsprozesse* (pp. 106–126). Jena: Friedrich-Schiller-Universität.

Krause, W., Seifert, R., & Sommerfeld, E. (1986). Effective cognitive structures in simple problem solving. In F. Klix & H. Hagendorf (Eds.), *Human memory and cognitive capabilities: Mechanisms and performances*. Amsterdam: North Holland. 1001–1015.

Krause, W., Sommerfeld, E. (1988). Gelerntes nutzen? — über Ursachen der Änderung gelernter Wissensstrukturen bei unterschiedlichen Anforderungen im elementaren Problemlösen. [It is used what has been learned? — Toward reasons of transformation of knowledge structures in elementary problem solving.] In J. Lompscher, W. Jantos, W., & S. Schönian, (Eds.). *Psychologische Methoden der Analyse und Ausbildung der Lerntätigkeit*. Berlin: DVW.

Krause, W., Sommerfeld, E., & Zießler, M. (Eds.) (1991). *Inferenz- und Interpretationsprozesse*. [Processes of inference and interpretation.] Jena: Friedrich-Schiller-Universität.

Leeuwenberg, E. (1968). *Structural information of visual patterns*. Paris: Mouton.

Lompscher, J. (1972). Wesen und Struktur allgemeiner geistiger Fähigkeiten. In J. Lompscher (Ed.), *Theoretische und experimentelle Untersuchungen zur Entwicklung geistiger Fähigkeiten*. [Nature and structure of general mental abilities.] Berlin: DVW.

MacKay, D. M. (1950). Quantal aspects of scientific information. *Phil. Mag.*, **41**, 289–311.

Maki, R. H. (1982). Why do categorization effects occur in comparative judgement tasks? *Memory and Cognition*, **10**, 252–264.

Mandl, H., & Spada, H. (1989). *Wissenspsychologie*. [Knowledge psychology.] München: Psychologie Verlags Union.

Mandler, J. M. (1983). Structural invariants in development. In L. S. Liben (Ed.), *Piaget and the foundations of knowledge* (pp. 97–124). Hillsdale, NJ: Erlbaum.

Mehlhorn, H.-G., & Mehlhorn, G. (1985). Zur intuitiven Komponente schöpferischer Leistungsprozesse. [Toward the intuitive component of creative performance.] In J. Müller (Ed.), *Probleme schöpferischer Ingenieurarbeit* (pp. 70–94). Karl-Marx-Stadt: Technische Hochschule.

Mesarovic, M. D. (1965). Toward a formal theory of problem solving. In Sass & Wilkinson (Eds.), *Computer augmentation of human reasoning*. Washington.

Moyer, R. S., & Bayer, R. H. (1976). Mental comparisons and the symbolic distance effect. *Cognitive Psychology*, **8**, 228–246.

Nenniger, P. (1980). Anwendungsmöglichkeiten der Graphentheorie in der Erziehungswissenschaft. [Possibilities of applying graph theory in pedagogical science.] *Zeitschrift für empirische Pädagogik*, **4**, 85–106.

Offenhaus, B. (1984). *Analyse des analogen Schlußprozesses unter differentiellem Aspekt*. [analysis of analogical inferences with respect to the differential aspect.] Dissertation A. Berlin: Humboldt-Universität.

Pliske, R. M., & Smith, K. M. (1979). Semantic categorization in a linear order problem. *Memory and Cognition*, **7**, 297–302.

Pohl, R. (1990). Position effects in chunked linear orders. *Psychological Research*, **52**, 68–75.

Posner, M. I. (1976). *Kognitive Psychologie*. [Cognitive psychology.] München: Juventa.

Rickheit, G. (1991). Inferenzprozesse in der Textverarbeitung. [Inference processes in text processing.] In W. Krause, E. Sommerfeld, & M. Zießler (Eds.), *Inferenz- und Interpretationsprozesse* (pp. 17–28). Jena: Friedrich-Schiller-Universität.

Rickheit, G., & Strohner, H. (Eds.). (1985). *Inferences in text processing.* Amsterdam: North Holland.

Rumelhart, D. E., Lindsay, R., & Norman, D. A. (1972). A process model for long-term memory. In E. Tulving & W. Donaldson (Eds.), *Organization of memory* (pp. 197–246). New York: Academic Press.

Schmidt, H. D., & Sydow, H. (1981). On the development of semantic relations between nouns. In W. Deutsch (Ed.), *The child's construction of language.* London: Academic Press.

Schnotz, W. (1989). Textverstehen als Aufbau mentaler Modelle. [Text comprehension as construction of mental models.] In H. Mandl & H. Spada (Eds.), *Wissenspsychologie.* München: Psychologie Verlags Union.

Schnotz, W., Ballstaedt, S.-P., & Mandl, H. (1981). Kognitive Prozesse beim Zusammenfassen von Lehrtexten. [Cognitive processes in summarizing educational texts.] In H. Mandl (Ed.), *Zur Psychologie der Textverarbeitung* (pp. 108–167). München: Urban & Schwarzenberg.

Shannon, C. E., & Weaver, W. (1949). *Mathematische Grundlagen der Informationstheorie.* München: Oldenbourg Verlag.

Sobik, F. (1990). An approach for the comparison of cognitive structures by similarity measures on the basis of graph metrics. In M. Brzezinski & T. Marek (Eds.), *Action and performance: Models and tests. Contributions to the quantitative psychology and its methodology* (pp. 121–127). Amsterdam: Rodopi.

Sommerfeld, E. (1990). Systematization and formalization of cognitive structure transformations on the basis of graph transformations. In M. Brzezinski & T. Marek (Eds.), *Action and performance: Models and tests. Contributions to the quantitative psychology and its methodology* (pp. 105–120). Amsterdam: Rodopi.

Sommerfeld, E. (1991). *Mathematisch-psychologische Analysen zur Ausbildung und Transformation kognitiver Strukturen.* [Mathematical-psychological analysis of formation and transformation of cognitive structures.] Habilitationsschrift, Humboldt-Universität, Berlin.

Sommerfeld, E., & Sobik, F. (1986). Graph theoretical models for transformation and comparison of cognitive structures. In F. Klix & H. Hagendorf (Eds.), *Human memory and cognitive capabilities: Mechanisms and performances.* Amsterdam: North Holland.

Spada, H. (1976). *Modelle des Denkens und Lernens.* [Models of thinking and learning.] Bern: Huber.

Strohner, H. (1991). Systemtheoretische Analyse von Inferenzprozessen. [Systemtheoretical analysis of inference processes.] In W. Krause, E. Sommerfeld, & M. Zießler (Eds.), *Inferenz- und Interpretationsprozesse* (pp. 29–41). Jena: Friedrich-Schiller-Universität.

Strube, G. (1985). *Knowing what's going to happen in live: a model of biographical knowledge.* München: Max-Planck-Institut für Psych. Forschung. Paper 13/85.

Sydow, H. (1970). Zur metrischen Erfassung von subjektiven Problemzuständen und zu deren Veränderung im Denkprozeß. I. [Toward metrical measurement of subjective problem states and their transformation in human thinking.] I. *Zeitschrift für Psychologie*, **177**, 145–198.

Sydow, H. (1970). Zur metrischen Erfassung von subjektiven Problemzuständen und zu deren Veränderung im Denkprozeß. II. [Toward metrical measurement of subjective problem states and their transformation in human thinking.] II. *Zeitschrift für Psychologie*, **178**, 1–50.

Sydow, H. (1980). Mathematische Modellierung der Strukturrepräsentation und der Strukturerkennung in Denkprozessen. [Mathematical modelling of structure representation and structure recognition in processes of thinking.] *Zeitschrift für Psychologie*, **2**, 166–197.

Sydow, H. & Petzold, P. (1981). *Mathematische Psychologie*. [Mathematical psychology.] Berlin: DVW.

Trabasso, T. & Riley, C. A. (1975). The construction and use of representations involving linear order. In R. L. Solso (Ed.), *Information processing and cognition.* Hillsdale, NJ: Erlbaum.

van Dijk, T. A. & Kintsch, W. (1983). *Strategies of discourse comprehension.* New York: Academic Press.

Wagener, M. & Wender, K. F. (1985). Spatial representations and inference processes in memory for text. In H. Rickheit & H. Strohner (Eds.), *Inferences in text processing.* Amsterdam: North Holland.

6 Process knowledge in Production Systems

Cees Witteveen
Delft University of Technology, Department of Mathematics and Computer Science, Julianalaan 132, NL–2628 BL Delft, The Netherlands
E-mail: witt@cs.tudelft.nl

6.1 Introduction

As models for procedural knowledge or knowledge of what to do, Production Systems have gained some importance in both artificial intelligence and the behavioral sciences. A major advantage of these systems is their ability to offer a modular representation of procedural knowledge: each rule occurring in a Production System model can be used to represent a meaningful unit of (potential) behavior. The set of rules making up the system can be conceived as a representation of the system's behavior potential. To use Production Systems as models of *actual* behavior, however, requires some kind of *process-control* in order to put rules into a right sequence of applications constituting the behavior of the system. In this article, we will introduce a particular representation of process control in the form of a *programmed control structure.* This control structure governs the successive application of the rules in a Production System. The set of rules together with a programmed control structure will be called a *Programmed Production System*, abbreviated as PPS.

Since Production Systems may be viewed as models capable of generating and explaining behavior, we can rephrase the central problem in the behavioral sciences

> *how can we induce the structure of a system from the behavior it shows?*

in this context as:

> *given the behavior of a Production System, how can we infer its set of rules and control structure?*

We do not pretend to attack this problem in its full generality[1]. instead, we will focus on a part of this problem, namely the inference of the control structure, or *process-control knowledge*, *given* the set of rules or *procedural knowledge*, the system uses. This problem will be called the *control identification problem.*

To give the reader some necessary intuitions about the problems we will deal with, a short introduction into Production Systems will be given and we

[1]From a computational point of view there are convincing arguments for declaring this problem to be unsolvable if we insist on identification in finite time.

discuss the roles and representation of procedural and process-control knowledge within this formalism. The control identification problem will be stated informally.

In Section 6.3 we discuss some necessary concepts and notations of formal language theory.

In Section 6.4, we introduce the concept of a Programmed Production System and we show how a relation between the control structure of a PPS and its behavior can be specified. Here, we distinguish between the *internal* behavior of the system, describing it in terms of the control structure used and the *external* behavior, describing the observable aspects of the behavior. We conceive the internal behavior as giving an explanation of the external behavior of the system. As the reader might expect, in general one-to-one correspondence between the internal and external behavior of a system does not exist in the sense that, in general, the same external behavior may be explained by different internal behaviors.

Therefore, in Section 6.5, we concentrate on the class of systems capable of explaining a given external behavior and we discuss the problem of finding a most simple or *minimal* control structure explaining the external behavior of a given PPS. We show here that this problem can be solved in an efficient way. This so-called *reconstruction* result underlies the solution to the control identification problem to be discussed in Section 6.6.

In Section 6.6, we show that a method of identifying a minimal PPS generating the external behavior exists, a *finite sample* of which we can observe, thus solving the control-identification problem for Programmed Production Systems.

Finally, in Section 6.7, we will discuss some further results obtained. We give some possibilities for future research and some suggestions for further reading.

6.2 Production Systems

We do not plan to give a complete treatment of Production Systems here. On the contrary, we assume the reader to have sufficient knowledge of the use of Production Systems in cognitive psychology, cognitive science or artificial intelligence to be prepared for a more rigorous treatment of Production Systems. This part is only meant to present the necessary details of the formalism we need in the subsequent parts.

Readers, however, who want to see more details about applications of Production Systems, should consult the suggested references to the Production Systems literature collected at the end of this chapter.

6.2.1 A general description of Production Systems

A *Production System*, abbreviated by PS, consists of the following parts:

1. A finite set R of *condition-action rules*, also called *productions*, specifying what to do (actions) under the circumstances (conditions) specified. Such a rule will be conceived as describing a fragment of *potential behavior* or *procedural knowledge* of the system. A shorthand notation for a rule is: $\langle$ condition $\rangle \rightarrow \langle$ action $\rangle$.
2. A specification of the *environment* or domain of application in which the rules of the system operate. Such an environment can be described as a collection of *states*. Each such a state describes the current environment of the system by specifying the set of conditions (facts) that are said to hold at that time. In the Production Systems literature, we speak about the environment as a *database* D. Slightly abusing language, this database then can be identified with the set of states it can be in, so we conceive D as a collection of states d. We distinguish a subset $D_0 \subseteq D$ as the set of *initial* states of D.
3. An *interpreter* or *inference engine* inspecting the current state of D and the set of rules R to see which rules can be applied to change the current state into a new state of D. In the execution of the interpreter we distinguish the following subprocesses:
   - *matching* the contents of the condition part of a rule against the contents of the current state d.
   - *selection* of a rule from a set of rules that could be matched successfully. This process is also called *conflict-resolution.*
   - *execution* of the action part of a rule selected. The execution modifies d, thereby creating a new state $d' \in D$.

   Together these subprocesses constitute the so-called *recognize-act cycle* of a PS.

EXAMPLE 6.2.1 Consider the following set of general rules:

- $r1$: loves(x, y) AND not(loves(y, x)) $\rightarrow$ not(happy(x))
- $r2$: loves(x, y) AND loves(y, x) $\rightarrow$ happy(x) AND happy(y)
- $r3$: not(happy(x)) $\rightarrow$ drinking(x)

These rules should be interpreted as a kind of inference rules in folk psychology permitting one to derive conclusions given a certain state of affairs. Assume the current state of the database D, conceived as a set of states, is:

$$d_0 = \{\text{loves(john , mary), not(loves(mary , john))}\}$$

In the first recognize-act cycle, $r1$ matches with the contents of d_0, finding the substitution $\{x = \text{john}, y = \text{mary}\}$; note that $r2$ and $r3$ cannot be matched. Conflict resolution is trivial since only $r1$ can be selected and executed. Execution of this rules changes the current state and results in the state:

$$d_1 = d_0 \cup \{\text{not(happy(john))}\}$$

The next cycle offers an opportunity for $r1$ and $r3$, since both match a part of the current contents of the database. Firing $r1$ does not change the contents of the database. Firing $r3$, however, results in the state

$$d_2 = d_1 \cup \{\text{drinking(john)}\} \qquad \square$$

The description of a PS operating on a database as sketched above is fairly typical for the classical view of Production Systems. In this view, conflict-resolution is an almost fixed subtask of the interpreter, structured in a way that is independent from the knowledge domain at hand.

It can be argued, however, (Witteveen, 1984) that a separate representation of process control allows for a clearer view of the two components of a PS, one component specifying what *may* be done, the other specifying what *is being* done. Such a distinction seems to be necessary if we want to deal with, for example, the modeling of skill-acquisition. Here we can assume that basic skills can be modeled by individual production rules. By means of instruction and practice, control knowledge has to be acquired to schedule to execution of the basic skills. At the same time, such a separation offers the possibility to study differential effects of control knowledge on the performance of complex skill behavior based on the same set of basic skills.

Therefore, instead of harnessing the interpreter with a fixed control-resolution scheme, we will use an explicitly and separately represented notion of process-control knowledge in the form of a separate *control structure*.

Summarizing, in a PS we distinguish three types of knowledge representations:

- *procedural knowledge*, knowledge that can be executed, stored in the rule base of the system,
- *declarative knowledge* stored in the database as a description of the environment of the system and
- *process-control knowledge*, bearing upon the appropriate selection of procedural knowledge to process a (complex) task (Georgeff, 1983; Boyle, 1985), represented in the form of a control structure.

We propose a view of Production Systems in which the set of rules specifies what *may* be done in a task environment, delimiting the behavioral potential of the system, while the control structure is used to model what *will* be done, by selecting the rules in such a way that the right sequence of rule applications is obtained.

6.2.2 A preview of the control identification problem

Until now, a PS has been conceived as a formal object, describing declarative, procedural and process-control knowledge. To give a more dynamic characterization of a PS, we will make a distinction between the (external) *behavior* of a PS and its internal *structure*.

We want to use a PS as a model of a subject performing some task. Therefore, we will speak about the set of initial states of D as defining the task environment: each initial state $d_0 \in D_0$ can be described as a set of facts specifying some task to be performed. For example, modeling the behavior of students solving simple arithmetic problems means that we should specify a set of initial states, each initial state giving a description of the arithmetic problem to be solved. The set of initial states then specifies which tasks will be given to the student.

Describing the external behavior of a PS means that we take a look at the system from an outside point of view. This implies that processes such as the scheduling of rules are unobservable and we are only able to observe some changes in the states of D caused by the *execution* of production rules. To keep the discussion simple, we will assume that we can identify each state-change with the execution of a particular rule in a unique way. Therefore, given an initial state d_0 of some Programmed Production System (PPS) M using a control structure C, we can associate with this initial state a sequence of rules executed starting with d_0. The set of all these sequences for every initial state of D defines the set of *execution traces* of a PS for the task environment at hand.

Such a set of traces can be conceived as a description of the behavior of the system as it could be observed from the outside. Now assume that we are able only to observe a *finite part* of this (in principle) infinite set of traces and assume that we know the *set* of rules used to perform the tasks, for example as the result of a detailed protocol analysis. Then we may ask which control structure could be responsible for the external behavior a part of which we observed. This problem we call the *control identification* problem.

EXAMPLE 6.2.2 To show the relevance of the problem in another way, suppose an expert system has to be built. Suppose furthermore that a set of production rules has been chosen to encode relevant pieces of expert knowledge. After these rules have been constructed, the problem arises how to schedule their application in order to solve the tasks the system is meant for. One possibility would be to observe the behavior of an expert, identifying the rules executed and trying to find a suitable control structure generating the same execution sequences. □

To solve the control identification problem, first we have to agree upon what will count as a solution to such a problem. As we will see in the next section, given a description of the external behavior, there might be several control structures that can be used to explain such a set of traces. Intuitively, we would prefer a most simple explanation, i. e. a most simple control structure, for the set of traces observed. Therefore, we will define what a most simple control structure will look like and we will discuss an efficient method to find such a control structure.

To say it more precisely, now the control identification problem can be stated as follows:

> *Given a (finite) set of traces of a PS whose control structure is unknown and given a class of control structures, choose a (most*

efficient) control structure generating a set of traces from which the given set was a sample.

In this article we will try to solve the control identification problem for the class of programmed control structures[2]. To attack the control identification problem, we will have to formalize rigorously the intuitive notions presented in this section. Having grasped the essentially simple ideas, the reader should have no difficulties in understanding what follows.

6.3 Preliminaries and notations

In this section we will present a working vocabulary in order to deal with Production Systems and the control identification problem. This vocabulary originates from formal language theory. Formal language theory is a part of theoretical computer science investigating the relations between languages as sets of symbol sequences and abstract formalisms such as grammars and automata characterizing such languages. In this part we will only deal with the way in which formal languages are represented.

Our starting point is a particular set, called the *alphabet*. An alphabet is a finite non-empty set of symbols and will be denoted by Σ. Familiar alphabets are $\{a, b, c, \ldots x, y, z\}$ and $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$. A *word over an alphabet* Σ is a *finite*, possibly empty, sequence of symbols taken from Σ. For example, if $\Sigma = \{a, b, c\}$ then $v = abab$, $w = bcab$ and $x = aaa$ are words over Σ. In particular, the *empty word*, denoted by λ, is a word (consisting of zero symbols taken from Σ) over Σ. In the sequel we will use *sequence* and *word* indifferently. Words (sequences) can be *concatenated* to build a new word (sequence). Concatenation means gluing the two words. For words α and β the concatenation is written as $\alpha\beta$. The *length* $l(w)$ of a word w over Σ is the total number of occurrences of symbols in w. So, if $w = babab$, $l(w) = 5$. For any natural number $k \geq 0$, Σ^k denotes the set of sequences over Σ of length k. So we have $\Sigma^0 = \{\lambda\}$ and $\Sigma^1 = \Sigma$. The set of all words over Σ, denoted by Σ^*, is defined as

$$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \ldots = \bigcup_{k \geq 0} \Sigma^k$$

For example, if $\Sigma = \{a\}$, the set Σ^* equals $\{a^n \mid n \geq 0\}$, where a^n is a shorthand for a sequence of n a's. A *formal language* L over Σ is just a subset of the set Σ^*. For example, $L_1 = \{a, b, ab, ba\}$ is a (finite) language and $L = \{b^m a^n \mid n + m \geq 1\}$ is an (infinite) language over $\Sigma = \{a, b\}$. We will need the following relations between words α and β over an alphabet Σ:

- α is a *subword* of β if there exist (possibly empty) words $\beta_1, \beta_2 \in \Sigma$ such that $\beta = \beta_1 \alpha \beta_2$. If $\beta_1 = \lambda$ we say that α is a *prefix* of β.

[2]The general control identification problem has been discussed in Witteveen (1987) for arbitrary classes of control structures.

- α is a *scattered subword* of β, denoted by $\alpha \preceq \beta$, if there exist words $\alpha_1, \alpha_2, \ldots \alpha_k \in \Sigma^*$ and $\beta_1, \beta_2, \ldots \beta_k, \beta_{k+1} \in \Sigma^*$ such that
$$\beta = \beta_1\alpha_1\beta_2\alpha_2 \ldots \beta_k\alpha_k\beta_{k+1}$$
and
$$\alpha = \alpha_1\alpha_2 \ldots \alpha_k$$
If $\alpha \preceq \beta$, then we will also say that β is an *extension* of α. Note that if α is a subword of β then also $\alpha \preceq \beta$ holds.

With respect to prefixes, the following concepts and notations will be used frequently:

- The set of all (non-empty) prefixes of a word α will be denoted by $\mathrm{Pref}(\alpha)$.
- A set S of sequences over an alphabet Σ is said to be *prefix-free* iff for every word $s \in S$, no prefix s' of s, such that $s' \neq s$, occurs in S, or equivalently, for every $s \in S$, $\mathrm{Pref}(s) \cap S = \{s\}$.
- A set S of sequences over an alphabet Σ is said to be *prefix-closed* if every non-empty prefix of every sequence $s \in S$ is contained in S, or equivalently, $S = \bigcup_{s \in S} \mathrm{Pref}(s)$.

EXAMPLE 6.3.1 Let $\Sigma = \{a, b, c\}$. Let
$$\alpha = abab$$
and
$$\beta = acacbbcaba$$
Then *aba* is a subword of α, $\alpha \preceq \beta$ and $\mathrm{Pref}(\alpha) = \{a, ab, aba, abab\}$. The set $S = \{a, b, ab, ba, abb\}$ is prefix-closed. The set $S' = \{bab, abb, ca\}$ is a prefix-free set. □

A sequence α is *repetition-free* if every non-empty subword of α occurs at most once in α, i.e. if $\alpha = \gamma_1 \beta \gamma_2$ for some non-empty word β then β is neither a subword of γ_1 nor a subword of γ_2.

If $\alpha = a\,\beta$ is a word with $a \in \Sigma$ and $\beta \in \Sigma^*$ then the *tail* of α, denoted $tl(\alpha)$, is equal to β. Let S be a finite set of non-empty words (sequences). Then S is said to be *tail-consistent* if for all $s_1, s_2 \in S$, if $tl(s_1) = \beta_1\,\alpha\,\beta_2$ and $tl(s_2) = \gamma_1\,\alpha\,\gamma_2$ for some non-empty word α, then either β_2 is a prefix of γ_2 or γ_2 is a prefix of β_2. In other words, tail-consistency means that whenever the tails of two sequences have a symbol in common, and this symbol is not the last symbol in these sequences, then the symbol occurring thereafter they also have in common. For example, the set
$$S = \{abcdefg, xyde, egh, xbcd\}$$
is tail-consistent.

REMARK 6.3.1 Often sequences are identified with functions. Let $\{n\}$ denote the set of natural numbers less than or equal to n, and Σ an alphabet. Then a word α over Σ can be identified with a function $\alpha' : \{n\} \to \Sigma$, where $\alpha'(i)$ is the i–th symbol of α and n is the length of α. □

EXERCISE 6.3.1 Redefine the notions *prefix, subword, scattered subword* and *tail-consistency* using the function-representation of sequences. □

6.4 Programmed Production Systems

To capture a notion of process-control knowledge in a PS we introduce a particular type of control structure, derived from the concept of a programmed grammar (Salomaa, 1973). We call such a system M a *Programmed Production System*, abbreviated as PPS. We give the main elements of this formalism.

The rules that make up M are ordinary production rules. They are collected in a set R. We are not interested in the contents of particular rules, so we will assume that there exists some alphabet Σ_R of symbols or labels of rules, such that every rule has a label in Σ_R. Formally, we express this assumption by assuming a surjective labeling function $Lab : \Sigma_R \rightarrow R$. In the sequel we will normally assume this labeling function also to be injective. Hence, we will often identify the rules r_i in the set $R = \{r_1, r_2, \ldots r_n\}$ with their labels.

The database or task environment of the system will be denoted by a denumerable set D of database states d_i. A special subset D_0 of D marks the set of initial states. These states can be thought as describing the set of tasks or problems the system has to solve. We assume the set D to be closed under all applications of the rules in R.

The programmed control structure of a PPS is a tuple $C = (\sigma, \phi)$ where

- $\sigma : \Sigma \rightarrow \Sigma$ is a partial mapping associating with each label j of a rule chosen the label $\sigma(j)$ of the rule to be chosen next if j can be executed in the current state of the database. This mapping is called the *success* function.
- the partial mapping $\phi : \Sigma \rightarrow \Sigma$ associates with each label j of a rule chosen the label $\phi(j)$ of the rule to be chosen next if j cannot be executed in the current state of the database. We call this mapping the *failure* function.

A PPS M then is a tuple

$$M = (\Sigma_R, R, D, D_0, C)$$

where usually Σ_R will be omitted.

To describe the dynamics of a PPS M, let d_0 be an initial state of D. M is said to compute new states in D by applying a sequence of rules to d_0. To start such a computation we assume a special label *in* in Σ referring to a hypothetical starting rule, not really occurring in R. This rule *in* is used to determine the first rule to select by M for every initial state of D. It is assumed that *in* is executable in every state of D and does not change any state of D. So $\sigma(in)$ selects the first rule to try for d_0. Hereafter, M continues by applying the rule i selected to the current state d of D. If i matches the current state, M executes i, updates the current state by computing the resulting state d' and chooses $\sigma(i)$ as the next rule to apply. If i does not match d, M leaves the current state unchanged and chooses $\phi(i)$ as the next rule to apply. Such a process will stop if, having selected a rule i, M notices that i can be executed but $\sigma(i)$ is undefined or i cannot be executed and $\phi(i)$ is undefined. Without loss of generality, we can assume that M halts only if a unique label $out \in \Sigma$ is selected. Analogously to in, out is the label of a hypothetical rule used to signal a halting state for M. It is assumed that out is executable in every state of D and does not occur in the domain of σ or ϕ.

EXAMPLE 6.4.1 The following PPS M (see Table 6.1) is a system to model

Table 6.1. A Production System for simple arithmetic

label	rule		σ	ϕ	
in				1	-
1	$\ldots(X)\ldots$	$\rightarrow$	$\ldots X \ldots$	1	2
2	$\ldots X \times Y \ldots$	$\rightarrow$	$\ldots \text{product}(X,Y)\ldots$	2	4
3	$\ldots X + Y)\ldots$	$\rightarrow$	$\ldots \text{sum}(X,Y))\ldots$	3	6
4	$\ldots X + Y \ldots$	$\rightarrow$	$\ldots \text{sum}(X,Y)\ldots$	4	5
5	X	$\rightarrow$	$\text{answer}(X)$	out	3
6	$\ldots$	$\rightarrow$	fail	out	-

a student performing a simple arithmetical task (after Sleeman & Smith, 1981): Here, X and Y denote arbitrary natural numbers, $\text{product}(X,Y)$ and $\text{sum}(X,Y)$ denote the results of the corresponding computations. A pattern $\ldots N \ldots$ means that it matches N in any context. An initial state d_0 of D is an arithmetical expression, specifying an arithmetic problem to solve.

Suppose $d_0 = 4 \times (2 + 4 \times 5 + 6)$ is such an initial state of D. To see

Table 6.2. Sequence of computation states

$s_1 = (4 \times (2 + 4 \times 5 + 6), in)$,	$s_2 = (4 \times (2 + 4 \times 5 + 6), 1)$,
$s_3 = (4 \times (2 + 4 \times 5 + 6), 2)$,	$s_4 = (4 \times (2 + 20 + 6), 2)$,
$s_5 = (4 \times (2 + 20 + 6), 4)$,	$s_6 = (4 \times (22 + 6), 4)$,
$s_7 = (4 \times (28), 4)$,	$s_8 = (4 \times (28), 5)$,
$s_9 = (4 \times (28), 3)$,	$s_{10} = (4 \times (28), 6)$,
$s_{11} = (fail, out)$.	

which computation M performs, we introduce the notion of a *computation state* s being a pair consisting of a current state and a rule selected. So let us start with the initial computation state $s_0 = (d_0, in)$. Since $\sigma(in) = 1$ and in can be executed, the next computation state is $s_1 = (d_0, 1)$. Now rule 1 cannot be executed in state d_0, so M chooses $\phi(1) = 2$ and the next state is $s_2 = (d_0, 2)$. Then it turns out that rule 2 can be applied successfully, so rule 2

Figure 6.1. Control graph of M

is executed, changing the current state to $d_1 = 4 \times (2 + 20 + 6)$ and the next computation state is $s_3 = (d_1, 2)$, since $\sigma(2) = 2$, implying that rule 2 is chosen as long as it can be applied successfully to the current state of the database. The whole sequence of computation states is given in Table 6.2. This example shows that M is not able to compute the correct answer 112, since M does not apply the rules in the correct sequence. This does not imply, however, that M cannot solve any arithmetical task. For example, M can solve the problem $d_0 = 4 \times 5 + 6$ correctly. □

Notice, that by introducing *out* the control functions σ and ϕ can be assumed to be *total* functions from $\Sigma_R \cup \{in\}$ to $\Sigma_R \cup \{out\}$ by defining the value of a control function to be *out* whenever it was undefined before and giving an arbitrary value to $\phi(in)$. To simplify notations, we will deal with the rules of the system by identifying R with $\Sigma_R \cup \{\ in, out\}$.

Remark 6.4.1 Sometimes it is useful to describe the control structure of a PPS M by a labeled directed (multi-)graph, called the *control graph* G_M of M. Informally, the nodes of G_M are the elements of Σ and there are two sets of edges. The set E^σ contains the pairs (i, j) such that $j = \sigma(i)$, while E^ϕ contains the pairs (i, j) such that $j = \phi(i)$. A pair $(i, j) \in E^\sigma$ will be represented by an edge labeled σ between i and j, a pair $(i, j) \in E^\phi$ will be represented by an edge labeled ϕ between i and j. As an example, Figure 6.1 gives the control graph of the system M presented in Example 6.4.1. □

Exercise 6.4.1 Try to improve the control structure of the production system given in Example 6.4.1 in such a way that it solves the problem correctly.

Can you prove that with your control structure every simple arithmetic problem using only $+$, $\times$ and parentheses can be solved? □

6.4.1 Behavior of a PPS

In Example 6.4.1, we introduced the notion of a *computation state* as a pair $s = (d, i)$, where d is a state of D and i is a (label of a) rule in R. We will use this notion to define the behavior of M. We say that a computation state s is *initial* if $s = (d_0, in)$ for some initial state $d_0 \in D_0$ and s is called *final* if $s = (d, out)$ for some state d of D. We say that a state $s' = (d', j)$ is *directly computed* from $s = (d, i)$, denoted as $s \Rightarrow s'$, iff exactly one of the following conditions holds:

1. i matches d, the result of executing i is state d' and $\sigma(i) = j$
2. i does not match d, $d' = d$ and $\phi(i) = j$

Now we can define a *computation* of a PPS M as a sequence of computation states

$$s_0 s_1 s_2 \ldots s_t = (s_j)_{j=0}^{t}$$

such that

1. s_0 is an initial computation state of M
2. for every $1 \leq j \leq t$, $s_{j-1} \Rightarrow s_j$
3. either t is finite and s_t is a final state or t is infinite

EXAMPLE 6.4.2 Given the initial state $d_0 = 4 \times (2 + 4 \times 5 + 6)$ in Example 6.4.1, M performs a finite computation where the last computation state is (fail, out). □

The set of all possible finite initial segments of such computations $(s_j)_{j=0}^{t}$ of a PPS M will be called the *behavior* of M. It constitutes the set of all finite sequences of computation states that can be obtained by starting M on an initial state $d_0 \in D$. Given such a finite initial segment of a computation $(s_j)_{j=0}^{t}$, where $s_j = (d_j, i_j)$, note that given the initial state d_0 in s_0, every d_j can be computed by applying the sequence of rules $i_0\, i_1\, i_2 \ldots i_{j-1}$ in succession on d_0. Hence it suffices to take the initial state d_0 and such a sequence of rules to recover the computation $s_0 s_1 \ldots s_j$ completely. Therefore, we shall introduce the set Apply(M) of all application sequences of M as follows.

DEFINITION 6.4.1 Let M be a PPS. The set of *application sequences*, denoted by Apply(M), is defined as the set of all pairs (d_0, α), where

1. d_0 is an initial state of D,
2. $\alpha = i_0\, i_1 \ldots i_k$ and
3. $(d_j, i_j)_{j=0}^{k}$ is a finite initial segment of a computation of M. □

Note that Apply(M) describes the behavior of M in terms of the sequence of rules applied by success or failure, giving a complete description of the way in which these rules have been selected. Therefore we consider Apply(M) to be the description of the *internal behavior* of M. At the same time, associated with a computation of M, there exists a sequence of *changes* of database states, caused by rules that are executed. These changes are of interest to the outside observer of the system: for, only the rules *executed* can affect the environment and are associated with observable actions of the system. Therefore we will filter out all occurrences of rules *not* executed in an application sequence to construct a description of the *external behavior* of M.

DEFINITION 6.4.2 Given a PPS M and Apply(M), the set of *execution traces* of M, denoted by $Tr(M)$, is defined as:

$$Tr(M) = \{(d,\beta) \mid (d,\alpha) \in \text{Apply}(M) \text{ and } \beta \text{ is obtained from } \alpha \text{ by deleting every occurrence of a rule not executed in } \alpha\ \} \qquad \square$$

REMARK 6.4.2 Note that both Apply(M) and $Tr(M)$ are sets of sequences which are *prefix-closed.* $\square$

Observing M from the outside, one expects M to execute a next rule as long as the rule just executed is not equal to *out*. While we allow for infinite computations (we just don't know when the system will halt) we don't allow the system to enter a *failing loop* i. e. an infinite number of successive selections of rules that fail to apply. Therefore we state the following technical assumption:

ASSUMPTION 6.4.1 For every rule $i \neq out$, if $(d, \beta\, i) \in Tr(M)$ then also $(d, \beta\, i\, j) \in Tr(M)$ for some rule j. $\square$

We feel this assumption to be justified in dealing with PPS's as models for (human) behavior.

EXERCISE 6.4.2 Prove that the assumption implies that any subsequence of an application sequence consisting of rules that could not be executed, is repetition-free. $\square$

To analyze relations between M, Apply(M) and $Tr(M)$, we define the n–th extension ϕ^n of ϕ inductively by

$$\phi^0(j) = j \quad \text{for all} \quad j \in \Sigma$$

$$\phi^{n+1}(j) = \begin{cases} \phi(\phi^n(j)) & \text{if } \phi^n(j) \text{ is defined and } \phi^n(j) \neq out \\ \text{undefined} & \text{else} \end{cases}$$

Note that a trace is a pair (d, β), where β is a sequence of rules executed. Given an infinite number of possible initial states, the set $Tr(M)$ can contain an *infinite* number of different sequences, thus providing an infinite description of the external behavior. So, what did we gain? On the one hand, we have a better description of the internal and observable behavior than is provided by the control structure, since there may occur rules in C that are never applied or executed for any initial state in D. On the other hand, this description of the observable behavior may be an infinite one. Of course, in specifying this behavior we want to have a finite description of it. Therefore, we will develop a finite *code* for the behavior of a PPS. We will start with defining the execution relation Ex_M for a PPS M.

Definition 6.4.3 Given $Tr(M)$, the *execution relation* Ex_M over Σ is defined as

$$Ex_M = \{(i,j) \mid \exists \alpha \in \Sigma^*[\alpha ij \in Tr(M)]\ \} \qquad \square$$

We will use $Ex_M(i)$ to denote the set $\{j|(i,j) \in Ex_M\}$. The set Ex_M gives us a finite description of an important part of the external behavior: for every $(i,j) \in Ex_M$, in at least one trace it has been observed that j was executed immediately after i. Notice that since $Tr(M)$ is prefix-closed, $(i,j) \in Ex_M$ iff there exists a sequence β in a trace such that ij is a subsequence of β. So $Ex_M(i)$ contains all rules that could be executed immediately after i. Since every trace has been derived from an application sequence, the following propositions can be easily seen to hold:

Proposition 6.4.1 *Given Ex_M, $Tr(M)$ and Apply(M), there exists a* unique *function $\pi_M : Ex_M \to \Sigma^*$ such that for every trace*

$$(d, i_0\, i_1\, i_2 \ldots i_k) \in Tr(M)$$

the sequence

$$\pi_M(i_0,i_1)\pi_M(i_1,i_2)\ldots\pi_M(i_{k-1},i_k)$$

occurs in Apply(M). $\square$

Proof. (Sketch) For every $(i,j) \in Ex_M$, we define the sequence

$$\pi_M(i,j) = i\, i_0\, i_1 \ldots i_m$$

where $i_k = \phi^k(\sigma(i))$ for $k = 0,1,2,\ldots m$ and m is the smallest integer such that $i_m = j$. Now it is not difficult to see that for every trace (d,β), $i\,j$ is a subsequence of β iff there exists an application sequence (d,α) such that $\pi_M(i,j)$ is a subsequence of α. Defining $\pi_M(i,j)$ in another way would either violate the properties of σ and ϕ or would violate Assumption 6.4.1. ■

We will call $\pi_M(i,j)$ the *control sequence* of (i,j).

We have almost reached our goal to show that there exists a finite encoding of the behavior of a PPS. Instead of the possibly infinite set Apply(M), we will show that it suffices to use the *finite* set Ex_M together with the finite function π_M to describe this behavior. To make this claim more precise and more convincing, we show that instead of the complete function π_M we only need the values of π_M for a subset of Ex_M:

Proposition 6.4.2 *Let $Ex_M(i) = \{j_1, j_2, \ldots, j_m\}$. Then there exists a $j \in Ex_M(i)$ such that for all $j_k \in Ex_M(i)$, $\pi_M(i,j_k)$ is a prefix of $\pi_M(i,j)$.* $\square$

Proof. By Proposition 6.4.1, for every $j_k \in Ex_M(i)$, there exists a least $n_{j_k} \geq 0$ such that $j_k = \phi^{n_{j_k}}(\sigma(i))$. Let j be the rule such that $n_j = \max\{n_{j_k} \mid j_k \in Ex_M(i)\}$. Then for every j_k , $\phi^{n_j}(\sigma(i)) = \phi^{n_j - n_{j_k}}(j_k)$. Hence, if $\pi_M(i,j_k) = \alpha_{ij_k}$, this implies that $\pi_M(i,j) = \alpha_{ij_k}\beta_{j_k}$ for some sequence $\beta_{j_k} \in \Sigma^*$, proving the proposition. ■

Remember that given a sequence α, Pref(α) denotes the set of all prefixes of α. The following proposition is an almost immediate consequence of Proposition 6.4.2:

PROPOSITION 6.4.3 *Let* $Ex_M(i) = \{j_1, j_2, \ldots, j_n\}$*. Then the relation* $<_i$ *over* $Ex_M(i)$ *defined by*

$$j_k <_i j_m \quad \textit{iff} \quad \pi(i, j_k) \in Pref(\pi(i, j_m))$$

is a total ordering. □

EXERCISE 6.4.3 Prove Proposition 6.4.3. □

Remember that Ex_M was introduced as a relation over Σ. In general, the *domain* of a (binary) relation R, $dom(R)$, is the set of all elements occurring as the first element of a pair in the relation, while the *range* of R, $rng(R)$, is the set of elements occurring as the second element of a pair in the relation.

DEFINITION 6.4.4 The *control function of* M, denoted by π_M, is a mapping $\pi_M : dom(Ex_M) \rightarrow \Sigma^*$, for every $i \in dom(Ex_M)$ defined by $\pi_M(i) = \pi_M(i, j)$ where j is the unique maximal element of $<_i$. The *control path of* i is defined as the value $\pi_M(i)$ and the set of all control paths of M, denoted by Cp(M), is defined as $Cp(M) = \{\pi_M(i) \mid i \in dom(Ex_M)\}$. □

We now claim that the tuple $(R, D, D_0, Cp(M))$ can be used as an alternative description of the internal behavior of M. To prove this claim, we will show that this description can be transformed in an effective way into the description of a PPS M' generating the same set of application sequences as M does. Notice that M' does not need to be *exactly* equal to M, we only require M and M' to be indistinguishable with respect to to their sets of application sequences[3]. The first difficulty we will meet in such a *reconstruction* attempt is the fact that $Cp(M)$ is a set of sequences instead of a control structure. So we need a method to transform a set of sequences into a description of the control structure of a PPS. The following definition establishes the missing link:

DEFINITION 6.4.5 Let $S \subset \Sigma^*$ be a finite set of finite sequences. The control structure $C_S = (\sigma_S, \phi_S)$ is said to be *derived from* S iff for all sequences $i_0\, i_1\, i_2 \ldots i_m \in S$ we have:

1. $i_1 = \sigma_S(i_0)$
2. $i_{k+1} = \phi_S(i_k)$ for $k = 1, 2, \ldots, m-1$.

where σ_S and ϕ_S are partial mappings from Σ to Σ. □

Obviously, not every set of sequences can be used to derive a control structure for a PPS. It is not difficult to see that such a set S has to be *tail-consistent* to meet the requirement that σ_S and ϕ_S should be *functions.*

[3] It might be that some rules of M do not occur in any application sequence of M. Then for those rules the control structure of M' might differ from the control structure of M, without affecting the internal behavior of M'.

EXERCISE 6.4.4 Show that for every PPS M, the set $Cp(M)$ is a tail-consistent set. □

Now it can be shown that M and $M_{Cp(M)}$ have the same internal behavior:

THEOREM 6.4.1 *Let $Cp(M)$ be the set of control paths of a PPS $M = (R, D, D_0, C)$. Let $M_{Cp(M)} = (R, D, D_0, C_{Cp(M)})$, where $C_{Cp(M)}$ is the control structure derived from $Cp(M)$. Then Apply(M) = Apply($M_{Cp(M)}$).* □

We omit the proof because it is of theoretical interest only. The reader might try to prove this result by a careful application of the preceding definitions. We conclude this part by observing that $Cp(M)$ gives us a (behavioral complete) alternative for the control functions σ and ϕ in the specification of a PPS. In the following section we will use $Cp(M)$ as a representation for the control structure trying to characterize

1. the class of control structures producing the same execution-traces as M does
2. within this class of *trace-equivalent* systems a most simple or *minimal* control structure.

This will give us the necessary machinery to develop in the next section a method to *infer* such a minimal control structure from the external behavior of a PPS.

6.5 Finding a minimal control structure

Given an environment D with a set of initial states D_0 and a fixed set of rules R, we are interested in the behavior of a PPS $M = (R, D, D_0, C)$. In particular, we try to investigate control structures producing the same behavior in such an environment, using a fixed set of rules. We will characterize such a class of systems *behaviorally equivalent* to M and we will try to find a most simple system M^*, capable of producing the same external behavior as M generates, but using a control structure that at least as simple as M's control structure.

From now on, we will only compare different systems, assuming that they have a fixed underlying production-rule based system (R, D, D_0) in common, differing only in the control structure that may be used.

Systems, like M and $M_{Cp(M)}$, generating the same set of application sequences exhibit *strongly equivalent* behavior. However, given an external description of the behavior of a PPS M, we are interested in a weaker form of equivalence, only distinguishing systems if they generate different sets of external behavior i. e. different *traces*:

DEFINITION 6.5.1 Let M and M' be PPS's. Then M is called *trace-equivalent* to M', denoted as $M \equiv M'$, iff $Tr(M) = Tr(M')$. □

The following observation is immediate:

OBSERVATION 6.5.1 If $M \equiv M'$, then $Ex_M = Ex_{M'}$. Hence $dom(\pi_M) = dom(\pi_{M'})$. □

First, we will try to find a characterization of the systems M' trace-equivalent to M. Thereafter, we will study systems which are trace-equivalent and most simple or *minimal* in some sense.

DEFINITION 6.5.2 For very $\pi_M(i)$ in $Cp(M)$, the *basic-control path* of i, denoted $\pi_M^0(i)$, is the result of deleting every label j in $\pi_M(i)$ that does not occur in $Ex_M(i)$. □

Note that Definition 6.5.2 implies that $\pi_M^0(i)$ is a scattered subword of $\pi_M(i)$, i. e. $\pi_M^0(i) \preceq \pi_M(i)$.

DEFINITION 6.5.3 Let $S \subset \Sigma^*$ be a finite set of finite sequences. S is said to *approach* $Cp(M)$ iff there exists a bijective function $f : dom(Ex_M) \to S$, such that $f(i) = i\alpha$ for all $i \in dom(Ex_M)$, where $i\alpha$ satisfies

$$\pi_M^0(i) \preceq i\alpha \preceq \pi_M(i).$$ □

EXERCISE 6.5.1 Prove that the set of all basic control paths of M and the set $Cp(M)$ itself approach $Cp(M)$. Give an example to show that a set S approaching $Cp(M)$ does not need to be tail-consistent. □

The following theorem states that tail-consistency of a set approaching $Cp(M)$ is a sufficient condition for the system M_S derived from S to be trace-equivalent to M:

THEOREM 6.5.1 *If S is a tail-consistent set of sequences approaching $Cp(M)$ then $M \equiv M_S$ and $Cp(M_S) = S$.* □

For the proof of this theorem, the reader is referred to Witteveen (1987).

Given a set of systems trace-equivalent to a given system M, a natural question to ask is for *minimal* systems capable of generating $Tr(M)$. Minimality in this context can be defined in terms of efficiency of the system in generating its behavior. As a measure of efficiency we take the overall ratio of the number of rules executed versus the total number of rules applied (by failure or by success). Clearly, an important determinant of the efficiency of a system then is the *length* of the control paths $\pi_M(i)$. Hence, being more efficient can be given an operationalization in terms of being a *reduction* in the following sense:

DEFINITION 6.5.4 Let $M \equiv M'$. Then M' is said to be a *reduction of* M, denoted as $M' \sqsubseteq M$, iff for all $i \in dom(Ex_M) = dom(Ex_{M'})$ we have $\pi_M(i) \preceq \pi_{M'}(i)$. □

So we consider M' to be as least as efficient as M if $M' \sqsubseteq M$ holds. Notice that the relation $\sqsubseteq$ defines a partial order on the set of systems trace-equivalent to M. Minimal elements of this order can be considered as the most simple and efficient systems capable of generating $Tr(M)$:

Definition 6.5.5 Let $M^* \sqsubseteq M$. M^* is said to be a *minimal reduction* of M iff for all $M' \sqsubseteq M$, $M' \sqsubseteq M^*$ implies $M^* \sqsubseteq M'$. □

Now, of course, the problem arises how to construct such a minimal reduction M^* of a given PPS M. We will show that there exists a simple method to find such a minimal reduction, using the set $Cp(M)$ and the set of basic control paths of M. First, we prove a simple proposition.

Proposition 6.5.1 *For every $M' \sqsubseteq M$ and every $i \in dom(Ex_M)$, $\pi^0_M(i) \preceq \pi_{M'}(i)$.* □

Proof. Since $M' \equiv M$, $Ex_M = Ex_{M'}$. Hence, for every $i \in dom(Ex_M)$, $\pi^0_M(i) = \pi^0_{M'}(i)$. Since $M' \sqsubseteq M$, by Definition 6.5.4, $\pi^0_{M'}(i) \preceq \pi_{M'}(i)$ and the proposition follows. ■

We are now able to characterize reductions as follows

Theorem 6.5.2 *$M' \sqsubseteq M$ iff $Cp(M')$ approaches $Cp(M)$.* □

Proof. The only-if part is trivial. So we prove the if-part. Since $Cp(M')$ is tail-consistent and $Cp(M')$ approaches $Cp(M)$, we have by Theorem 6.5.1 that $M_{Cp(M')} = M'_{Cp(M')} \equiv M$. From Theorem 6.4.1 it follows that $M' \equiv M'_{Cp(M')}$. Since $\equiv$ is transitive, it follows that $M' \equiv M$. Since $Cp(M')$ approaches $Cp(M)$, we have $\pi_{M'}(i) \preceq \pi_M(i)$ for every $i \in dom(Ex_M) = dom(Ex_{M'})$. Hence, $M' \sqsubseteq M$. ■

To construct a minimal reduction, in the light of Theorem 6.5.2 and Proposition 6.5.1, the set of basic-control paths of M should be a starting point. A minimal reduction then, can be obtained by adding to each basic-control path as few rules as possible to make the set of sequences thus obtained a tail-consistent set. Theorem 6.5.2 then guarantees that a (minimal) reduction has been obtained.

We will now discuss a method to obtain such a minimal number of additions to basic-control paths. We start with stating, without proof, a technical lemma, suggesting how to extend a set of basic-control paths without loosing the possibility of obtaining a minimal reduction of M:

Lemma 6.5.1 *Let $M^* \sqsubseteq M$ be a minimal reduction of M and S a set of sequences approaching $Cp(M^*)$. If there exist sequences $i\,\alpha$, $j\,\beta \in S$ such that for some $k, l, m \in \Sigma$*

1. *$i\,k\,m \preceq i\,\alpha$ and $j\,k\,l \preceq j\,\beta$*
2. *$i\,k\,l\,m \preceq \pi_M(i)$*

then

$$i\,k\,l\,m \preceq \pi_{M^*}(i).$$ □

In plain words, this lemma tells us that if we have a current set of sequences S approaching $Cp(M^*)$ and we detect that this set is tail-inconsistent we might remove this particular occurrence of tail-inconsistency by inserting a label in one of the sequences without loosing the property that the new set of sequences S' approaches $Cp(M^*)$.

In fact, Lemma 6.5.1 allows for an iterative algorithm, computing a set of sequences S such that M_S is a minimal reduction of M:

Algorithm **MINIMAL**
begin
  Let S be the set of basic-control paths π_i^0 of M;
  while S is not tail-consistent
    do
      take two sequences $i\alpha km\beta, j\alpha' kl\beta' \in S$ with $l \neq m$;
      if $i\alpha klm\beta \preceq \pi_M(i)$ **then**
        $S := S - \{i\alpha km\beta\} \cup \{i\alpha klm\beta\}$
      else $S := S - \{j\alpha' kl\beta\} \cup \{j\alpha' kml\beta\}$
      fi
    od
end

Observing that the invariant for this algorithm is

S is approaching $Cp(M^*)$

the reader should have no difficulty in verifying the finiteness and correctness of this algorithm.

Example 6.5.1 Consider the following artificial PPS M, whose set $Cp(M)$ of control paths is:

$\pi(in) = in\,1\,2\,3\,4$	$\pi(3) = 3\,1$
$\pi(1) = 1\,2\,3\,4\,5\,1$	$\pi(4) = 4\,5\,1\,2\,3$
$\pi(2) = 2\,2\,3\,4\,5$	$\pi(5) = 5\,out$

with the following set S_0 of basic-control paths:

$\pi^0(in) = in\,1\,4$	$\pi^0(3) = 3\,1$
$\pi^0(1) = 1\,2\,4\,1$	$\pi^0(4) = 4\,5\,3$
$\pi^0(2) = 2\,2\,3\,5$	$\pi^0(5) = 5\,out$

Since S_0 is not tail-consistent (see for example $\pi^0(1)$ and $\pi^0(2)$), $\pi^0(1)$ is replaced by the sequence $1\,2\,3\,4\,1$. Comparing again this sequence with $\pi^0(2)$, we notice again a tail-inconsistency. This time $\pi^0(2)$ is replaced by the sequence $2\,2\,3\,4\,5$ and again $\pi^0(1)$ and $\pi^0(2)$ are not tail-consistent. $\pi^0(1)$ is replaced by $1\,2\,3\,4\,5\,1$. Continuing this process, the reader should not have difficulties to see that after five iterations, we obtain the following set of sequences:

$\pi'(in) = in1\,3\,4$	$\pi'(3) = 3\,1$
$\pi'(1) = 1\,2\,3\,4\,5\,1$	$\pi'(4) = 4\,5\,1\,3$
$\pi'(2) = 2\,2\,3\,4\,5$	$\pi'(5) = 5\,out$

Since this set is tail-consistent, it is the set of control paths of a minimal system. □

REMARK 6.5.1 The time-complexity of Algorithm MINIMAL is $O(n^3)$, where n is the number of rules involved. This can be improved by constructing a smarter algorithm taking into account the order in which additions to sequences can be made, reducing the complexity to $O(n^2)$ (Witteveen, 1987). □

6.6 Inferring a PPS from a finite set of traces

In the previous section we discussed a method to find a canonical representation of the control structure of a PPS with respect to the observable behavior it generates. To find, however, such a minimal reduction, we needed *complete* information about both the existing control structure and the set of traces $Tr(M)$ of a PPS M.

We will now discuss a method to reconstruct in a *finite* amount of time a PPS M if only a *finite* amount of information about its trace set $Tr(M)$ has been presented, thus solving the so-called *control-identification* problem for this class of systems.

6.6.1 Samples and failure sets

Suppose we have a PPS $M = (R, D, D_0, C)$, but

1. the control structure C is *unknown* and
2. we only have access to a finite amount of trace information, i. e. we have access to a finite part of the external behavior of M.

On the basis of this finite amount of trace information (a *sample of traces*) we want to *reconstruct* the control structure C by identifying a minimal reduction $M^* \equiv M$. We will show that there are (finite) samples of traces such that this problem can be solved. First we introduce some concepts and notations.

DEFINITION 6.6.1 A sample S_M for M is a *finite, prefix-closed* subset of $Tr(M)$. □

If M is understood we will omit the subscript $_M$ in S_M.

DEFINITION 6.6.2 Given a sample $S \subset Tr(M)$ for a PPS M, the *execution relation* Ex_S is defined as the set of ordered pairs of labels (i, j) such that ij occurs as a subsequence of some trace in S. Furthermore, $Ex_S(i)$ is defined as the set of labels following i in some trace of S: $Ex_S(i) = \{j|(i, j) \in Ex_S\}$ □

In the following we will omit the subscript $_S$ in Ex_S and $Ex_S(i)$ except when confusion would arise.

When we set about selecting a sample S from $Tr(M)$, it is clear that not every sample will qualify to obtain information about $Tr(M)$. At least we

Table 6.3. PPS for string reversal

label	rule	σ	ϕ
in	—	3	
1	$\$\$ \to \lambda$	4	2
2	$\$xy \to y\$x$	2	3
3	$\lambda \to \$$	1	
4	$\$ \to \lambda$	4	*out*

would require the sample to be complete in that it reflects *structural* properties of $Tr(M)$.

Definition 6.6.3 A sample S is called a *complete sample* if $Ex_S = Ex_M$. □

From the definition of $Tr(M)$ the following proposition can be verified immediately.

Proposition 6.6.1 *For every PPS M there exists a complete sample S_M.* □

Proof. Since Ex_M is a finite set and $Tr(M)$ is prefix-closed, we can collect a finite subset F of $Tr(M)$ such that for every $(i,j) \in Ex_M$, there exists exactly one trace $\alpha ij \in F$. Let $S = \bigcup_{\beta \in F} Pref(\beta)$. Then S is a complete sample, since $Ex_S = Ex_M$. ■

Example 6.6.1 The following PPS M is a simple production system for string reversal (Zisman, 1978), where Σ_d is some alphabet not containing the symbol \$, $D = (\Sigma_d \cup \{\$\})^*$ and the set of initial states is $D_0 = \Sigma_d^*$. The production rules have to be interpreted as rewriting rules and it is assumed that the left-most occurrence of the left-hand side of a rule in a string of D is rewritten to the right-hand side of that rule. Labels, rules and control functions are given in Table 6.3. Here x and y denote arbitrary symbols occurring in Σ_d. Remember that λ stands for the empty word with length 0. For the sake of simplicity, assume that $\Sigma_d = \{a,b,c\}$. The set Ex_M is equal to

$$\begin{aligned} Ex_M = \{ & (in,3),(1,4),(1,out),(2,2), \\ & (2,3),(3,1),(3,2),(3,3),(4,4),(4,out)\ \} \end{aligned}$$

Consider the execution trace

$$(d,\omega) = (abc, in\,3\,2\,2\,3\,2\,3\,3\,1\,4\,4\,out)$$

Let d_ω be the state of the database after execution of ω on the initial state d. The reader is invited to check that $d_\omega = cba$. Let

$$S_1 = \{(d,\alpha) \mid \alpha \in \mathrm{Pref}(\omega)\}$$

Since $Ex_{S_1} = Ex_M - \{(1, out)$, S_1 is not a complete sample. Let

$$(d', \omega') = (\lambda, in\,3\,3\,1\,out)$$

and let

$$S_2 = S_1 \cup \{(\lambda, \alpha') \mid \alpha' \in \mathrm{Pref}(\omega')\}$$

Then $Ex_{S_2} = Ex_M$, so S_2 is a complete sample. □

Note that in the preceeding sections we have discussed the possibility of *coding* the control structure of a PPS by means of a finite set of control paths. These control paths contain complete information about the control structure used. Surely, given a complete sample, we now do know which labels do occur in the *basic* control paths $\pi_M(i)$ for each $i \in dom(Ex_M)$ but we do not know in which *order* they occur. Therefore, we will try to *estimate* for every $(i, j) \in Ex_S$, the control sequence from i to j in order to synthesize the set of control paths from this information. The first notion we want to introduce is the notion of a *sample failure set.*

DEFINITION 6.6.4 Let $(d, \alpha ij)$ be a trace in S. Then the *sample failure set of* i *and* j *with respect to* $(d, \alpha ij)$, denoted by $f_S(d, \alpha ij)$, is defined by:

$$f_S(d, \alpha ij) = \{k \in \Sigma_S |\ k \text{ is not executable in state } d_{\alpha i}\ \}$$

where $\Sigma_S = rng(Ex_S)$ and $d_{\alpha i}$ denotes the state obtained after execution of αi on d. □

Notice that $f_S(d, \alpha ij)$ tells us something about the labels occurring in the control path $\pi_M(i)$ between i and j. Let m be such a label. Since m is applied by failure *whenever* i *has been executed followed by an execution of* j, such a rule m must occur in $f_S(d, \alpha ij)$. Hence, $f_S(d, \alpha ij)$ is a superset of the set of labels occurring between i and j in the control sequence $\pi_M(i, j)$. Since this line of reasoning holds for every trace $(d, \alpha ij)$, we can find an even better upperbound for this set of labels by defining the *sample failure set* of (i, j) as

DEFINITION 6.6.5 For every $(i, j) \in Ex_S$, the *sample failure set of* (i, j), denoted by $f_S(i, j)$, is defined by:

$$f_S(i, j) = \bigcap_{(d, \alpha ij) \in S} f_S(d, \alpha ij)$$

□

Analogously to the sample failure set of (i, j) we can define the (real) failure set of $(i, j) \in Ex_M$:

DEFINITION 6.6.6 For every $(i, j) \in Ex_M$, the *failure set of* (i, j) with respect to M, denoted by $\phi_M(i, j)$, is defined as the set of rules occurring after i and before j in the control path $\pi_M(i) \in Cp(M)$. □

Note that the sample failure set $f_S(i, j)$ plays the role of an *estimator* for the real value $\phi_M(i, j)$.

EXAMPLE 6.6.2 Consider Example 6.6.1 and the sample S consisting of

$$(d, \omega) = (abc, in\,3\,2\,2\,3\,2\,3\,3\,1\,4\,4\,out)$$

and all of its prefixes. Note that

$$f_S(d, in\,3\,2) = f_S(d, in\,3\,2\,2\,3\,2) = \{1\}$$

Hence,

$$f_S(3, 2) = \{1\}$$

The reader might check that

$$f_S(3, 3) = f_S(d, in\,3\,2\,2\,3\,2\,3\,3) = \{1, 2\}. \qquad \square$$

To show that sample failure sets derived from complete samples can be used as estimators for real failure sets, we have the following proposition:

PROPOSITION 6.6.2 *Let S be a complete sample for a PPS M. Then for every M' such that $M' \equiv M$ and every $(i, j) \in Ex_M$, $\phi_{M'}(i, j) \subseteq f_S(i, j)$.* $\square$

The significance of this proposition is that it allows us to estimate the real failure set of every trace-equivalent system M', even if M' is a minimal reduction of M.

PROOF. For every $M' \equiv M$, obviously, $Ex_M = Ex_{M'}$. Since S is a complete sample, for every $(i, j) \in Ex_{M'}$, $f_S(i, j)$ is defined.

Take some M' such that $M' \equiv M$ and let $k \in \phi'_M(i, j)$ and suppose k does not occur in $f_S(i, j)$. Then there is a trace $(d, \alpha ij) \in Tr(M) = Tr(M')$ such that k is executable in $d_{\alpha i}$. However, since $k \in \phi'_M(i, j)$, k occurs in the control path $\pi_M(i)$ *before* j. Therefore, if k is executable in the resulting state $d_{\alpha i}$, k would have been applied successfully and therefore k would have been executed instead of j.

So $(d, \alpha\, i\, j)$ cannot be a trace of M and a contradiction has been derived. Therefore, for every $k \in \phi'_M(i, j)$, k occurs in $f_S(d, \alpha\, i\, j)$ for all $(d, \alpha\, i\, j) \in S$ and therefore, $\phi'_M(i, j) \subseteq f_S(i, j)$. $\blacksquare$

Given a complete sample S_M, we can derive Ex_M and for every $(i, j) \in Ex_M$ an estimator $f_S(i, j)$ for $\phi_M(i, j)$. For every rule i, we know that every rule j occurring in $Ex_M(i)$ occurs in the control path of i for every trace-equivalent M'. The problem, of course, is how the rules in $Ex_M(i)$ should be *ordered* in order to build an adequate control structure for a system mimicking the external behavior of M. We will use the failure sets $f_S(i, j)$ to impose restrictions on the order in which the rules may occur in a control sequence. The resulting order will be called an *adequate ordered extension*.

DEFINITION 6.6.7 Let S_M be a complete sample. The sequence $i\, j_1\, j_2\, j_3 \ldots j_m$ is an *adequate ordered extension*, abbreviated a-o extension, of $Ex_S(i)$ if the following conditions hold:

1. $Ex_S(i) \subseteq \{j_1, j_2, \ldots, j_m\}$
2. for $k = 2, \ldots, m$, if $j_k \in Ex_S(i)$ then $\{j_1, j_2, \ldots, j_{k-1}\} \subseteq f_S(i, j_k)$

□

Such an extension is called adequate because the ordering of the rules is compatible with the sample failure set information.

It is not difficult to see that, for every $M' \equiv M$, the control path $\pi_{M'}(i)$ is an adequate ordered extension of $Ex_S(i)$ for every (complete) sample S from $Tr(M)$.

DEFINITION 6.6.8 Let S be a sample. The set $E(S)$ is said to be an S–*admissible extension* if the following conditions hold:

1. For every $i \in \Sigma_S - \{out\}$, $E(S)$ contains exactly one sequence $i\,\alpha_i$ and this sequence is an adequate ordered extension of $Ex_S(i)$.
2. $E(S)$ is tail-consistent. □

This definition immediately suggests a close correspondence between admissible extensions and control paths.

Our next theorem states that among all possible admissible extensions there exists at least one extension that can be used to reconstruct the unknown control structure C of M:

THEOREM 6.6.1 *If S is a complete sample from $Tr(M)$, there exists an S–admissible extension $E(S)$ such that $Tr(M) = Tr(M_{E(S)})$.* □

PROOF. Since S is a complete sample, for every $i \in dom(Ex_M) = dom(Ex_S)$, $\pi_M(i)$ is an adequate ordered extension of $Ex_S(i)$. Hence, since $Cp(M)$ is tail-consistent, there exists an S–admissible extension $E(S)$ such that $E(S) = Cp(M)$. By Theorem 6.4.1, M and $M_{Cp(M)}$ are strongly equivalent. Hence, they are trace-equivalent and the result follows. ■

To find a suitable control structure, Theorem 6.6.1 allows us to restrict the search space to S–admissible extensions of complete samples. This theorem, however, does not provide a *decision criterion* for deciding whether or not a given S–admissible extension can be used to build a trace-equivalent system. To show that Theorem 6.6.1 cannot be strengthened dealing only with complete samples, the following example shows that not every S–admissible extension can be used to identify a correct control structure.

EXAMPLE 6.6.3 To show that completeness alone is not sufficient to identify a control structure, consider the PPS M with a set of states $D = \{a, b\}^+$ and the rules and control functions as given in Table 6.4. For $D_0 = D$,

$$Ex_M = \{(in, 1), (in, 2), (1, out), (2, out)$$

Therefore, the set

$$S = \{(a,\ in\,1\,out), (b\ ,\ in\,2\,out)$$

is a complete sample from $Tr(M)$. Consider the following failure sets computed from S:

$$f_S(in, 1)) = \{2\} \quad , \quad f_S(in, 2)) = \{1\}$$

Table 6.4. Simple PPS for showing the insufficiency of complete samples

label	rule	σ	ϕ
in		1	*out*
1	$a \to \lambda$	*out*	2
2	$b \to a^2$	*out*	*out*

Table 6.5. Alternative PPS M'

label	rule	σ	ϕ
in		2	*out*
1	$a \to \lambda$	*out*	*out*
2	$b \to a^2$	*out*	1

Now consider the PPS M' with the same set of (initial) states, given in Table 6.5. It is not difficult to see that both M and M' are compatible with the sample S. Since M and M' cannot be distinguished on the basis of the sample given, both can be selected as a solution. However, note that for $d_0 = ab$, M generates the sequence $(in\,1\,out)$, computing the final state $d = b$, while M' generates the sequence $(in\,2\,out)$ computing the final state $d = a^3$. Hence, $Tr(M) \neq Tr(M')$ but it is impossible on the basis of S alone to decide which one is a correct solution. □

6.6.2 Context-complete samples

Although a complete sample can be used to obtain some information about the control structure responsible for generation of the observed behavior, completeness alone is too weak an assumption to guarantee correct identification of a PPS. Therefore we will formulate a stronger assumption that can be used to solve the identification problem in an effective way.

Note that completeness in a complete sample refers to the completeness of the set of rules that does occur after i in the sample set, for every executed rule i. Now there is another kind of completeness that has a bearing on the *contexts* in which a given rule j is executed immediately after the execution of rule i. What we would like to know is *when* a particular rule is selected for execution. So what we need is complete information about the *contexts* in which a given rule j is executed immediately after execution of rule i. Such a context can be given by specifying the set of rules that are executable (i. e. those rules whose condition part matches the current state of D) after i has been executed in a given state.

To formalize this notion of context-completeness of a sample we need a few definitions.

DEFINITION 6.6.9 For every trace $(d, \omega) \in Tr(M)$, the set $Pos_M(d, \omega)$ is defined as the set of rules that are *possible to execute* in the state d_ω, i. e. the set of rules matching in the state d_ω obtained from d by applying the sequence ω to d. □

Note that $Pos_M(d, \omega i)$ defines the *context* in which the rule to be actually executed next to the execution of i has to be chosen. This formulation of a context, however, is still dependent on the choice of the initial state. Since we want to formulate context-completeness as a sample property, we need a specification of context-completeness that is independent from such a particular choice of an initial state.

Fortunately, we can prove that such a context $Pos_M(d, \omega i)$ is *independent* from the *particular* initial state d chosen, in the sense that for every other pair $(d', \omega' i)$ having the same context, the same rule j will be chosen next:

PROPOSITION 6.6.3 *For all* $(d_1, \alpha i), (d_2, \beta i) \in Tr(M)$, *if*

$$Pos_M(d_1, \alpha i)) = Pos_M(d_2, \beta i)$$

then

$$(d_1, \alpha ij) \in Tr(M) \quad \textit{iff} \quad (d_2, \beta ij) \in Tr(M).$$ □

PROOF. Suppose we have $Pos_M(d_1, \alpha i)) = Pos_M(d_2, \beta i)$ for some (d_1, α_i), (d_2, β_i) in $Tr(M)$. Consider the control path $\pi_M(i) = i\, i_1\, i_2 \ldots i_n$ of i. If $i = out$ then $n = 0$, so neither $(d_1, \alpha\, ij)$ nor $(d_2, \beta\, ik)$ belongs to $Tr(M)$ for any choice of j. If $i \neq out$, $n \geq 1$ and there exist labels j and k occurring after i in $\pi_M(i)$ such that $(d_1, \alpha ij) \in Tr(M)$ and $(d_2, \beta ik) \in Tr(M)$. But then, since $Pos_M(d_2, \beta i)) = Pos_M(d_1, \alpha i)$, j and k both belong to $Pos_M(d_2, \beta i))$ and $Pos_M(d_1, \alpha i)$. However, since j is executed after i for d_1 and k after i for d_2, j should occur before k and k before j in $\pi_M(i)$. This is impossible unless $j = k$ and then the proposition follows. ■

Proposition 6.6.3 allows us to define for every PPS M a *context-selection* function Cs_M without the need to specify initial states: Let $Cs_M : \Sigma \times 2^\Sigma \to \Sigma$ be a function defined as:

$$Cs_M(i, X) = \begin{cases} j & \text{if } \exists (d, \alpha ij) \in Tr(M)\ [Pos_M(d, \alpha i) = X] \\ \text{undefined} & \text{else} \end{cases}$$

So Cs_M tells us for every pair (i, X) where i is a rule executed and X a context for some pair $(d, \alpha i)$, which rule j will be executed next given such a context. By the previous proposition, we see that Cs_M indeed is a (partial) function.

Note, in particular, that $dom(Cs_M)$ is a *finite* set of pairs. By restricting the sets Pos_M and the function Cs_M to a sample S of $Tr(M)$, the sets Pos_S and the function Cs_S can be defined analogously.

Now we call a sample to be *context-complete* if the sample context-selection function Cs_S provides as much information as the context-selection function Cs_M:

Table 6.6. Example of a context-selection function

i	X	$Cs_M(i,X)$
in	$\{1,2,out\}$	1
in	$\{1,out\}$	1
in	$\{2,out\}$	2
1	$\{1,2,out\}$	out
1	$\{2,out\}$	out
1	$\{out\}$	out
2	$\{1,2,out\}$	out
2	$\{1,out\}$	out

Table 6.7. PPS for string-reversal

label	rule	σ	ϕ
in		3	
1	$\$\$ \to \lambda$	4	2
2	$\$xy \to y\$x$	2	3
3	$\lambda \to \$$	1	
4	$\$ \to \lambda$	4	out

DEFINITION 6.6.10 Let M be a PPS and S a sample from $Tr(M)$. Then S is called a *context complete sample*, abbreviated by *c-complete sample*, iff for all $i \in \Sigma_M$ and $X \subseteq \Sigma_M$, $Cs_S(i,X)) = Cs_M(i,X)$. □

EXAMPLE 6.6.4 Consider the PPS M discussed in Example 6.6.3. The function Cs_M is given in Table 6.6. Consider the following traces: $(a, in\,1\,out)$, $(b, in\,2\,out)$, $(ab, in\,1\,out)$ and $(aab, in\,1\,out)$. The reader should have no difficulties in verifying that the sample consisting of these traces is a context-complete sample. Since for this sample S we have $f_S(in,1) = \emptyset$ and $f_S(in,2) = 1$, the alternative PPS, given in Example 6.6.3 is no longer compatible with S. □

The following observation can be easily verified:

OBSERVATION 6.6.1 For every PPS M, there exists a c-complete sample S_M. Furthermore, every c-complete sample is a complete sample. □

EXAMPLE 6.6.5 Consider the PPS for string-reversal discussed in Example 6.6.1 (see also Table 6.7). Let the sample S consist of the traces (and their prefixes)

$$(\lambda, in\,3\,3\,1\,out)$$

and

$$(abc, in\, 3\,2\,2\,3\,2\,3\,3\,1\,4\,4\, out).$$

Table 6.8 contains the arguments and values of the sample context selection

Table 6.8. Context-selection function for PPS

label i	X	$Cs_S(i,X)$
in	$\{3\}$	3
1	$\{3,4\}$	4
2	$\{2,3,4\}$	2
	$\{3,4\}$	3
3	$\{1,3,4\}$	1
	$\{2,3,4\}$	2
	$\{3,4\}$	3
4	$\{3,4\}$	4
	$\{3\}$	*out*

function Cs_S (we have ommitted the occurrence of *in* and *out* in the contexts X because they occur by definition in every context). By inspecting the control structure of M the reader can verify that for $D_0 = \{a,b,c\}^*$, S is a c-complete sample. □

Note that there exists an intimate relation between sample failure sets and sample context-selection functions:

PROPOSITION 6.6.4 *Let Σ_S denote the set (of labels) of rules occurring in S. Then*

$$f_S(i,j) = \Sigma_S - \bigcup_{\mathrm{Cs}_S(i,X)=j} X.$$

□

EXERCISE 6.6.1 Prove Proposition 6.6.4 by careful rewriting the definitions. □

We will show that a context-complete sample can be used to offer a solution to the control-identification problem. Our first result will be given without proof. (but see Witteveen, 1987).

LEMMA 6.6.1 *Let S be a context-complete sample from $Tr(M)$. If $E(S)$ is an S–admissible extension, then $M \equiv M_{E(S)}$.* □

Compare this lemma to Theorem 6.6.1. Notice that context-completeness allows us to conclude that *every* S–admissible extension is a solution to the problem of identifying M, while Theorem 6.6.1 only stated that there exists a solution among a possibly large number of admissible extensions. Now our main result follows almost immediately from Lemma 6.6.1:

THEOREM 6.6.2 *Given a context-complete sample S for an arbitrary PPS M, there exists an effective procedure to reconstruct M by a returning a minimal reduction of M.* □

PROOF. Note that using S, we can effectively determine the sets Ex_S and for every $(i,j) \in Ex_S$, the (sample) failure sets $f_S(i,j)$.

Let Σ_S be the set of labels occurring in S. Clearly, Σ_S is a finite set and, since every sequence α occurring in a tail-consistent set of sequences over Σ_S has to be repetition-free, there is a finite number of sets of tail-consistent sequences over Σ_S and each such a set can be constructed effectively. For each such a tail-consistent set E of sequences over Σ_S, it can be effectively determined whether or not E is an S–admissible extension. If so, according Lemma 6.6.1, the PPS M derived from E is trace-equivalent to M.

By the reduction results obtained we can find a minimal reduction of M_E in an effective way. ■

EXAMPLE 6.6.6 Let us try to construct a minimal PPS M' equivalent to M, given only the set of rules, the sample S and the context-selection function Cs_S given in Example 6.6.5. First we compute, for every $(i,j) \in Ex_S$, the

Table 6.9. Sample failure sets computed from sample S

Ex_S	$f_S(i,j)$
$(in, 3)$	$\{1,2,4\}$
$(1,4)$	$\{1,2\}$
$(2,2)$	$\{1\}$
$(2,3)$	$\{1,2\}$
$(3,1)$	$\{2\}$
$(3,2)$	$\{1\}$
$(3,3)$	$\{1,2\}$
$(4,4)$	$\{1,2\}$
$(4, out)$	$\{1,2,4\}$

sample failure function $f_S(i,j)$ given in Table 6.9. From this information, it is not difficult to see how the rules in $Ex_S(i)$ should be ordered in sequences to obtain an S–admissible function.

- for every a-o extension of $Ex_S(2)$, 2 should occur before 3
- for every a-o extension of $Ex_S(3)$, 1 should occur before 3
- 1 can occur before 2 and 2 can occur before 1 in an a-o extension $Ex_S(3)$.
- 4 should occur before *out* in every a-o extension of $Ex_S(4)$.

Table 6.10 contains the set of sequences that can be used to build an S–admissible set of sequences. Note that the first set of sequences S_1 is tail-consistent and therefore already is an S–admissible extension. The second set S_2 has to be adapted: extending the sequence 2 2 3 to the also adequate ordered

Table 6.10. The sets of sequences to build admissible extensions

S_1	S_2
$in\,3$	$in\,3$
1 4	1 4
2 2 3	2 2 3
3 1 2 3	3 2 1 3
4 4 out	4 4 out

Figure 6.2. Control graphs of the minimal systems inferred. For explanation, see text

extension 2 2 1 3 of $Ex_S(2)$ makes the set tail-consistent. Then this set is also an S–admissible extension. According to Theorem 6.6.2, both M_{S_1} and M_{S_2} are trace-equivalent to M. Note that both systems are minimal reductions. The control graphs of both systems are given in Figure 6.2. □

REMARK 6.6.1 Both a complete and a context-complete sample require in the worst case $O(n^2)$ different sample traces having no prefix in common. Assuming that the sample information is given in the form of a sample context function, we are able to show that a PPS can be reconstructed in a time polynomially bounded by the number of rules if there exists an M whose control graph does not contain E^{ϕ}–cycles. The general problem, however, is likely to be NP-hard and for this case we use a backtracking algorithm. □

Exercise 6.6.2 Construct an algorithm to find an S–admissible extension given a sample S. Your algorithm should construct adequate extensions in a clever way combining them to admissible extensions. Estimate the time-complexity of your algorithm. □

Exercise 6.6.3 Consider the rule-based system presented in Example 6.4.1 and let the following states be the initial states:
$d_1 = 2+4\times5+3$, $d_2 = 4\times(2+4\times5+6)$, $d_3 = 1+(4\times(2+4\times5+2)+3)$ and $d_4 = (1+4\times(5+6\times7+3)\times5+1)$.
The following sample, where all prefixes of traces have been omitted, can be assumed to be context-complete:

$$\begin{aligned} S = \{ \quad & (d_1, 2\,4\,4\,5), (d_2, 3\,2\,3\,1\,2\,5), \\ & (d_3, 3\,2\,3\,1\,2\,3\,1\,4\,5), (d_4, 3\,3\,2\,3\,1\,2\,2\,3\,1\,5)\,\} \end{aligned}$$

Find a minimal control structure explaining this set of traces. □

6.7 Discussion

6.7.1 Applications

Elsewhere (Witteveen, 1984) we have given some applications of the identification of programmed control structures. These applications ranged from subjects as serial pattern recognition, the analysis of role programs and modeling student's problem solving behavior to the simplification of programmed grammars in the literature. In general, we concluded that, using the PPS model, we could infer more efficient and simpler control structures than the one tradionally used. Furthermore, we could more clearly separate in protocol analysis the stages of rule induction and behavior simulation by first concentrating on the set of rules to be induced and then using the identification method to organize the rules in a proper way to explain the behavior sequences.

6.7.2 Further research

Generalizations of PPS Of course, the PPS model is a very simple model. In our experiments we found some situations in which the model was not appropriate. But several generalizations can be conceived: First we can extend the domain and range of the control functions σ and ϕ to Σ^* instead of Σ, allowing for formalizations of strategies. It is also possible make the selection of a next rule dependent on *predicates* occurring in the rules instead of the labels of the rules directly. In general we have to experiment with different (admissible) control structures, motivating them on subject-dependent theoretical grounds and analyzing their properties mathematically. Other aspects such as the *compilation of control knowledge* in the rules themselves, such as is observed in skill-learning, could be modeled by processes as rule composition induced by a control structure.

Comparing control constructs In our approach we hope to develop a formal theory of behavioral expressiveness. Note that we considered a set of rules as a description of potential behavior. Given such a rule base, a class of control structures defines a set of subsets of set of all possible behaviors. Now we can compare the expressiveness of different types of control constructs with respect to an arbitrary rule base. For example, a Markov type of control construct (Waterman & Hayes-Roth, 1978) defines a total priority order on the set of production rules such that always the rule highest in priority will be chosen for execution. It is simple to show that given an arbitrary rule base, for every Markov control structure there is a trace-equivalent PPS control structure but not the other way around. Therefore, the PPS control construct is more expressive than the Markov control construct.

Concurrent control Another line of further research could be the study of concurrent control. A promising framework for studying concurrent processes is Petri-Net theory. Representing a control structure by such a Petri net would allow for modelling behavior of distributed systems, where concurrency is more rule than exception.

6.7.3 Suggestions for further reading

Production Systems Hayes-Roth (1985) is an introduction to Production Systems dealing with the formalism, its applications and relations to other approaches in expert-systems technology. Hayes-Roth, Waterman, and Lenat (1983) and Buchanan and Shortliffe (1984) contain a more comprehensive treatments of applications of rule-based systems. Klahr, Langley, and Neches (1987) contains a number of applications in the behavioral sciences. With respect to representation of control, we refer to Davis (1980), Georgeff (1982, 1983) and Clancey (1983). Although a little out of date, Waterman and Hayes-Roth (1978) still contains some valuable contributions.

Formal languages and theory of computation In this chapter we have only touched upon some concepts. Hopcroft and Ullman (1980), or Salomaa (1973) are excellent introductions. Also Bobrow and Arbib (1974) contains some good chapters on formal language theory, automata and computability. Matters of complexity are dealt with in the classic Aho, Hopcroft, and Ullman (1974).

References

Aho, A. V., Hopcroft, J. E., & Ullman, J. D. (1974). *The design and analysis of computer algorithms*. Reading, MA: Addison-Wesley.

Bobrow, L. S., & Arbib, M. A. (1974). *Discrete mathematics*. Washington, DC: Hemisphere Publishing Corporation.

Boyle, C. D. B. (1985). Acquisition of control and domain knowledge by watching in a blackboard environment. In M. Merry (Ed.), *Expert Systems 85* (British

Computer Society Workshop Series, pp. 273–286). London: Cambridge University Press.

Buchanan, B. G., & Shortliffe, E. H. (1984). *Rule-based expert systems: The MYCIN experiments of the Stanford Heuristic Programming Project.* Reading, MA: Addison-Wesley.

Clancey, W. J. (1983). *The advantages of abstract Control Knowledge in expert system design* (STAN–CS–83–995). Stanford, CA: Department of Computer Science.

Davis, R. (1980). Meta-rules: Reasoning about control. *Artificial Intelligence*, **15**, 179–222.

Georgeff, M. P. (1982). Procedural control in production systems. *Artificial Intelligence*, **18**, 175–201.

Georgeff, M. P. (1983). Strategies in heuristic search. *Artificial Intelligence*, **20**, 393–425.

Hayes-Roth, F. (1985). Rule-based systems. *Communications of the ACM*, **28**(9), 921–934.

Hayes-Roth, F., Waterman, D. A., & Lenat, D. B. (1983). *Building expert systems.* Reading, MA: Addison-Wesley.

Hopcroft, J. E., & Ullman, J. D. (1980). *Introduction to automata theory, languages and computation.* Reading, MA: Addison-Wesley.

Klahr, D., Langley, P., & Neches, R. T. (1987). *Production system models of learning and development.* Cambridge: MIT Press.

Salomaa, A. (1973). *Formal languages.* New York: Academic Press.

Sleeman, D. H., & Smith, M. J. (1981). Modelling student's problem solving. *Artificial Intelligence*, **16**, 171–188.

Waterman, D. A., & Hayes-Roth, F. (1978). An overview of pattern directed inference systems. In D. A. Waterman & F. Hayes-Roth (Eds.), *Pattern directed inference systems* (pp. 3–24). New York: Academic Press.

Witteveen, C. (1984). *Programmed production systems.* Utrecht: Thesis University of Utrecht.

Witteveen, C. (1987). *Identification of control structures* (TWI Report 87–66). Delft: Department of Mathematics and Computer science, Delft University of Technology.

Zisman, M. D. (1978). Use of production systems for modelling asynchronous, concurrent processes. In D. A. Waterman & F. Hayes-Roth (Eds.), *Pattern directed inference systems* (pp. 53–68). New York: Academic Press.

7 Phenomena of self-organization

Hermann Rodenhausen[1]

Universität zu Köln

Seminar für Mathematik und ihre Didaktik, Gronewaldstr. 2, D-50931 Köln

7.1 Introduction

In recent years, the theory of neural nets provided new tools to explain cognitive phenomena of various kinds. Applications have been given to fields such as learning and thinking, problem solving, language comprehension, perception and pattern recognition. As an example of modeling cognitive phenomena on the basis of neural nets in this paper we will present a formal approach to phenomena of self-organization. Self-organizing systems are structures that react in an adaptive way to signals in the environment. In this way, they are able to build up "meaningful" ordering states that in a sense represent the structure of the environment. The name "self-organization process" refers to the fact that the adaptation process takes place without "supervision" simply as a result of the external signals and the systems internal activity.

We give here a brief introduction to the rather complex subject of self-organization processes. We report on some observations and computer simulations that have been done in this area. We then concentrate on the question how these phenomena can be described and understood in a formal mathematical approach.

We will see from our discussion how mathematical methods can be the natural technique in formalizing certain interrelationships of a complex nature as they occur in cognitive sciences and how highly non-trivial mathematical problems may arise from such a description. We will see that, apart from numerical analyses, mathematics can provide substantial insights and be an indispensable tool in understanding natural phenomena on a qualitative level.

We assume some basic knowledge and acquaintance with mathematical notation in linear algebra, real analysis, and probability theory. For background the reader is referred to Lang (1971), Rosenlicht (1968), and Lamperti (1966), respectively.

---

[1]This paper was written while the author held a research position in the research project "'Werten und Wissen"' (Grant Al 205/4 to D. Albert) at the Psychological Institute of Heidelberg University. The financial support by the Deutsche Forschungsgemeinschaft is gratefully acknowledged.

7.2 Empirical observations and computer simulations

It has been observed by neurophysiologists that in some parts of the brain, especially in the cerebral cortex, specific tasks can be assigned to specific locations in the brain. For example, there are areas that are "responsible" for the analysis of sensory signals (classified according to their modality, i. e., visual, auditory, somatosensory, etc.) or for tasks of motor control, etc. (Kohonen, 1988). The neural interconnections establishing these correspondences have revealed a certain fine-structure: they preserve topological order. This means the following. Let us consider, for example, the neural connections between cells of the retina and those locations in the cortex they are transmitted to. Then the order structure is this: neighboring cells on the retina are mapped onto neighboring cells in the cortex.

It has been argued (Kohonen, 1988; Buhmann et al., 1987) that this order principle cannot be explained purely genetically. Instead, it is believed that self-organizing processes are responsible for this ordering effect; we will see later how models of such processes can indeed explain the generation of topology-preserving mappings under the influence of statistically structured sensory inputs.

The above ordering phenomenon has been investigated by computer simulations (Buhmann et al., 1987; Ritter et al., 1992). We will briefly report on this approach because it represents an important step towards a formal representation of these effects.

Buhmann et al. (1987) present an abstract model system in which "neurons" are mapped onto elements of a space of "sensory cells". The system is exposed to a series of external from signals the space of sensory cells. Under the influence of these external signals, alteration of the system parameters occurs. Under suitable conditions, the system settles down in a final state. In that state, the system of neurons in a sense represents the structure of the "environment" given by the external inputs. The cortical area in the brain, i. e., the set of neurons, is modeled by the square

$$N = \{(i, j) \in \mathrm{N}^2 \mid 1 \leq i, j \leq l\},$$

l being a fixed number which determines the size of the set N. The set of sensory cells is modeled by the set

$$S = \{(x, y) \in \mathrm{R}^2 \mid 0 \leq x \leq 1, 0 \leq y \leq 1\}.$$

It is assumed that each neuron is connected to a point in the sensory square S by means of a nerve fiber. This connection is modeled by a mapping from N into S, each "neuron" (i, j) is associated with a certain point $m_{i,j} \in S$. It is assumed that this mapping is time-dependent: the associations $(i, j) \to m_{i,j}$ are changed under the influence of a sequence of "external signals" from S. At the beginning of the procedure, the assignment $(i, j) \to m_{i,j}$ is random; see figure 7.1. The first picture represents the set S at the beginning. Points $m_{i,j}$ and $m_{i',j'}$, that correspond to neighboring neurons (i, j) and (i', j') are

1 2 3

Figure 7.1. Example of a self-ordering process (from Buhmann et al., 1987); see text

connected by a line.

In the subsequent adaptation process, the values $m_{i,j}$ are stepwise adjusted as a result of randomly chosen "sensory inputs" $(x, y) \in S$. The procedure is as follows. Each time a sensory input (x, y) occurs, the neuron (i_0, j_0) is chosen which is "most responsible" for this location in S, i. e., the neuron (i_0, j_0) such that the distance between (x, y) and m_{i_0,j_0} is minimal. For all neurons (i, j) in a neighborhood of (i_0, j_0) including (i_0, j_0), the connections $m_{i,j}$ are then shifted a little towards (x, y). The amount of the shift of $m_{i,j}$ decreases as a function of the distance of (i, j) and (i_0, j_0) and the iteration time t. In detail, the shift of $m_{i,j}$ at iteration time t is given by

$$m_{i,j} \to m_{i,j} + H[(i - i_0)^2 + (j - j_0)^2, t] \cdot ((x, y) - m_{i,j}) \tag{7.1}$$

where

$$\begin{aligned} H[D, t] &= H_0 \cdot \exp(-a_0 t - D b^{-2}(t)) 9 D > 0, \\ b(t) &= b_0 \cdot e^{-c_0 t}, \end{aligned} \tag{7.2}$$

with H_0, a_0, c_0 positive. The function H determines the size of the corrections and the size of the affected neighborhood of (i_0, j_0). From equations (7.1) and (7.2) one sees that the amount of the corrections decays exponentially as a function of time (at a rate determined by a_0) and as a function of the distance of (i, j) and (i_0, j_0). The size of the affected neighborhood of (i_0, j_0) shrinks with time which is expressed by the term $b^{-2}(t)$ in (7.2). H_0 determines the size of the corrections at the beginning.

This procedure is applied repeatedly, each time a 'sensory input" (x, y) is chosen randomly. Pictures 2 and 3 in fig. 7.1 show how the system develops. The assignment $(i, j) \to m_{i,j}$ at the beginning is arbitrary; neurons may correspond to quite different locations in the set of sensory inputs S. Locations in S that are assigned to neighboring neurons are connected by a line.

Figure 7.1 shows that the system becomes more and more ordered; in the

Figure 7.2. Phoneme maps (from Kohonen, 1988)

final state (picture 3) one can recognize the following two effects:

(i) neighboring neurons are mapped to neighboring points in the set of sensory inputs;

(ii) the neural connections $m_{i,j}$ are distributed homogeneously in the set of sensory inputs.

It has been shown (Ritter & Schulten, 1986) that there is a more general law behind the effect (ii): the final distribution of the neural connections $m_{i,j}$ in a sense approximates the distribution according to which the sensory inputs are randomly chosen. Areas in which sensory inputs occur frequently are covered rather densely by neural connections; neural connections disappear in areas where only few excitations occur. From a biological and also from a technological point of view this principle seems very economical.

To illustrate the significance of this kind of self-ordering process, we briefly describe two further simulation experiments that have been done by Kohonen (1988).

The first experiment was done in connection with experiments on automatic speech recognition. In the experiment, phonemes of the Finnish language were presented to a self-organizing system as input vectors (repeatedly, in random order). The phonemes were described by their frequency spectra, taken at 15 different frequency channels (so the phonemes were represented as vectors in a 15–dimensional space).

The final state is shown in figure 7.2; the first picture (a) shows which "neuron" each phoneme became "most sensitive" to; the second picture (b) shows which phoneme each "neuron" became most sensitive to (the points in the plane corresponding to neurons in both cases).

This experiment shows that the self-organizing system may be used to display similarities between the members of an "abstract" pattern space in a metric way by means of a lower-dimensional representation.

The second example shows a system which is able to build up virtual images of its environment. The mechanism is shown in fig. 7.3. The two arms of a robot simultaneously touch points in a plane in random order. For each such point there are corresponding pairs of bending angles (ξ_1, ξ_2) and (ξ_3, ξ_4) for the two joints of each arm. If the data (ξ_1, ξ_2) or (ξ_3, ξ_4) are led to a self-organizing system in a suitable way as input signals, a self-ordering process takes place resulting in an "internal representation" of the plane; each "neuron" becomes most sensitive to one particular point in the plane (represented by a pair of angles). Each of the two arms in this way gives rise to a virtual image of the plane, i. e., a mapping from the set of neurons into the "real" plane. In this experiment done by Kohonen (1988) the two images created by the two arms coincided very well.

7.3 Formalization of the self-organization process

We will now clarify the structure of the experiments described above by looking at them again from a mathematical point of view. They all have a common over-all structure which we are now going to describe on a more formal level. We here basically follow the approach given by the fundamental work of Kohonen (1988).

In each of the above examples, there is a set of patterns or signals, whose members occur in a random order as inputs to the system. In order to build up a mathematical model, let us denote this pattern space by S, its members by x, y, etc. Next, we have a set of processing units, called neurons, which we denote by N. In each of the experiments, there is an initial state which is given by a (randomly determined) mapping $N \to S$. We further have a random sequence $x_1, x_2, \ldots$ of inputs chosen from S. In mathematical terms, this is suitably expressed by saying that $x_1, x_2, \ldots$ is a sequence of random variables with values in S. Their probability distribution is characterized by some probability measure P on S. The input signals $x_1, x_2, \ldots$ chosen at random give rise to an adaptation process which is described by a random mapping m from (N into S^n) where n is the number of neurons, i. e., the cardinality of N. The nat-

Figure 7.3. Feeler mechanism (from Kohonen, 1988)

ural numbers $\mathbf{N}$ play the role of time. For time $k \in \mathbf{N}$ and $i \leq n$, $m_i(k)$ denotes the (random) image in S of the i–th neuron, given an enumeration $\nu_1, \ldots, \nu_n$ of the set of neurons N. We usually suppress the explicit dependence of $m(k)$ upon chance in the notation.

The initial state of the adaptation process is given by $m(0)$. The mechanism by which the adaptation takes place as function of the input signals $x_1, x_2, \ldots$ will be specified below. Under suitable conditions, the system will converge to some final state $m(\infty)$, i. e., $m(k) \to m(\infty)$ as $k \to \infty$. This final state has, under appropriate conditions, two important properties: It is topology-preserving (see below) and it approximates the (stationary) distribution of the input signals (see below).

The mathematical problem arising here is to give a formal explanation for these phenomena. As the problems concerned are highly substantial, we will here restrict ourselves to a simple model system and show how some of the properties mentioned above can be understood in a rigorous mathematical way. In order to make the mathematical problem precise, we now specify further assumptions.

Suppose S is a one-dimensional set, typically, the interval

(7.3) $$S = \{x \in \mathbf{R} \mid 0 \leq x \leq 1\}.$$

Suppose further that N is some initial segment of the set of natural numbers, say

(7.4) $$N = \{i \in \mathbf{N} \mid 1 \leq i \leq n\}$$

for some $n \in \mathbf{N}$. We assume that the random variables x_k are identically distributed and pairwise independent. That is, each of the $x_k's$ has the same probability distribution and the outcome of x_k is stochastically independent of that of x_l for $k \neq l$. We assume that the common distribution of the input signals x_k is the uniform probability distribution on the interval $[0, 1]$, i. e., for each subinterval $[c, d]$ of $[0, 1]$ and for each k

$$P(x_k \in [c, d]) = d - c.$$

For each $i \in N$ let U_i be the neighborhood of i consisting of the neurons

(7.5) $$\{i - 1, i, i + 1\} \cap N$$

i. e., all "nearest neighbors" of i that are contained in N, including i itself. (Alternative definitions of this system of neighborhoods are possible; it has been shown especially by computer simulations that it may be useful to start with neighborhoods that are fairly large at the beginning and let them shrink with time; cf. Buhmann et al. (1987), Ritter & Schulten (1989), Ritter et al. (1992)). The adaptation mechanism is then defined by induction on k as follows. Suppose $m(k)$ has been defined. The best match for the signal x_k occurring at time k is the neuron i satisfying

$$|m_i(k) - x_k| = \min\{|m_j(k) - x_k| \mid j \in N\};$$

here $m_j(k)$ denotes the value of $m(k)$ at neuron j. The value of $m(k + 1)$ is then given by

$$(7.6) \qquad m_j(k+1) = \begin{cases} m_j(k) & \text{if } j \notin U_i \\ m_j(k) + \alpha(k)(x_k - m_j(k)) & \text{otherwise} \end{cases}$$

where $(\alpha(k))_{k \in \mathrm{N}}$ is a sequence of positive parameters converging to 0 as $k \to \infty$. So the new value of m at time $k+1$ results from $m(k)$ by shifting $m_j(k)$ for those neurons j which are neighbors of the “best match” for x_k a little towards the signal x_k; $m_j(k)$ remains unchanged for all other neurons. Given any initial configuration $m(0) \in S^n$ a value $m(k)$ is defined for each $k \in \mathrm{N}$ by the recursive definition (7.6). Since the $x_k's$ depend on chance, so does $m(k)$.

Equation (7.6) corresponds to equation (7.1) on page 225. The difference between the two equations is that the sets of neurons and external signals are two-dimensional (so m is indexed by two parameters) in case of (7.1) and one-dimensional in case of (7.6). In addition, the neighborhood condition in (7.6) is encoded in the continuous function H in case of (7.1).

The significance of applying the sequence $\alpha(k)$ in (7.6) lies in the following. Since the system is expected to stabilize, $\alpha(k)$ must converge to 0 as $k \to \infty$ because otherwise the corrections of $m(k)$ that are done according to (7.6) keep the system changing all the time. On the other hand, $\alpha(k)$ should not converge to 0 too rapidly because then the system might be “frozen” in a state that is not ordered yet. Computer simulations and analytical considerations of Ritter and Schulten (1989) suggest that a choice for $\alpha(k)$ is suitable if it satisfies

(i) $\quad \alpha(k) \to 0$ as $k \to \infty$

but so slowly that

(ii) $\quad \sum_{k=1}^{\infty} \alpha(k) = \infty.$

Exercise 7.3.1 A point is selected at random (uniform distribution) from the square

$$(7.7) \qquad A = \{(x, y) \in \mathrm{R}^2 \mid 0 \le x \le 1,\ 0 \le y \le 1\}.$$

How big is the probability that $x \le y$? Are the random variables x, y on the probability space A independent? What about these questions if A is replaced by the disc

$$(7.8) \qquad A' = \{(x, y) \in \mathrm{R}^2 \mid x^2 + y^2 \le 1\}? \qquad \square$$

Exercise 7.3.2 Recall that two events A and B are independent if $P(A \cap B) = P(A) \cdot P(B)$ (P denoting the probability function). Show the following:

(i) Two events A and B are independent if and only if A and $\bar{B}$ (= set-theoretic complement of B) are independent.

(ii) If A, B, and C are independent, then $A \cup B$ and C are independent.

(iii) Give an example of a probability space Ω and events A, B, $C \subset \Omega$ such that A, B, C are not independent but A, B, C are pairwise independent.

$\square$

EXERCISE 7.3.3 Try to generalize the dynamics (7.6) to higher dimensions, for example, dimension 2 (instead of 1). In this case, work with a 2–dimensional finite set of neurons, in analogy with the 1–dimensional case (7.4). Also the set S given by (7.3) has to be replaced by a corresponding 2–dimensional set. How can neighborhoods U_i for neurons i in analogy with (7.5) be defined reasonably? □

EXERCISE 7.3.4 Explain how the entities occurring in figures 1–3 are related to the abstract model quantities N, S, m, etc. introduced above. □

7.4 Mathematical formulation of the ordering property

The ordering effects can now be formulated more precisely by the following statements.
(i) *Ordering.* There exist two states $m^{(1)} : N \to S$ and $m^{(2)} : N \to S$ such that, if $\alpha(k) \to 0$ sufficiently slowly as $k \to \infty$, the following holds: for each initial state $m(0)$, $m(k)$ converges with probability 1 to either $m^{(1)}$ or $m^{(2)}$ as $k \to \infty$. $m^{(1)}$ is a monotonously increasing function, i. e., $m_i^{(1)} < m_j^{(1)}$ if $i < j \in N$. $m^{(2)}$ is the reversed vector in the sense that $m_i^{(2)} = m_{n-i+1}^{(1)}$ for $i = 1, \ldots, n$.

REMARK 7.4.1 $\alpha(k)$ converging to 0 sufficiently slow as $k \to \infty$ in the above statement means the following:

(1) there exists a sequence $\alpha(k)$ satisfying the claim of the statement (i)
(2) for any sequence $\alpha'(k)$ converging more slowly than $\alpha(k)$ to 0 as $k \to \infty$, i. e., for any sequence $\alpha'(k)$ converging to 0 and satisfying $\alpha'(k) \geq \alpha(k)$ ($k \in \mathbb{N}$), the claim of the statement (i) also holds. □

In the following statement, the number n of neurons is not kept fixed any more. It says something about the asymptotic states of the self-organizing process in the limit $n \to \infty$.
(ii) *Asymptotic distribution.* Consider for each $n \in \mathbb{N}$ the system with n neurons according to (7.6). The corresponding asymptotic states according to statement (i) above, $m^{(1)}$ and $m^{(2)}$, respectively, approximate the uniform probability distribution on $S = [0, 1]$ in the following sense. If $\delta_{m^{(1)}}$ denotes the Dirac measure associated to $m^{(1)}$, that is, the measure having point mass 1 at each of the n values $m_i^{(1)}$ $i \leq n$ and λ denotes the uniform probability distribution on $S = [0, 1]$, then

$$\frac{1}{n}\delta_{m^{(1)}} \to \lambda \text{ as } n \to \infty$$

in the sense of weak convergence of probability measures (Lamperti, 1966):

$$\int_S F d(\frac{1}{l}\delta_{m^{(1)}}) \to \int_S F d\lambda \qquad \text{as } l \to \infty \tag{7.9}$$

for each bounded continuous function $F : S \to \mathbb{R}$.

Intuitively, (7.9) states that the measure $\frac{1}{n}\delta_{m^{(1)}}$ approximates λ in the sense that integrals with respect to these measures converge.

Statement (i) tells us that there are two ordering states which may be achieved as asymptotic states: the m_i finally become ordered either in an ascending or a descending chain. This ordering property can be generalized to higher dimensions (Kohonen, 1988; Ritter & Schulten, 1989; Ritter et al., 1992; also compare diagrams 1–3).

Statement (ii) can also be formulated in a more general setting. In particular, for any stationary probability distribution of the input vectors x_k, not just the uniform one, it would state that the asymptotic states have a limit distribution (Ritter & Schulten, 1986).

The above statements will not be proved in this paper. They are well supported by computer simulations (Kohonen, 1988). For a related model, the corresponding properties have been proved by Ritter & Schulten (1986; 1989).

We will in this paper restrict ourselves to a simpler property of the model. We will show that, under a suitable choice of the sequence $\alpha(k)$, the system becomes ordered with probability 1. That means, that with probability 1 the vector $m(k)$ will be monotonous (i. e., either monotonously increasing or monotonously decreasing) for large enough k. This is the content of the subsequent theorem 7.4.1. We then show that, for large times, the process $m(k)$ is approximated by a simpler deterministic process. This result is given by theorem 7.4.2.

In order to prove the ordering property stated in theorem 7.4.1, we need some preparation which is given by the following proposition. This proposition presents a key idea in the proof of theorem 7.4.1: For each state of the system, i. e., for each $m \in S^n$, there exists a sequence of signals from S that turns m into an ordered state. So for each state, there is a chance that it will become ordered within a finite number of steps. The number of steps necessary depends on α.

PROPOSITION 7.4.1 *Consider the process* $m(k)$ *defined by* (7.6) *with* $\alpha(k) = \alpha$ *fixed. For each* $m \in S^n$ *and each* $\alpha > 0$, *there is a sequence of points* $\xi_1, \ldots, \xi_r$ *from* S *such that* $m(r)$ *defined by* (7.6) *with* $m(0) = m$ *and* $x_l = \xi_l$ $(l \leq r)$ *is monotonous. Moreover, the number* r *of time steps needed to pass into a monotonous state can be chosen to depend only on* α, *not on* m. $\square$

The proof of this proposition is left as an exercise to the reader (exercise 7.4.2).

THEOREM 7.4.1 *If, as* $k \to \infty$, $\alpha(k) \to 0$ *sufficiently slowly, then for all initial states there exists, with probability* 1, *a* k_0 *such that* $m(k_0)$ *is monotonous.* $\square$

PROOF. The idea of the proof is the following. For each state of the system, there is a positive probability that the state will become ordered within a finite number of time steps. This is shown by proposition 7.4.1.

Consequently, the process $m(k)$ can be compared with a Markov process with the two states 'ordered' and 'not ordered'. Once the process has reached the ordered state, it remains there for all later times; at each time there is a non-zero probability to pass from the state 'non ordered' to the state 'ordered'.

In detail, one proceeds as follows. As a consequence of proposition 7.4.1, the probability that the process $m(k)$ becomes monotonous within r time steps can be estimated by a non-zero probability p uniformly for all initial configurations $m \in S^n$, r and p depend on α which we express in notation by writing $r(\alpha)$ and $p(\alpha)$, respectively. Now define the following Markov process Δ.

Recall that a sequence of random variables $(X(s))_{s\in\mathbb{N}}$ with values in a set of states J is called a *Markov process* if for all $s \in \mathbb{N}$ and for any sequence of states $j_1, \ldots, j_s, j_{s+1} \in J$

$$\begin{aligned} &Prob(X(s+1) = j_{s+1} \,|\, x(1) = j_1, \ldots, x(s) = j_s) = \\ &Prob(X(s+1) = j_{s+1} \,|\, x(s) = j_s) \end{aligned} \tag{7.10}$$

(Prob denoting the probability law associated to the sequence $(X(s))_{s\in\mathbb{N}}$). (7.10) means, loosely speaking, that for all times s, the probabilities for passing into the next state at time $s+1$ are completely determined by the present state of the process and do not depend on the previous history of the process.

In order to define the Markov process Δ, let $\alpha_1, \alpha_2, \ldots$ be a sequence of numbers from the interval $(0, 1)$. Let $\{0, 1\}$ be the state space of the process (standing for the states 'ordered' and 'not ordered', respectively) and define the time-dependent dynamics

$$\begin{aligned} Prob(\Delta(s+1) = 1|\Delta(s) = 1) &= 1 \\ Prob(\Delta(s+1) = 1|\Delta(s) = 0) &= p(\alpha_s) \qquad (s \in \mathbb{N}), \end{aligned} \tag{7.11}$$

Prob refers to the probability associated to the Markov process Δ, $Prob(X|Y)$ is the conditional probability of getting X given Y. The process $\Delta(s)$ can be compared with the process $m(k)$ in the following way. If $\alpha(k)$ is chosen according to the following rule:

$$\begin{aligned} &\alpha(k) = \alpha_1 \quad \text{for} \quad 0 \le k < r(\alpha_1) \\ &\alpha(k) = \alpha_2 \quad \text{for} \quad r(\alpha_1) \le k < r(\alpha_1) + r(\alpha_2) \\ &\alpha(k) = \alpha_3 \quad \text{for} \quad r(\alpha_1) + r(\alpha_2) \le k < r(\alpha_1) + r(\alpha_2) + r(\alpha_3) \end{aligned}$$

and so on, i. e., if $\alpha(k)$ remains constant on successive time intervals of lengths $r(\alpha_i)$, then

$$P(m(k) \text{ will finally become monotonous}) \ge Prob(\Delta(s) \text{ will finally be } 1)$$

(here p denotes the probability associated to the process m). For the latter probability we have the explicit formula

$$Prob(\Delta(s) \text{ will finally be } 1) = 1 - \prod_{s=0}^{\infty}(1 - p(\alpha_s)) \tag{7.12}$$

by the Markov property of the process Δ (see exercise 7.4.3). Letting $\alpha_s \to 0$ sufficiently slowly as $s \to 0$, the product in 7.12 will be 0, so the probability in question will be 1. ■

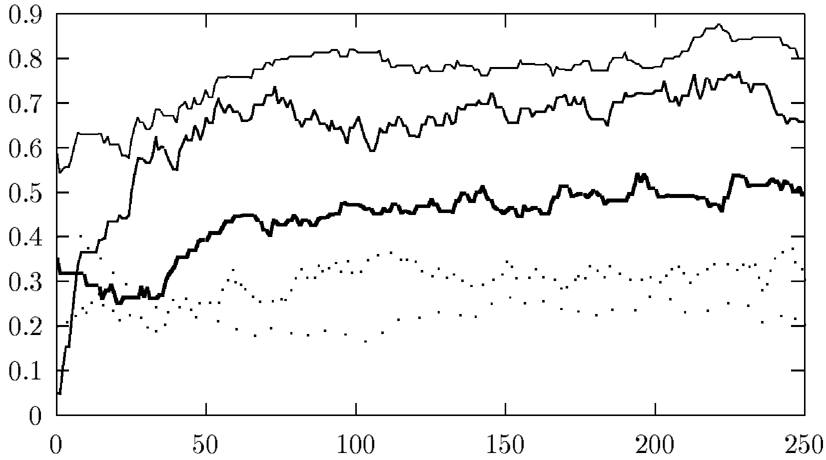


Figure 7.4. Self-ordering process with $\alpha = 0.1$, 250 time steps

While theorem 7.4.1 tells us that ordering will occur at some time with probability 1, the following proposition states that the ordering is preserved for all later times. Theorem 7.4.1 and proposition 7.4.2 combined show that, with probability 1, the system will be finally ordered, i. e., it will be ordered from some time on.

PROPOSITION 7.4.2 *Suppose that for some* $k_0 \in \mathrm{N}$ $m(k_0)$ *is monotonous. Then, with probability* 1, $m(k_0)$ *is monotonous for all* $k \geq k_0$. □

The proof of this proposition is easy and is left as an exercise (exercise 7.4.4).

We finish this section by describing the dynamics of the system in the limit of a large time scale and $\varepsilon \to 0$. We show, that, as a consequence of the law of large numbers, for large times the dynamics is approximated by a deterministic dynamics, i. e., a dynamics that does not depend on chance.

To explain that, consider figures 7.4–7.6. Figures 7.4–7.6 show the development of the process $m(k)$ given by (7.6) under three different conditions. In the first case, $\alpha(k)$ is chosen to be 0.1 constantly, in the second case 0.05, and in the third case 0.01. In all cases, $n = 5$, i. e., there are 5 neurons. The values of $m_i(k)$ $(i = 1, \dots, 5)$ as functions of time are depicted in each of the diagrams, so in each diagram, there are 5 curves. The initial configuration in the space $[0, 1]$ is in all three cases chosen to be $(0.4, 0.2, 0.35, 0.5, 0.6)$. In figure 7.4, the process is shown for 250 time steps, in figure 7.5 for 500 time steps, and in figure 7.6 for 1000 time steps. The curves depicted are the result of a simulation.[2]

One sees that, by letting α become smaller and observing more time steps, the process m in a sense converges to a smooth limiting process. Smaller values of α (which determine the size of the corrections made at each time step, see (7.6)) are compensated by a larger time range. Random irregularities of the curves representing the time development in this way vanish.

[2]The author is grateful to Matthijs Kadijk for providing these simulation results.

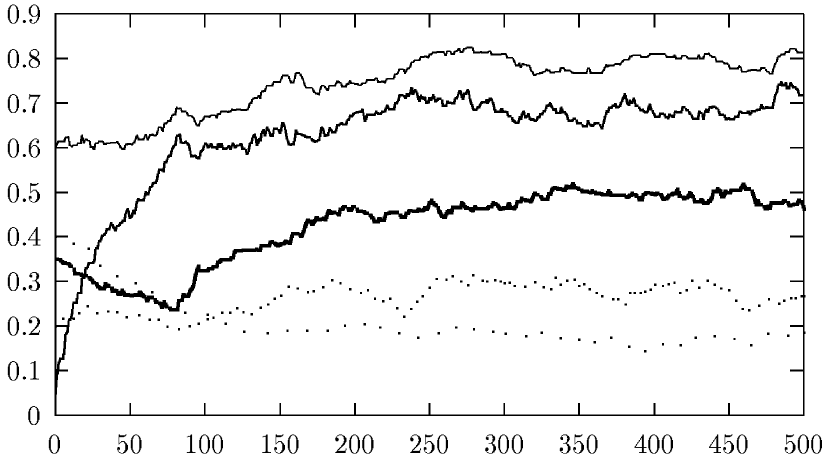


Figure 7.5. Self-ordering process with $\alpha = 0.05$, 500 time steps

To give an explanation of this fact, consider the case that for some k the $m_i(k)$'s for some k have already ordered, that is, $m(k)$ is monotonous. Without loss of generality, we may assume that the ordering is increasing (the opposite case is handled symmetrically). According to proposition 7.4.2, the ordering will be preserved for all later times. In this case, the description of the dynamics of the system is less complicated than in general. One easily checks that by equation (7.6) a value m_i can be affected only if the signal x lies in an interval S_i defined as follows:

$$
\begin{array}{rcllcl}
\text{for } i & = & 1: & S_i & = & [0, \frac{1}{2}(m_2 + m_3)] \\
\text{for } i & = & 2: & S_i & = & [0, \frac{1}{2}(m_3 + m_4)] \\
\text{for } 3 \leq i & \leq & l-2: & S_i & = & [\frac{1}{2}(m_{i-2} + m_{i-1}), \frac{1}{2}(m_{i+1} + m_{i+2}) \\
\text{for } i & = & l-1: & S_i & = & [\frac{1}{2}(m_{l-3} + m_{l-2}), 1] \\
\text{for } i & = & l & S_i & = & [\frac{1}{2}(m_{l-2} + m_{l-1}), 1].
\end{array}
$$

Since the probability distribution of the x_k's is known, one can calculate the expected change of the m_i's under the condition that the state at time k is m,

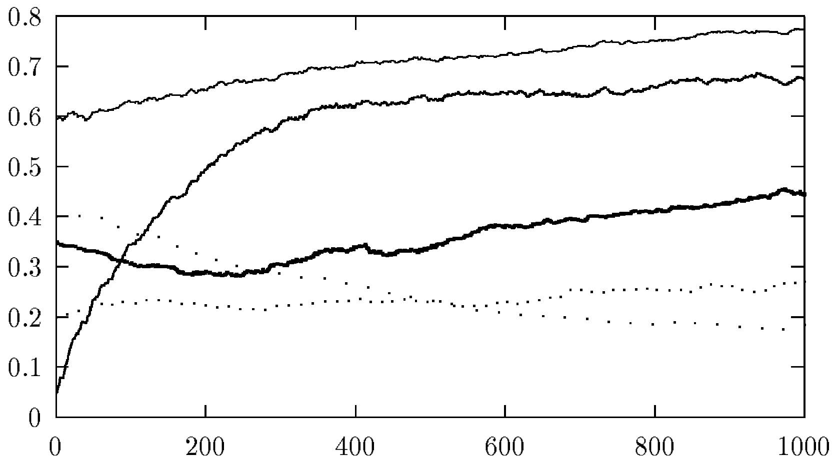


Figure 7.6. Self-ordering process with $\alpha = 0.01$, 1000 time steps

using formula (7.6). An easy integration shows

$$\text{for } 3 \le i \le n-2: \quad E(m_i(k+1) - m_i(k) \,|\, m(k) = m) = \frac{\alpha(k)}{2}[(\frac{m_{i+1}+m_{i+2}}{2} - m_i)^2 - (\frac{m_{i-1}+m_{i-2}}{2} - m_i)]^2.$$

Here $E(\;|\;)$ denotes the conditional expectation. Similar formulas, corresponding to (7.9), hold for the cases $i = 1$, $i = 2$, $i = n-1$, $i = n$.

Now, in the long run, the actual distribution of the signals x_k will resemble with high probability their stationary probability distribution. This is a consequence of the law of large numbers (Lamperti, 1966), which can be applied here since the x_k's are mutually independent. According to the subsequent theorem 7.4.2, the dynamics of $m(k)$ is therefore approximated by the following deterministic dynamics $u(t)$ corresponding to (7.6) in which the influence of the random x_k's is modeled simply as if at every instant of time they were smeared out over $[0,1]$ according to their probability distribution:

$$\text{for } 3 \le i \le n-2: \quad \frac{du_i}{dt} = \int_{S_i} (x_i - u_i)d\lambda$$

$$(7.13) \qquad = \frac{1}{2}[(\frac{u_{i+1}+u_{i+2}}{2} - u_i)^2 - (\frac{u_{i-1}+u_{i-2}}{2} - u_i)^2]$$

Similar equations apply to the cases $i = 1$, $i = 2$, $i = n-1$, $i = n$.

u is a function $N \times \mathrm{R}_+ \to S$, where $\mathrm{R}_+ = \{t \in \mathrm{R} \,|\, t \ge 0\}$ plays the role of time. The approximation argument is made precise by the following theorem. By m^α for $\alpha > 0$ we denote the process m as defined above with $\alpha(k) = \alpha$ constantly for all k.

Theorem 7.4.2 *Consider the process $m^\alpha(k)$ defined as above with $\alpha > 0$ fixed instead of $\alpha(k)$. Let P denote the probability law associated to the process $m(k)$. Define*

$$M^\alpha(t) = m^\alpha([\alpha^{-1}(t)]) \qquad (t \in \mathrm{R}_+),$$

where $[\alpha^{-1}t]$ denotes the largest natural number which is $\le \alpha^{-1}t$.

Let $m(0) \in S^n$ be monotonous, i. e., $m_i^{(0)} \le m_j^{(0)}$ for $i \le j \le n$. If u denotes the solution of (7.13) *with initial value $m^{(0)}$, we have*

$$M_\alpha \to u \text{ as } \alpha \to 0$$

in the following sense: for any time interval $[0,T]$, $T > 0$, for any $\varepsilon > 0$,

$$(7.14) \qquad P(\sup_{t\in[0,T]} |M^\alpha(t) - u(t)| > \varepsilon) \to 0 \text{ as } \alpha \to 0.$$

This means that the probability that $M^\alpha(t)$ deviates from $u(t)$ for some $t \in [0,T]$ by more than ε converges to 0 as $\alpha \to 0$. □

This theorem intuitively says the following. If in equation (7.6) the size of the corrections due to single signals is decreased (via α) but, on the other hand, more corrections are made due to a correspondingly larger time scale, the resulting process converges to the deterministic process u. Since the sequence $\alpha(k)$ in (7.6) was required to converge to 0, this shows that, for large times, the process $m(k)$ is approximated by the deterministic process u. The proof of theorem 7.4.2, which is rather technical, will be given in the appendix.

We have thus completed our task of giving a formalization of the self-organizing processes mentioned earlier. The formalization enabled us to formulate some properties that give, at least for a simple model system, an explanation of the ordering phenomenon occurring in the self-ordering processes concerned. Theorem 7.4.1 and proposition 7.4.2 show that, in a probabilistic sense, ordering will occur almost surely, if the function $\alpha(k)$ is suitably chosen. Theorem 7.4.2 shows that in a sense there exists a deterministic approximation of the process m. Future research may aim at a more rigorous understanding of the asymptotic properties of the process m. In particular, a further study of the limiting behavior of the process u for large times might leed to a proof of the ordering property stated on page 230, using the approximation result given by theorem 7.4.2.

An analysis of the asymptotic behavior of a similar self-organizing process, based on a different method of proof, has been given by Ritter & Schulten (1989). In that paper, among other things, also the statistical fluctuations around the asymptotic equilibrium state are studied. The asymptotic state itself has been investigated in Ritter & Schulten (1986).

Exercise 7.4.1 Show for simple cases of non-monotonous initial conditions (for example: $n = 3$, $0 < m_1 < m_3 < m_2 < 1$) how a sequence of signals x_k can be selected so that the vector m becomes monotonous by applying the dynamics (7.6). Suppose $\alpha(k)$ has some constant value between 0 and 1. □

Exercise 7.4.2 Give a proof of proposition 7.4.1 (Hint: select the sequence of signals $\xi_1, \dots, \xi_r$ according to the following strategy. By induction on $l < n$, "clean up" the intervals $[0, m_l]$ in the sense that none of the points $m_{l+1}, \dots, m_n$ falls into the interval $[0, m_l]$. For $l = n$, this means that m is monotonously increasing. If a monotonously decreasing state is aimed at, work with intervals $[m_l, 1]$ instead of $[0, m_l]$. The question whether m should become monotonously increasing or monotonously decreasing depends on the position of m_n relative to m_1. If $m_n > m_1$ at the beginning, let m become monotonously increasing, otherwise monotonously decreasing.) □

Exercise 7.4.3 Give a proof of formula (7.12) in the proof of theorem 7.4.1. (Hint: Show that the probability of the complementary event, namely the event that $\Delta(s)$ will stay 0 for all times, equals $\prod_{s=1}^{\infty}(1 - p(\alpha(s)))$. Do that by showing that for each t

$$Prob(\Delta(s) = 0 \text{ for } s = 1, 2, \dots, t) = \prod_{s=1}^{t}(1 - p(\alpha(s))).$$

Prove that by induction on t, using (7.11) and the Markov property of Δ. □

Exercise 7.4.4 Give a proof of proposition 7.4.2. Show that, under the assumption that the initial configuration $m(0)$ is monotonous, for any choice of the signal x_0, $m(1)$ as calculated according to equation (7.6) is monotonous. □

EXERCISE 7.4.5 Let (Ω, P) be a probability space and let X_m for $m \in N$ and Y be random variables $\Omega \to \mathrm{R}$. The sequence $(X_m)_{m \in N}$ is said to converge to Y in probability if the following holds: for any $\varepsilon > 0$,

$$P(|X_m - Y| > \varepsilon) \to 0 \text{ as } m \to \infty.$$

Check how this concept of convergence is related to the following alternative ones. (Check if some of these concepts are related by an implicative relation by proving the implication or by finding counterexamples. In counterexamples, one can conveniently use $\Omega = [0,1]$ with the uniform probability measure.)

(i) for all $\omega \in \Omega$, $X_m(\omega) \to Y(\omega)$ as $m \to \infty$
(ii) there exists $\omega \in \Omega$ such that $X_m(\omega) \to Y(\omega)$ as $m \to \infty$
(iii) $E(X_m) \to E(Y)$ as $m \to \infty$ where E denotes the expected value
(iv) $E|X_m - Y| \to 0$ as $m \to \infty$.

Is the concept of weak convergence in any way related to the statement (7.14)? □

7.5 Appendix:

In this appendix we prove theorem 7.4.2.

Let us fix a monotonous initial value $m^{(0)} \in S^n$ and consider the corresponding processes $M^\alpha(t)$ and $u(t)$ with $u(0) = m^{(0)}$. Let us also fix $T > 0$. We are going to verify the claim for this T. Choose $\tau > 0$. We will estimate how much $M^\alpha(\tau)$ differs from $u(\tau)$, if α and τ are small.

Let δ_x for any $x \in \mathrm{R}$ denote the point mass at point x, that means the measure given by

$$\delta_x(A) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{otherwise} \end{cases} \quad \text{for any } A \subset \mathrm{R}$$

Let ν be the (random) measure

$$\sum_{k=0}^{[\tau\alpha^{-1}]} \delta_{x_k}$$

This is the distribution of signals x_k up to time $\tau\alpha^{-1}$ where $[\tau\alpha^{-1}]$ is the integer part of $\tau\alpha^{-1}$.

Now define the vector $(M_i^{(\alpha,\tau)})_{i \le n}$ componentwise by

$$(M_i^{(\alpha,\tau)}) = m_i(0) + \int_{S_i} [x - m_i(0)] d\nu(x).$$

This vector describes the state of the following alteration of the process M^α at time τ: the state vector is not updated at every time step; instead, signals x_k are collected over the time interval $[0, \tau\alpha^{-1}]$ and then updating is done all at once for this collection of signals. The areas of influence S_i are defined as on pg. 234, referring to the initial state $m(0)$.

A suitable quantification of the difference of $M^\alpha(\tau)$ and $M^{(\alpha,\tau)}$ is the term

$$E|M^\alpha(\tau) - M^{(\alpha,\tau)}|$$

where E denotes the expected value.

One shows, by using the dynamical equation of M^α, that

$$E|M^\alpha(\tau) - M^{(\alpha,\tau)}| = o(\tau) \tag{7.15}$$

uniformly in α. The notation $F(\tau) = o(\tau)$ for real functions F means that

$$\tau^{-1}F(\tau) \to 0 \quad \text{as } \tau \to 0.$$

Now, by the law of large numbers (Lamperti, 1966), if λ denotes the Lebesgue measure on $S = [0,1]$ (i. e., the uniform measure), we have $\alpha\nu \to \tau\lambda$ as $\alpha \to 0$ in the sense that for any bounded continuous function $f : S \to \mathbf{R}$

$$E|\alpha \int_S f d\nu - \tau \int_S f d\lambda| \to 0 \text{ as } \alpha \to 0.$$

Consequently, for each $i \leq n$ and for each $\tau > 0$,

$$E|M_i^{(\alpha,\tau)} - (m_i(0) + \tau \int_{S_i} x - m_i(0) d\lambda)| \to 0 \text{ as } \alpha \to 0. \tag{7.16}$$

Corresponding to the estimate (7.15) one obtains for $u(\tau)$ the estimate

$$|u_i(\tau) - [m_i(0) + \tau \int_{S_i} x - m_i(0) d\lambda]| = o(\tau) \tag{7.17}$$

So combining (7.15)–(7.18) we have the estimation

$$E|M^\alpha(\tau) - u(\tau)| \leq \varphi(\tau) + \psi(\alpha) \tag{7.18}$$

for some functions φ and $\psi : \mathbf{R}_+ \to \mathbf{R}$ satisfying

$$\varphi(\alpha) \to 0 \text{ as } \alpha \to 0$$

$$\tau^{-1}\varphi(\tau) \to 0 \text{ as } \tau \to 0.$$

We note that all estimates (7.15)–(7.18) hold uniformly for all initial values $m^{(0)} \in S^n$. Applying the same argument for each time interval $[\kappa\tau, (\kappa+1)\tau]$, $\kappa = 1, 2, \ldots, [\frac{T}{\tau}]$, to the solution $u(t)$ of (7.13) with random initial condition $u(\kappa\tau) = M^\alpha(\kappa\tau)$, one gets the estimate

$$E|M^\alpha(\kappa\tau) - u(\kappa\tau)| \leq C\tau^{-1}\varphi(\tau) + \psi(\alpha) \tag{7.19}$$

for some constant $C > 0$ for all $\kappa \leq [\frac{T}{\tau}]$, using differentiability of $u(t)_{t \leq T}$ with respect to the initial value $u(0)$ (see Coddington & Levinson, 1955).

Moreover, using (7.19) for a sufficiently dense net of points of the form $\kappa \cdot \tau (\tau > 0, \kappa \leq [\frac{T}{\tau}])$ in $[0, T]$ and taking into account the boundedness of all parameters appearing in the dynamical equations for u and M^α, one then gets for every $\varepsilon > 0$

$$P(\sup_{t \leq T} |M^\alpha(t) - u(t)| > \varepsilon) \to 0 \text{ as } \alpha \to 0$$

as we claimed.

We note that the convergence (7.19) is again uniform in the initial condition $m^{(0)} \in S$.

References

Buhmann, J., Divko, R., Ritter, H., & Schulten, K. (1987). Physik und Gehirn. Wie dynamische Modelle von Nervennetzen natürliche Intelligenz erklären [Physics and brain. How dynamic models of neural nets explain natural intelligence]. *mc–grundlagen*, **9**, 108–120.

Coddington, E., & Levinson, M. (1955). *Theory of Ordinary Differential Equations*. New York: Mc Graw-Hill.

Kohonen, T. (1988). *Self-Organization and Associative Memory* (2nd ed.). New York: Springer-Verlag.

Lang, S. (1971). *Linear Algebra*. Reading, MA: Addison Wesley.

Lamperti, J. (1966). *Probability*. Reading, MA: Benjamin.

Ritter, H., Martinetz, T., & Schulten, K. (1992). *Neural computation and self-organizing maps: An introduction*. New York: Addison Wesley.

Ritter, H., & Schulten, K. (1986). On the stationary state of Kohonen's self-organizing sensory mapping. *Biological Cybernetics*, **54**, 99–106.

Ritter, H., & Schulten, K. (1989). Convergence properties of Kohonen's topology conserving maps: Fluctuations, stability and dimension selection. *Biological Cybernetics*, **60**, 59–71.

Rumelhart, D., Mc Clelland, J., & the PDP research group (1986). *Parallel distributed Processing. Explorations in the Microstructure of Cognition.* Cambridge, MA, MIT Press/Bradford Books.

List of Symbols

Symbols	*Meaning*
logical	
$\neg$	not
$\wedge$	and
$\vee$	or
$\Rightarrow$	implies
$\Leftrightarrow$	iff
$\exists$	exists
$\forall$	for all
set theoretical	
$\in$	element of
$\notin$	not element of
$\subset$	binary subset relation
$\subseteq$	reflexive binary subset relation
$\cup$	union of sets
$\cap$	intersection of sets
$\emptyset$	empty set
$\{a_1, a_2, \ldots, a_n\}$	listing elements of sets
$\mathcal{P}(M)$ or 2^M	power set of M
lattice theoretical	
$\sqsubseteq$	lattice theoretical binary relation
$\sqcup$	lattice theoretical union
$\sqcap$	lattice theoretical intersection
structure theoretical, algebraic, and others	
$\langle\ \rangle$	angle brackets (listing a relational structure)
$\times$	Cartesian product of sets
$f : M \longrightarrow N$	function f mapping set M to set N
$x \longmapsto y$	mapping element x to element y
$\leq$	binary relation less or equal than, *on numbers only*
$\preceq$	binary relation, *psychological*
$\sum$	sum symbol
$\mathbf{R}$	set of real numbers
$\mathbf{N}$	set of natural numbers

Author Index